Theory and Algorithms for Hidden Markov Models and Generalized Hidden Markov Models

by

Daniel Ray Upper

B.S. (Rice University) 1989

A dissertation submitted in partial satisfaction of the requirements for the degree of

Doctor of Philosophy

in

Mathematics

in the

GRADUATE DIVISION

of the

UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor Morris W. Hirsch, Chair
Doctor James P. Crutchfield
Professor Jacob Feldman
Professor David R. Brillinger

1997

Theory and Algorithms for Hidden Markov Models and Generalized Hidden Markov Models

Copyright © 1997
by
Daniel Ray Upper

Abstract

Theory and Algorithms for Hidden Markov Models and Generalized Hidden Markov Models

Hidden Markov Models (HMMs) are widely used in pattern recognition applications, most notably speech recognition. However, they have been studied primarily on a practical level, with the HMM matrices as the fundamental objects and without considering the viewpoint of an observer trying to accurately predict the future output. This dissertation is a study of the processes represented by HMMs using the concepts and techniques of stochastic automata derived from the study of dynamical systems and their complexity. The goal is to understand these processes in the language of stochastic automata. Along the way, certain ideas of stochastic automata are characterized in a measure-theoretic manner.

We begin by defining a process to be a stationary measure space of bi-infinite sequences. We define a process state to be a conditional distribution on the future of a process which corresponds to the state of knowledge held by an observer who has seen some or all of the process's history. This definition is similar in spirit to ideas used in dynamical systems, and it is a formalization of the notion of a "deterministic state" used in automata theory. We describe the process states of HMM processes. And we give a necessary condition for a process to have an HMM representation.

Following [7] and [8], we define Generalized Hidden Markov Models (GHMMs). These are structurally and operationally the same as HMMs, except that parameters which are interpreted as probabilities in defining HMMs are allowed to be negative in GHMMs. We describe necessary and sufficient conditions for two GHMMs (and thus two HMMs) to represent the same process, and we give a method for finding the smallest possible GHMM equivalent to a given one.

Going further, we give an algorithm for constructing a GHMM that represents a process from the probabilities that process assigns to words. We prove that, for every process, either the algorithm constructs a GHMM that represents the given process or that no such representation exists. This characterizes the set of process representable by GHMMs. Finally, we describe an implementation of this algorithm which constructs GHMMs from sample sequences.

Contents

1 Introduction 1
2 Processes and Process States 7
2.1 Sequence Space 7
2.2 Processes 9
2.3 Past and Future 11
2.4 Histories and States 13
2.5 Process States 15
2.6 Transient and Recurrent States 20
2.7 Synchronization 23
3 Hidden Markov Models 25
3.1 Notation for Markov Chains 25
3.2 Hidden Markov Models 27
3.3 HMMs as Processes 30
3.4 Mixed States 35
3.5 Examples 43
3.6 When Does a Process have an HMM Presentation? 49
4 Generalized Hidden Markov Models 52
4.1 Generalized Hidden Markov Models 52
4.2 Redundancy and Linear Algebra 61
4.3 Equivalence and Minimization of GHMMs 72
5 Reconstruction 83
5.1 Constructing a Presentation for a Process 84
5.2 Reconstruction from a Sample 94
6 Conclusions and Further Directions 104
6.1 Process states and presentations 104
6.2 Generalized Hidden Markov Models 106
6.3 Converting GHMMs to HMMs 107
6.4 Reconstruction 110
6.5 Last remarks 110
Appendix A Notation 112
Appendix B Selected Probability Theory 116
B.1 Kolmogorov's Extension Theorem and Process Existence 116
B.2 Martingales 123
Bibliography 126

Acknowledgments

I thank my coworkers James Crutchfield, Karl Young, Jim Hanson, Lisa Borland, and Deirdre Des Jardins for the many discussions and suggestions that helped me develop this work.

I thank James Akao, William Grosso, and the members of the games group for their camaraderie and collegiality.

I thank Dorothy Brown for helping me focus on my writing.

I thank Professor Feldman and Professor Brillinger for their feedback on the various drafts of my dissertation.

I thank Professor Hirsch for giving me the freedom to take off in my own directions and work closely with Doctor Crutchfield. I would like to thank Doctor Crutchfield for his insights, advice, and enthusiasm and for understanding what I was doing. I am indebted to both of these advisors for the support, financial and otherwise, they have given me over the years.

Finally, I thank my wife, Nancy Jamieson, for her everyday presence, her emotional support, and her help in many practical ways, which together enabled me to complete this dissertation.

This work was supported at UC Berkeley by grants ONR N00014-95-1-0524 and AFOSR 91-0293 and at the Santa Fe Institute by grant ONR N00014-95-1-0975.

1 Introduction

This dissertation is about a class of stochastic processes that are usually presented as having a finite set of states, but which, in another sense, may have an infinite number of states. These processes are known variously as *Hidden Markov Models* (HMMs), functions of a Markov Chain, or stochastic finite automata, all of which are essentially equivalent. HMMs are used most widely and will be used here.

A process in the HMM class can be described as a finite-state Markov Chain with a memoryless output process which produces symbols in a finite alphabet. This is the sense in which these processes have finitely many states. However, from the perspective of an observer who knows the parameters of some representation of the process and is able to observe the output symbols but not the internal states, things look different. For some processes there are infinitely many distinct states of such an observer's knowledge about the status of the process. This knowledge is defined in terms of conditional distributions on future symbols. This is the sense in which there can be infinitely many states. These states are more relevant than the original finite set of states to the study of the process, since they allow for optimal prediction.

Functions of Markov Chains were the first descriptions of these processes to be studied, and they were initially studied as mathematical information sources [1]. There were a handful of papers such as [2] published in the 1950s and early 1960s, which define HMMs and lay out these theoretical questions. What is the entropy rate for a function of a Markov Chain? Do these two functions of Markov Chains define the same process (the identifiability question)? What is the smallest function of a Markov Chain equivalent to the given one (the minimality question)? This work was done by researchers with mathematical backgrounds, studying HMMs from a perspective of probability and information theory.

From the 1970s onward, HMMs have been used for modeling observed patterns, especially in speech recognition. There are a large number of papers, such as [3–5], that present HMMs as tools for use on these practical problems. These papers are written by researchers interested in pattern recognition, often from a viewpoint in engineering or computer science, and they usually focus on algorithms and on results in practical situations.

A new insight in 1987 led to a generalization of HMMs and to resolution of the identifiability and minimality questions about Hidden Markov Models [6]. These results appear in a very few papers that treat HMM matrices as objects of linear algebra without regard to the signs of the transition matrix entries — which have traditionally been interpreted as probabilities. These papers include [7] and [8], which solve the identifiability and minimality questions for HMMs by solving them for a generalization of HMMs known as *Generalized Hidden Markov Models* (GHMMs).

In addition to this development of HMMs per se, there is a substantial literature in computer science dealing with finite state machines. An emerging branch of this literature deals with stochastic finite automata, and falls under the heading of complex systems. This body of work, composed of papers such as [9], focuses on intrinsic processes rather than representations, and looks at stochastic automata as the simplest systems in which to study how natural systems process information. Because the definition of a stochastic finite automaton is quite similar to the definition of a Hidden Markov Model, this work provides an alternative way of looking at HMMs.

This dissertation developed from looking at HMMs from the viewpoint of the work on stochastic finite automata. This approach led to the material of chapters 2 and 3. The work in chapters 4 and 5 followed from this and used the generalization of HMMs mentioned above. The next paragraphs contain brief overviews of these chapters and are intended to give the reader an idea of what is to come.

The primary objects of chapter 2 are *processes* and *process states*. A process is a stationary probability measure on the space of bi-infinite sequences of symbols, where a *symbol* is an element of a finite set called the alphabet.* Indices into these sequences are thought of as times, as if the process were a laboratory apparatus that emits a symbol with every tick of a clock. Thus, negative indices refer to symbols that were emitted in the past and that may be known, and nonnegative indices refer to symbols that have not yet been emitted and may not have been internally determined yet. We define a word to be a finite string of symbols in the alphabet. A process assigns a probability to each word and is uniquely determined by these probabilities.

*In the stochastic process literature, the term "state" is usually used where we use "symbol". In papers such as [2,10], this leads to the somewhat confusing use of "state" to refer to both symbols and presentation states. In this dissertation, the term "state" is used exclusively to refer to objects which render the future conditionally independent of the past.

A process state is a conditional distribution over the future — a measure space of semi-infinite sequences of symbols — induced by conditioning on known historical information, such as what particular symbols were emitted at the last few ticks of the clock. The process states are the possible states of knowledge of an observer who wishes to predict the future symbols with high accuracy. This observer knows the design of our hypothetical apparatus, but not the current status of its internal components.

The definition of a process state is new in this dissertation, but the underlying idea is not. It is used, for example, in [11]. What the author has done here is to formalize this fundamental idea. The definition of a process is, of course, standard.

In chapter 3, we will introduce Hidden Markov Models (HMMs). An HMM consists of a recurrent finite-state Markov Chain, an alphabet of output symbols, and a distribution over that alphabet for each transition in the Markov Chain. The states and transitions of the Markov Chain are hidden from observation so that only the output symbols are visible. We represent an HMM primarily by a set of matrices $\left\{T^k\right\}$, one matrix T^k for each symbol k in the alphabet — note that the superscript k is an index, not an exponent. Each entry T^k_{ij} is the probability, if the Markov Chain is in state i, of emitting symbol k and going to state j. An HMM defines a process in a natural way, and so HMMs provide a convenient way to represent some processes. The states of an HMM's underlying Markov Chain are quite different from process states. We will refer to the Markov Chain's states as *presentation states*.

We compute the process states for a process represented by an HMM in terms of the presentation. We show that they are represented by probability distributions over the presentation states or, equivalently, by mixtures of presentation states. We call them *mixed states*† and we identify the real role of presentation states. The presentation states are the things we combine to make mixed states. Essentially, presentation states are basis vectors for a vector space containing the mixed states, and mixed states represent the states of knowledge of an observer. In this interpretation, the matrices T^k define linear transformations among these vectors.

Finally, we consider the question of when a process can be represented by an HMM. We prove that the following condition is necessary: if a process has an HMM presentation, then the span of the process states — a subspace of the space of conditional

†The term *mixed states* is used in the context of HMMs, with an entirely unrelated meaning in [12].

distributions on the future — is finite-dimensional. This result is a natural consequence of the fact that the process states of such a process can be represented by linear combinations of presentation states.

The definition of an HMM given in chapter 3 is one of several equivalent definitions. Mixed states first appeared in [1], although not with that name. Most of the remaining material in the chapter has not previously appeared in print, but has undoubtedly been deduced a number of times before. Stating these results in terms of process states is the author's work, as is the question, "What are the process states for a process represented by an HMM?" and the resulting interpretation of presentation states as basis vectors, both of which are crucial to this dissertation. The final section of this chapter is entirely the author's own.

Chapter 4 introduces a new class of presentations, Generalized Hidden Markov Models. We represent a GHMM like an HMM, by a matrix for each symbol, but in a GHMM we do not interpret the entries in these matrices as probabilities. Indeed, we allow them to be negative or greater than one. We do constrain these entries, and we constrain them in such a way that the calculations we use with HMMs produce meaningful results for GHMMs. In this way, a GHMM assigns probabilities to words and so it defines a process.

We justify this change as follows. For a process represented by an HMM, we calculate the probabilities of words by simple linear algebra and we represent process states as vectors in a space generated by the presentation states. But because we restrict the entries in the matrices T^k to be positive, we restrict the class of linear transformations they can represent and we restrict the set of possible mixed states to a small subset of the vector space. If we remove this constraint, we can make use of all sensible linear transformations and we can use any portion of the vector space. This is what we allow when we use GHMMs.

Two long-standing problems for HMMs can be readily solved in terms of GHMMs, namely the equivalence question — Do these two HMMs (or GHMMs) represent the same process? — and the minimization question — How small is the smallest HMM which is equivalent to a given HMM? These questions were resolved in [7]. The present work presents a new and relatively clean resolution of these questions that is similar in

spirit to the work just cited. But it is different in details and includes an important new construction, the standard presentation.

Chapter 5 contains two significant results. The first is a proof that every process such that the span of its process states is finite-dimensional can be represented as a GHMM. We show this by constructing such a representation. This completes a characterization of the processes representable by GHMMs, the first half of which appears for HMMs in chapter 3 and is generalized to GHMMs in chapter 4. The complete characterization is this: a process has a GHMM presentation if and only if the span of its process states is finite-dimensional.

It is noteworthy that we can actually construct a GHMM presentation for any process which has one. This construction leads to the second result of chapter 5: a technique for constructing a GHMM from a sample of the output from a process. This technique, which we call the *reconstruction algorithm*, is completely unlike the forward-backward algorithm, the most widely used technique for constructing HMMs from sample output. It needs further development, but it has much more solid theoretical footing than the forward-backward algorithm. Indeed, it may eventually replace the forward-backward algorithm. The work in this chapter is entirely that of the author.

In the chapter overviews, we introduced a number of concepts that may be unfamiliar to the reader. We conclude the introduction with an example, to make some of them clearer and more concrete to the reader. Let us consider the Hidden Markov Model, depicted in figure 1.1.

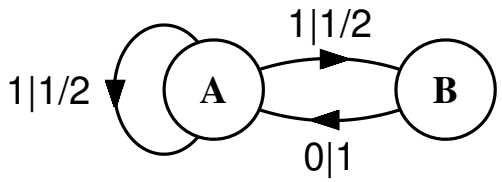

Fig. 1.1 First example HMM.

The circles **A** and **B** represent presentation states, the states of the underlying Markov Chain. The arrows connecting them represent the transitions of the HMM and the labels indicate the symbol that will be emitted if that transition is taken and the probability of that transition. For example, the label 1|1/2 at the top indicates that the symbol 1

is emitted when this transition is made and that this transition is taken on $1/2$ of the occasions when a transition is made from state **A**. We represent this same HMM with the transition matrices

$$T^0 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \text{ and } T^1 = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 0 \end{pmatrix}. \tag{1.1}$$

This is not the full description of an HMM that we will give in chapter 3. In particular, we have not defined an initial state distribution for this HMM.

It should be clear to the reader that each presentation state in our example defines a distribution on the future symbol sequences. We call this the conditional distribution on the future given that presentation state.

Now, suppose that we know the transition matrices and can see the output of this machine, but do not know what presentation state the HMM is in. That is, the presentation states are hidden from us, whence the term *Hidden* Markov Models. If the most recent symbol we have seen is a 0, then we can deduce that the HMM is in presentation state **A**. But if the most recent symbol we have seen is a 1, we cannot deduce which presentation state the HMM is in. We can, however, infer a distribution over the presentation states. The HMM is in either state **A** or state **B** with probability $1/2$ each. And we can deduce a conditional distribution on the future given that the most recent symbol was a 1. If we do this we will find that it is equal to $1/2$ of the sum of the two conditional distributions on the future given presentation states **A** and **B**, respectively.

In the terminology of this dissertation, we have now seen two process states. The first is the conditional distribution on the future given that the most recent symbol was a 0. This happens to be identical to the conditional distribution on the future given presentation state **A**. The second is the conditional distribution on the future given that the most recent symbol was a 1, which does not correspond to either presentation state. We can represent the first of these process states by the mixed state $(P_{\mathbf{A}} = 1, P_{\mathbf{B}} = 0)$, a distribution over the presentation states which puts all probability on state **A**. Likewise, we represent the second of our process states by the mixed state $(P_{\mathbf{A}} = 1/2, P_{\mathbf{B}} = 1/2)$, a distribution which makes each presentation state equally likely.

We begin in chapter 2 with a full definition of processes and process states.

2 Processes and Process States

In this chapter and the next one, we define *processes*, the central objects of this work. We will work with them in two ways, from two different viewpoints. First we will define processes in the abstract, as probability measures on a sequence space. Second, more concretely, we will define *Hidden Markov Models* — partially observed finite state Markov chains — which we will use to represent processes. In this chapter we will introduce the first of these viewpoints, and in chapter 3 we will introduce the second and then bring the two together.

This chapter builds on, and uses the concepts and terminology of, probability theory. Appendix B contains a few necessary definitions and theorems with which the reader may be unfamiliar. A presentation of the basics, if needed, may be found in a number of standard texts, e.g. [13,14].

The reader may find it helpful to keep in mind the following metaphor. A process may be thought of as a black box on a laboratory bench with a row of lights on it. These lights are our symbols, and the row of them is our alphabet. There is a flash from one or another of these lights every second, which we describe as the emission of a symbol. This box has been running forever, and will continue running forever. We may know the design of the box, and we may have observed a number of recent symbol emissions, but we cannot observe the current configuration of the box's internal parts.

2.1 Sequence Space

In this section, we define and introduce notation for the sets upon which we will build our probability spaces.

Begin with a finite set $\mathcal{X}$ of *symbols*, which we will call an *alphabet*. Our canonical choice will be $\mathcal{X} = \{0, 1, \ldots, m-1\}$ for some natural number m. Let $\mathcal{X}^{\mathbb{Z}}$ be the space of bi-infinite sequences of elements of $\mathcal{X}$. That is, if $\mathbf{x} \in \mathcal{X}^{\mathbb{Z}}$, then $\mathbf{x}$ is a bi-infinite sequence $\ldots x_{-3}x_{-2}x_{-1}x_0x_1x_2x_3\ldots$, so that for any integer t, there is a symbol $x_t \in \mathcal{X}$. We will think of the indices as denoting measurement times, with negative indices referring to the past and nonnegative indices indicating the future. Often, we will assume that the symbols with negative indices are known and the symbols with

nonnegative indices are unknown. We will want to think of each measurement time as being either in the past or in the future, so that we will only need to consider our state of knowledge after one event an before another, and we will not need to consider our state of knowledge "as a symbol is becoming known". Thus we will be thinking of time $t = 0$ as being in the future. The reader may wish to think of the "present" as a small negative number, perhaps $-\frac{1}{2}$. Because time $t = 0$ is the smallest future time at which a symbol is observed, we will refer to x_0 as the *next symbol*.

In these terms, a *word* w of length l is an l-tuple of elements of $\mathcal{X}$, $w \in \mathcal{X}^l$. We will denote the empty word, of length zero, by λ, and the length of the word w by $|w|$. For, example, if $0, 1 \in \mathcal{X}$, $|01101| = 5$. A *subsequence* s is a structure $s = (w, (a, b))$, where w is a word and (a, b) is a pair of times such that w has length $b - a + 1$. The subsequence s is said to be an *instance* of the word w from which it is formed, and w is said to be the *base word* of the subsequence $s = (w, (a, b))$. s may also be denoted $s_a s_{a+1} \ldots s_b$. a and b are called the *start time* and *end time*, respectively, of s. The length of a subsequence is the length of its base word. An instance of λ, then, is written $(\lambda, (a, a - 1))$. A sequence $\mathbf{x}$ is said to contain, or *match*, a subsequence $s = (w, (a, b))$ if, for all $t \in \{a, \ldots, b\}$, $s_t = x_t$. We will most often refer to the *next word*, which is any subsequence $s = (w, (a, b))$ with start time $a = 0$, or the *history suffix*, which is any subsequence with end time negative one $b = -1$.

When writing a subsequence which contains x_{-1} or x_0, we will sometimes use the decimal point to denote this and to imply the start and end times. For example, $1011.$ is denotes history suffix $(1011, (-4, -1))$ and $.0110$ is denotes next word $(0110, (0, 3))$. We will not always be precise about distinguishing words from subsequences, nor about using w for words and s for subsequences. If w and z are words, then wz denotes their concatenation. The set of all words will be denoted by $\mathcal{X}^*$; this set contains λ.

The set A_s of sequences which match a subsequence s,

$$\left\{ \mathbf{x} \in \mathcal{X}^{\mathbb{Z}} | x_i = s_i \text{ for all } i \in \{a, \ldots b\} \right\} \tag{2.1}$$

is called the *cylinder set* defined by s. If s is an instance of λ, The cylinder set defined by any instance of λ is $\mathcal{X}^{\mathbb{Z}}$.

2.2 Processes

Processes are the central objects of this work. The definition of a process is this: a process is a stationary probability measure on a space of sequences. In this section we will develop this definition.

A probability measure is a function which assigns probabilities to sets — in this case, sets of sequences. To what subsets of $\mathcal{X}^{\mathbb{Z}}$ will our process assign probabilities? That is, what is the domain of this function? We need to define this set of subsets of $\mathcal{X}^{\mathbb{Z}}$, which is called a σ-field. Our choice $\mathbb{X}$ is defined to be the σ-field generated by the cylinder sets. That is, $\mathbb{X}$ is the smallest collection of subsets of $\mathcal{X}^{\mathbb{Z}}$ such that:

1. for every subsequence s, $A_s \in \mathbb{X}$, and
2. $\mathbb{X}$ is closed under complements and countable unions.

The pair $\left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}\right)$ is the measurable space in which we will be working.

Essentially, a probability measure on $\left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}\right)$ is something which assigns probabilities to the cylinder sets defined by subsequences. If s is a sequence and A_s is the cylinder set of sequences which match s, we define $\mathbf{P}(s)$ — the probability of s — to be the probability of the cylinder set $\mathbf{P}(A_s)$. Because $\mathbf{P}$ is a probability measure, $\mathbf{P}\left(\mathcal{X}^{\mathbb{Z}}\right) = 1$. Recall that $\mathcal{X}^{\mathbb{Z}} = A_\lambda$, so we have $\mathbf{P}(\lambda) = 1$.

As we stated above, we will require our processes to be stationary. A *stationary* process is one in which the probability of a subsequence does not depend on its start time. Stationarity will allow us to disregard the time index when we do not need it explicitly. In particular, it allows us to define the probability of a word to be the probability of any instance of it, as we will see shortly. Virtually all the probability measures on sequence space addressed in this work are stationary, and the exceptions will be identified as such.

The *shift map* T on a sequence space $\mathcal{X}^{\mathbb{Z}}$ is a map $T : \mathcal{X}^{\mathbb{Z}} \to \mathcal{X}^{\mathbb{Z}}$ such that for all $\mathrm{x} \in \mathcal{X}^{\mathbb{Z}}$, for all t, $(T(\mathrm{x}))_t = \mathrm{x}_{t+1}$. The shift map "shifts" a sequence over by one; it moves the time origin.

Definition 2.1.1. A process $\mathcal{P}$ is *stationary* if for all sets $A \in \mathbb{X}$, $\mathbf{P}(T(A)) = \mathbf{P}(A)$.

Note that we do not need to require shift invariance in both directions, as it is automatic. Since T is invertible in a space of bi-infinite sequences, let $B = T^{-1}(A)$, and apply the definition of stationarity to B, and we have $\mathbf{P}\left(T^{-1}(A)\right) = \mathbf{P}(A)$.

Definition 2.1.2. A *process* $\mathcal{P}$ is a stationary probability space $\left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$. That is, $\mathcal{P}$ is a stationary probability measure $\mathbf{P}$ on the measurable space $\left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}\right)$.

The author has chosen to work exclusively with stationary processes for convenience and for reasons which have their roots in the historical development of this work. However, virtually all of the results in this dissertation are valid with a definition which does not require processes to be stationary and instead expects a process to have a start and stop at finite times. Some of the statements made here, and many of the proofs, require modifications in order to apply under such a definition.

Let w be a word and let s be an instance of w. If $\mathbf{P}$ is stationary, all instances of w have the same probability. Thus, we can define the probability of w to be $\mathbf{P}(w) = \mathbf{P}(s)$, so we can think of a process $\mathcal{P}$ as a function which assigns probabilities to words. We will use W_l to denote the set of all words of length l with positive measure. This leads us to two facts, which are trivial consequences of the properties of probability measures. The first of these is that, because the cylinder set induced by λ is defined to be all of $\mathcal{X}^{\mathbb{Z}}$,

$$\mathbf{P}(\lambda) = 1. \tag{2.2}$$

The second is that, for any word w and any length $l \geq 0$,

$$\sum_{z \in W_l} \mathbf{P}(wz) = \sum_{z \in W_l} \mathbf{P}(zw) = \mathbf{P}(w). \tag{2.3}$$

As it happens, the converse of this pair of facts is true — any function on $\mathcal{X}^*$ which satisfies equations 2.2 and 2.3 defines a process. This will be our primary tool for showing that a particular object is a stationary process.

In the statement of the theorem B.1.1, we have separated f from $\mathbf{P}$ for clarity. When we use this theorem , we will not usually mention f explicitly. Instead, we will use $\mathbf{P}$ in the role which f serves here, verify equations 2.2 and 2.3, and invoke the theorem to assert that $\mathbf{P}$ describes a unique process. This renders the distinction between f and $\mathbf{P}$ moot.

Theorem B.1.1. Given a map $f : \mathcal{X}^* \to [0, 1]$ satisfying

1. $f(\lambda) = 1$, and
2. For all words $w \in \mathcal{X}^*$, $f(w) = \sum_{z \in \mathcal{X}} f(zw) = \sum_{z \in \mathcal{X}} f(wz)$,

there is a unique process $\mathcal{P} = \left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$ such that for all $w \in \mathcal{X}^*$, $\mathbf{P}(w) = f(w)$.

This theorem follows directly from Kolmogorov's extension theorem. Proving it is conceptually simple but requires some rather serpentine logistics. Instead of appearing here as a interruption of the conceptual development, the proof appears in appendix B.1.

A few examples are in order before we go on. We will call the first of these the *fair coin*. This process should be thought of as a bi-infinite sequence of coin flips being revealed (or flipped) as time passes. Here $\mathcal{X} = \{\text{heads}, \text{tails}\}$, and x_t is the result of the coin flip at time t. Recall that we think of negative times as the past, and nonnegative times as the future, so that we have seen all the results x_t for $t < 0$ and we have not yet seen any x_t for $t \geq 0$. For any word w of length l, $\mathbf{P}(w) = 2^{-l}$. The symbols are independent and identically distributed (iid) so the process must be stationary. Every sequence in $\mathcal{X}^{\mathbb{Z}}$ is a realization (defined in section 2.3) of the fair coin process. Although we don't need theorem B.1.1 to prove that this process exists, we will verify that equations 2.2 and 2.3 are satisfied. For 2.2, $f(\lambda) = 2^0 = 1$. For 2.3, if w has length l and $z \in \mathcal{X}$, then $f(wz) = f(zw) = 2^{-(l+1)}$, so we have $2 \cdot 2^{-(l+1)} = 2^l = f(w)$.

The second of our examples is a strictly periodic process, in which a fixed word is repeated over and over. In this example, we choose the word 10000, and a typical realization looks like

$$\ldots 0001000010000100001000010000100\ldots \tag{2.4}$$

If the phase (i.e. the index associated to a 1, taken modulo 5) is uniformly distributed, $\mathbf{P}$ is stationary, and we have a process. It is not difficult to see that every word is assigned a probability of either 0 or $\frac{1}{5}$, with the exception of λ, 0, 00, and 000, which have probabilities 1, $\frac{4}{5}$, $\frac{3}{5}$, and $\frac{2}{5}$, respectively. This is clearly a process — the measure can be described explicitly. It assigns measure $\frac{1}{5}$ each to five bi-infinite sequence and 0 to all the others.

2.3 Past and Future

At times, we will need to treat the past and the future separately. In this section we will introduce a decomposition of a processes underlying probability space $\left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}\right)$ which will facilitate this.

Let $\mathcal{X}^+$, the *future space*, be the usual set of semi-infinite sequences of elements of $\mathcal{X}$,

$$\mathcal{X}^+ = \left\{ \mathbf{x}^+ = x_0, x_1, \ldots \,|\, \text{for all } i \geq 0,\ x_i \in \mathcal{X} \right\} \tag{2.5}$$

and $\mathcal{X}^-$, the *history space*, be the set of semi-infinite sequences of elements of $\mathcal{X}$ with negative indices:

$$\mathcal{X}^- = \left\{ \mathbf{x}^- = \ldots, x_{-2}, x_{-1} |\, \text{for all } i < 0,\ x_i \in \mathcal{X} \right\} \tag{2.6}$$

We will think of a bi-infinite sequence $\mathbf{x} \in \mathcal{X}^{\mathbb{Z}}$ as an ordered pair of semi-infinite sequences $\left(\mathbf{x}^-, \mathbf{x}^+\right)$, where $\mathbf{x}^- \in \mathcal{X}^-$ and $\mathbf{x}^+ \in \mathcal{X}^+$. Thus, we can write $\mathcal{X}^{\mathbb{Z}}$ as a product of sets, $\mathcal{X}^{\mathbb{Z}} = \mathcal{X}^- \times \mathcal{X}^+$. We will refer to an element $\mathbf{x}^-$ of $\mathcal{X}^-$ as a *history*, and an element $\mathbf{x}^+$ of $\mathcal{X}^+$ as a *future*. Thus a history suffix, which we defined in section 2.1 to be a subsequence with end time -1, is in fact a suffix of a history.

Next, we will define the *history σ-field* $\mathbb{H}$ and the *future σ-field* $\mathbb{F}$, both of which are σ-fields on $\mathcal{X}^{\mathbb{Z}}$ and are sub-σ-fields of $\mathbb{X}$. $\mathbb{H}$ and $\mathbb{F}$ are generated by the cylinder sets in $\mathcal{X}^-$ and $\mathcal{X}^+$ respectively. That is, $\mathbb{H}$ is defined to be the σ-field on $\mathcal{X}^{\mathbb{Z}}$ generated by all cylinder sets defined by subsequences with negative end times, and $\mathbb{F}$ is defined to be the σ-field on $\mathcal{X}^{\mathbb{Z}}$ generated by all cylinder sets with nonnegative start times. Thus if $s = (w, (a, b))$ is a subsequence with end time $b \leq -1$, then A_s — the set of all bi-infinite sequences $\mathbf{x} \in \mathcal{X}^{\mathbb{Z}}$ which match s — is an event in $\mathbb{H}$. We will refer to elements of $\mathbb{H}$ as *history events*. Intuitively, a history event is a set for which membership depends only on the history part of a sequence. A similar statement is true for $\mathbb{F}$, and we will refer to elements of $\mathbb{F}$ as *future events*.

In addition, we will need to define a series of finite history σ-fields $\{\mathbb{H}_l | l \in \{0, 1, \ldots\}\}$, where $\mathbb{H}_l$ is the σ-field on $\mathcal{X}^{\mathbb{Z}}$ generated by all length l history suffixes. ($\mathbb{H}_0$ is the trivial σ-field on $\mathcal{X}^{\mathbb{Z}}$.) An event in $\mathbb{H}_l$, then, is the set of bi-infinite sequences $\mathbf{x}$ which satisfy some in positions $x_{-l}, \ldots, x_{-1}$. As the reader may readily verify, the finite history σ-fields form an increasing sequence. That is, for all $l \geq 0$, $\mathbb{H}_l \subset \mathbb{H}_{l+1}$. In addition, the union of the finite history σ-fields is the (infinite) history σ-field:

$$\bigcup_{l=0}^{\infty} \mathbb{H}_l = \mathbb{H}. \tag{2.7}$$

A *realization* from a process $\mathcal{P}$ is a bi-infinite sequence in $\mathcal{X}^{\mathbb{Z}}$ which, loosely speaking, is within the support of the process' measure $\mathbf{P}$. Formally, the support of a measure depends on the topology of the measure space, and we have not defined one. Instead of doing so now, we will define a realization directly. Take a sequence $\mathrm{x} \in \mathcal{X}^{\mathbb{Z}}$, and consider the set of all subsequences which x contains. If, for each of these subsequences, the set of sequences which contains it has positive measure, then x is a realization of $\mathcal{P}$. In the same vein, a *null word* is a word with probability zero; and a history x^{-} is a *null history* if it has a suffix that is a null word.

2.4 Histories and States

In the last section, we established our notation and technical framework. The material in this section has these components, but it also has a conceptual component. We will defer most of the technical details until the next section. The material in this section draws on the concepts and terminology of Markov chains. Readers unfamiliar with Markov chains may wish to peruse section 3.1 at this time.

Consider the following situation from a time series perspective. Suppose we are watching a sequence of symbols appear as the output of an apparatus. Suppose further that this apparatus is known to be described by the process $\mathcal{P} = \left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$. Given $\mathbf{P}$ and some natural additional information — namely, the most recent few symbols — what can we predict about the future? In particular, how can we simulate future output from this apparatus?

We may assume, due to stationarity, that the next symbol will appear at time $t = 0$. Then the symbols we know form a subsequence w with stop time $b = -1$ — that is, w is a history suffix. We will make the reasonable assumption that $\mathbf{P}(w) > 0$. Then w induces a conditional distribution on x_0, $\mathbf{P}(x_0 = k|w)$.

Suppose we choose a symbol x_0 according to this conditional distribution, and build a new history suffix z from it and w shifted by one: $z_{-1} = x_0$, and $z_{t-1} = w_t$ if t is between w's start and end times. We now find ourselves in the same situation we started in, only one time step further along and with a new history suffix. We can repeat this process as many times as we like and thus (theoretically, at least) generate synthetic data.

In effect, we are using the history suffixes as states of an infinite Markov chain. If we output each new symbol as we choose it, we can accurately simulate the original

apparatus. Essentially, we've built what's called a Hidden Markov Model representation of our process. (A Hidden Markov Model is a partially observed Markov chain — see chapter 3.) However, because there are infinitely many history suffixes, it has infinitely many states, and none of these states are recurrent. That is, once a state has been seen it can never be revisited. This recasting of the process is not useful in the sense that it doesn't tell us anything new beyond knowing $\mathcal{P}$, and we have to use all the history we know in each simulation step. Is there a more concise or informative way of simulating $\mathcal{P}$?

Let's back up, and look at what we can say about distributions on the future space. Our known history suffix w induces a conditional distribution $\mathbf{P}(\cdot|w) \equiv \mathbf{P}(\cdot|A_w)$ on the future space $(\mathcal{X}, \mathbb{F})$.

$\mathbf{P}(\cdot|w)$ is an example of a *conditional future distribution*, a distribution on the future which arises when we condition on a history event. Conditional future distributions are of substantial importance. The conditional future distribution reflects our *state of knowledge* about possible future observations from $\mathcal{P}$. It includes all the information we have about the future — if we know the conditional future distribution induced by a history suffix, we may as well forget the history suffix itself. This, loosely speaking, is the sense in which $\mathbf{P}(\cdot|w)$ is a state.

In the next section, we will consider conditional future distributions which arise when we condition on either a history s or a history suffix $\mathtt{x}^{-}$. Here, however, we will restrict ourselves to distributions which arise when we condition on history suffixes. If two history suffixes, y and z, lead to the same conditional future distribution, $\mathbf{P}(\cdot|z) = \mathbf{P}(\cdot|y)$, then we will say that they are equivalent, denoted $y \sim z$. They provide the same information about the future. We define the *equivalence class* C_z of a history suffix z to be the set of all history suffixes y such that $y \sim z$: $C_z = \{y \in \mathcal{X}^* | y \sim z\}$

These equivalence classes should be thought of as condensed versions of the past, in the sense that if we remember only which equivalence class C_z a history suffix z belongs to and we forget z itself, then we have lost no information about the future. More formally, suppose we define a random variable S which maps a history suffix to its equivalence class, $S(z) = C_z$. For all future words s, the definition of C_z tells us that the conditional probability $\mathbf{P}(s|z)$ is equal to the conditional probability $\mathbf{P}(s|C_z)$. Then the history suffix and the future are conditionally independent given S.

We can now use these equivalence classes to address the problem of simulating future output. Start with equivalence class C_w which contains our known history suffix w. C_w induces a distribution on x_0. As before, we choose a symbol $k \in \mathcal{X}$ according to this distribution. Next, we choose any history suffix y in C_w — we need not choose w — and append k, and shift it over by one to get $z = yk$, which is again a history suffix. From z, we get back to an equivalence class $C_z = S(z)$. Note that any choice of $y \in C_w$ yields the same C_z, because equality of distributions over the entire future space implies equality of distributions over the subspace of sequences which start with a given x_0.

As before, we have built a Hidden Markov Model presentation of our process. This time, however, we may have gained by doing so. It is possible for the states — that is, equivalence classes — to be recurrent. For example, in the fair coin process, all history suffixes are equivalent, so there is exactly one equivalence class, and it is visited after every time step. We may have infinitely many states, or we may have only finitely many, depending on the structure of the set of equivalence classes. Essentially, if we remember only the current equivalence class, we are keeping only the information from the past that is relevant to predicting the future. This recasting, as we will see, is fundamental. These conditional distributions on the future are the basis of the primary notion of "state" that we will be using. However, because there is more than one kind of object we will want to call a state, we will refer to the equivalence classes — or rather, the conditional future distributions they induce — as *process states*.

We can extend this idea to include conditional future distributions induced by conditioning on an entire history. When $\mathbf{x}^-$ is a history and $\mathbf{P}(\cdot|\mathbf{x}^-)$ is the conditional future distribution it induces, we can compare $\mathbf{P}(\cdot|\mathbf{x}^-)$ to a conditional future distribution $\mathbf{P}(\cdot|z)$ induced by a history suffix z. Considering conditional future distributions induced by histories introduces some complications, with which we will deal in the next section.

2.5 Process States

In this section, we will go through the preceding development again, rigorously. Those readers for whom the preceding discussion constitutes an adequate definition are advised to skim the this section rather than reading it closely.

First, we will develop a rigorous definition of $\mathbf{P}(\cdot|w)$, where w is a history suffix. We are only interested in conditioning on non-null history suffixes. Let $s_l = x_{-l} \ldots x_{-1}$

and let R_l be the set of all non-null length l history suffixes, $R_l = \{s_l | \mathbf{P}(s_l) > 0\}$. Then $R = \bigcup_{l=0}^{\infty} R_l$ is the set of all non-null history suffixes. Let w be any history suffix not in R, and let $B_w \in \mathbb{H}$ be the history event containing all histories which match w. For any future event $A \in \mathbb{F}$, we define the conditional probability $\mathbf{P}(A|w)$ by Bayes rule. That is,

$$\mathbf{P}(A|w) = \frac{\mathbf{P}(A \cap B_w)}{\mathbf{P}(B_w)}, \tag{2.8}$$

which we will write as

$$\mathbf{P}(A|w) = \frac{\mathbf{P}(A, w)}{\mathbf{P}(w)}. \tag{2.9}$$

Next, we will develop a definition for a conditional future distribution given a history, $\mathbf{P}(\cdot|\mathbf{x}^-)$. We will define $\mathbf{P}(\cdot|\mathbf{x}^-)$ in terms of the conditional distributions $\mathbf{P}(\cdot|w_l)$, where w_l is taken to be the length l suffix of the history $\mathbf{x}^-$. For every word s, we will define

$$\mathbf{P}(s|\mathbf{x}^-) = \lim_{l \to \infty} \mathbf{P}(s|w_l). \tag{2.10}$$

Note that if we let

$$1_A(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in A \\ 0 & \text{otherwise,} \end{cases} \tag{2.11}$$

$\mathbf{P}(A|w_l)$ is the conditional expectation $\mathbf{E}(1_A|\mathbb{H}_l)(\mathbf{x})$ for any sequence $\mathbf{x}$ which matches w. Thus the limit in equation 2.10 may be written $\lim_{l \to \infty} \mathbf{E}(1_A|\mathbb{H}_l)$, which converges almost surely to $\mathbf{E}(1_A|\mathbb{H})(\mathbf{x})$ by theorem B.2.3, which is a martingale convergence theorem. Thus we have

$$\lim_{l \to \infty} \mathbf{P}(\cdot|w_l) = \mathbf{P}(\cdot|\mathbf{x}^-) \tag{2.12}$$

for almost every $\mathbf{x}^-$. This is our definition of $\mathbf{P}(\cdot|\mathbf{x}^-)$.

Our next step is to define the set of histories on which we will condition when defining process states. We have just shown that this definition gives a well-defined conditional future distribution for almost all histories, and we will condition only on those histories.

Let us examine how $\mathbf{P}(\cdot|\mathbf{x}^-)$ can fail to be well-defined for a given history $\mathbf{x}^-$. For each of the countably many words s there may be a set of histories N_s — a set

of measure zero — on which $\lim_{l\to\infty} \mathbf{P}(s|w_l)$ does not converge. For example, if x^- has a suffix w_l with probability zero, then for all $m > l$, $\mathbf{P}(w_m) = 0$, and $\mathbf{P}\big(s|\mathrm{x}^-\big)$ is ill-defined for all s. The union $\mathcal{N}$ of the N_s is itself a null set. On its complement $\mathcal{X}^{\mathbb{Z}} \setminus \mathcal{N}$, which is thus a set of full measure, $\mathbf{P}\big(\cdot|\mathrm{x}^-\big)$ exists. We will call $\mathcal{N}$ the set of *bad histories*, and we will say that a history is *good* if it lies in $\mathcal{X}^{\mathbb{Z}} \setminus \mathcal{N}$.

Now we are ready to define process states.

Definition 2.5.1. A *process state* is a conditional future distribution which arises in conditioning on a non-null history suffix or a good history. That is, a process state is either a conditional future distribution $\mathbf{P}(\cdot|w)$ for some $w \in R$ or it is a conitional future distribution $\mathbf{P}\big(\cdot|\mathrm{x}^-\big)$ for some $\mathrm{x}^- \in \mathcal{X}^- \setminus \mathcal{N}$. Thus the *set S of all process states* for a given process is

$$S = \{\mathbf{P}(\cdot|w)|w \in R\} \cup \big\{\mathbf{P}\big(\cdot|\mathrm{x}^-\big)|\mathrm{x}^- \in \mathcal{X}^- \setminus \mathcal{N}\big\}. \tag{2.13}$$

In section 2.4, we developed the idea of process states in terms of equivalence classes of history suffixes. We have now developed a formal definition of process states, and we have defined a process state to be a conditional future distribution and not an equivalence class of history suffixes. This does not require changing how we think about process states, because there is a natural correspondence between equivalence classes as we described them in section 2.4 and conditional future distributions induced by non-null history suffixes. Recall that two history suffixes are said to be equivalent if they induce the same conditional future distribution. Thus every equivalence class has an associated conditional future distribution. At the same time, every process state is induced by at least one non-null history suffix or at least one good history. All history suffixes which induce a given process state are automatically equivalent to each other, as are all histories which induce a given process state. So every process state has an associated equivalence class of history suffixes, an equivalence class of histories, or both.

In addition, some of our terminology will refer to process states as if they were equivalence classes of history suffixes. In particular, we will say that a history or history suffix is a *member* or an *element* of the process state which it *induces*. In addition, we define the map $G : \mathcal{X}^- \to S$ which takes a non-null history to the process state it induces, as $G\big(\mathrm{x}^-\big) = \mathbf{P}\big(\cdot|\mathrm{x}^-\big)$

Definition 2.5.2. The *inducing set of histories* of a process state $\mathbf{A}$ is the set $G^{-1}(\mathbf{A})$ of all histories which induce $\mathbf{A}$.

$G^{-1}(\mathbf{A})$ may be empty; this occurs if $\mathbf{A}$ is induced only by history suffixes.

We will want to define a probability to $G^{-1}(\mathbf{A})$. As it happens, we do not know that $G^{-1}(\mathbf{A}) \times \mathcal{X}^+$ — which lies in $\mathcal{X}^{\mathbb{Z}}$ — is measurable. This next result tells us that, if it does not lie in the σ-field $\mathbb{H}$, $G^{-1}(\mathbf{A}) \times \mathcal{X}^+$ differs from a measurable set by a subset of a set of measure zero. Essentially, this means the we can reasonably assign a measure to it. From here on, we will do so without comment.

Proposition 2.5.3. For any process state $\mathbf{A}$, $G^{-1}(\mathbf{A}) \times \mathcal{X}^+$ can be written as the intersection of a measurable set and a set of full measure.

Proof. In this proof, we will be referring to truncations of both the history and the future. We will use l for history length and k for future length. Also, the reader is reminded that the process state $\mathbf{A}$ is a probability distribution on the future space, and so it makes sense to refer to $\mathbf{A}(w)$, the probability $\mathbf{A}$ assigns to a word w.

For every natural number l and word w, and for every history $\mathbf{x}^-$, define

$$f_l^w(\mathbf{x}^-) = \mathbf{E}(1_{A_w}|\mathbb{H}_l)(\mathbf{x}^-) = \mathbf{P}(w|x_{-l}\dots x_{-1}) \tag{2.14}$$

and

$$f^w(\mathbf{x}^-) = \mathbf{E}(1_{A_w}|\mathbb{H})(\mathbf{x}^-) = \mathbf{P}(w|\mathbf{x}^-). \tag{2.15}$$

By B.2.4, we know that $\lim_{l\to\infty} f_l^w = f^w$ almost surely. f_l^w is a measurable function, and so its limit f^w must be measurable.

If we have two process states $\mathbf{B}$ and $\mathbf{C}$, we will say that they are *k-equivalent* if, for all words w of length less than or equal to k, $\mathbf{B}(w) = \mathbf{C}(w)$. Let $\mathbf{A}$ be a process state and let $A_k(\mathbf{A})$ be the set of all histories $\mathbf{x}^-$ which induce process states $G(\mathbf{x}^-)$ that are k-equivalent to $\mathbf{A}$. In other language, that is,

$$A_k(\mathbf{A}) = \{\mathbf{x}^- | \text{ for all } w \text{ with } |w| \le k,\ \mathbf{P}(w|\mathbf{x}^-) = \mathbf{A}(w)\}. \tag{2.16}$$

That is,

$$\begin{aligned} A_k(\mathbf{A}) &= \bigcap_{w:\,|w|=k} \{\mathbf{x}^- | \mathbf{P}(w|\mathbf{x}^-) = \mathbf{A}(w)\} \\ &= \bigcap_{w:\,|w|=k} (f^w)^{-1}(\mathbf{A}(w)). \end{aligned} \tag{2.17}$$

$\{\mathbf{A}(w)\}$ is a measurable set and f^w is a measurable function, so the inverse image $(f^w)^{-1}(\mathbf{A})$ must be measurable. Thus $A_k(\mathbf{A})$ is a finite intersection of measurable set and must be measurable. Further, if we take

$$A(\mathbf{A}) = \bigcap_{k=1}^{\infty} A_k(\mathbf{A}), \tag{2.18}$$

$A(\mathbf{A})$ is a countable intersection of measurable sets, and so it is measurable.

Now, every history in $A(\mathbf{A}) \cap \left(\mathcal{X}^{\mathbb{Z}} \setminus \mathcal{N}\right)$ induces the process state $\mathbf{A}$, and no history which is not in $A(\mathbf{A})$ induces $\mathbf{A}$. Thus $A(\mathbf{A}) \cap \left(\mathcal{X}^{\mathbb{Z}} \setminus \mathcal{N}\right) = G^{-1}(\mathbf{A})$. Note that the $\mathcal{X}^{\mathbb{Z}} \setminus \mathcal{N}$ has full measure, so we are done. ■

The processes addressed in this section and the previous one output bi-infinite sequences, but the idea of a process state is no less relevant to processes with finite or semi-infinite output. The essential idea is that a state is a prediction of, or distribution on, the future reached by some knowledge of the past.

Before going on to the next section, let us look an example. This process has elements of both of the examples from the previous section. The alphabet is $\mathcal{X} = \{0, 1\}$. We first define a nonstationary probability measure $\mathbf{Q}$ on $\left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}\right)$, which generates sequences $\mathbf{y} = (\ldots y_{-1} y_0 y_{1\ldots}) \in \mathcal{X}^{\mathbb{Z}}$ such that

1. if t is even, $y_t = 1$, and
2. if t is odd, then y_t is either 0 or 1 with probability $\frac{1}{2}$ each.

Now, we define the process $\mathcal{P} = \left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$ by $\mathbf{P} = \frac{1}{2}(\mathbf{Q} + T(\mathbf{Q}))$, where T is the shift map. In words, our process consists of alternating 1s and coin flips, and the coin flips are equally likely in either even or odd positions. We will not prove that this is a process, nor will we do so for the examples in the next section. (In chapter 3, we will develop a systematic approach to calculating probabilities of words, which will make it practical to present such proofs). The process $\mathcal{P}$ is called the *band-merging process* for historical reasons[15].

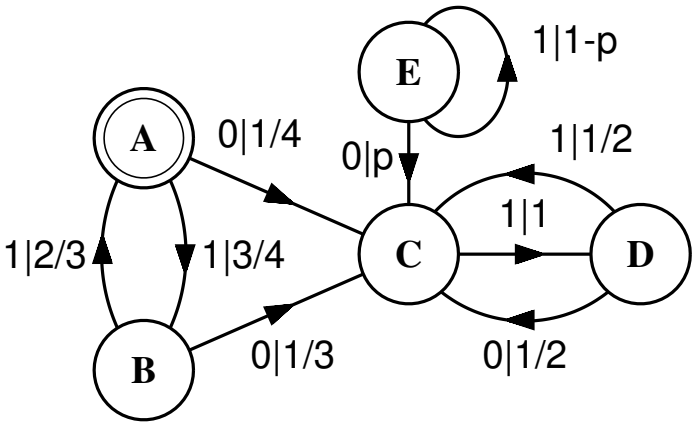

Fig. 2.1 Process state graph representation of the band-merging process. **A**, **B**, **C**, **D**, and **E** are the process states. **A** is the start state (induced by λ). The labels on the edges consist of a symbol followed by a vertical bar and a transition probability. Here p= $1 - \frac{\sqrt{2}}{2}$.

Figure 2.1 is a process state graph representation of the band-merging process. The small circles ($\mathbf{A}$, $\mathbf{B}$, C, $\mathbf{D}$, and $\mathbf{E}$) represent process states. The double circle for state $\mathbf{A}$ indicates that it is the process state induced by λ, which is called the *start state*. The process is said to be *in* a state if the known history or history suffix induces that state. The edges represent possible transitions between states and the labels are of the form *symbol|transition probability*. For instance, the edge from state $\mathbf{A}$ to state $\mathbf{B}$, labeled with "1|3/4" indicates that if the process is in state $\mathbf{A}$, then with probability $\frac{3}{4}$ it will emit a 1, after which it will be in state $\mathbf{B}$.

The band-merging process has five process states. State $\mathbf{A}$ is induced by any history suffix consisting of an even number of 1s, and state $\mathbf{B}$ is induced by any history suffix consisting of an odd number of 1s. States C and $\mathbf{D}$ correspond to any history or history suffix ending with a 0 followed by an even or odd number of 1s, respectively, and state $\mathbf{E}$ is induced only by the history consisting entirely of 1s. State $\mathbf{E}$ is an interesting example of a state which we could ignore because it is irrelevant to the study of the process — it induced only by a single history which has mass zero — but which is well defined. Such states are said to be *elusive*. We will discuss elusive states more in the next section, after which we will usually ignore them.

2.6 Transient and Recurrent States

In section 2.5, we defined a process state to be a distribution on the future which

is induced by a finite length history suffix or a semi-infinite history. This means that there may be some process states which are induced by history suffixes and not by any histories, some states which are induced by histories and not by history suffixes, and some states that are induced by histories and history suffixes. In this section we classify process states by the history objects which induce them.

First, we will define terms for the above distinctions.

Definition 2.6.1. A process state is *infinitely preceded* if it is induced by at least one good history.

Definition 2.6.2. A process state is *reachable* if it is induced by at least one history suffix w with $\mathbf{P}(w) > 0$.

In addition, we need to define one more property. For a process state $\mathbf{A} \in S$, consider $G^{-1}(\mathbf{A})$, the set of histories which induce $\mathbf{A}$. We know that we can assign this set a measure, for which we will now use the shorthand $\mathbf{P}(\mathbf{A}) = \mathbf{P}\big(G^{-1}(\mathbf{A})\big)$, where $\mathbf{P}^-$ is the measure on the history space. That is, if we have seen a history s, $\mathbf{P}(\mathbf{A})$ is the probability that s induces process state $\mathbf{A}$.

Definition 2.6.3. A process state is *recurrent* if $\mathbf{P}(\mathbf{A}) > 0$.

The term *positive recurrent* is often used for this concept [13]. We have chosen to use *recurrent* for brevity and because null recurrence does not happen in the systems of interest here.

Now, for any process state, we may ask three questions. Is it reachable? Is it infinitely preceded? And, is it recurrent? There are eight triples of answers, of which three are impossible and five are observed.

Proposition 2.6.4.

1. Every unreachable process state is infinitely preceded.
2. Every recurrent process state is infinitely preceded.

This proposition asserts that the following three kinds of states do not occur:

- unreachable recurrent states which are not infinitely preceded,
- reachable recurrent states which are not infinitely preceded, and

- unreachable states which are neither recurrent nor infinitely preceded.

Proof. Statement 1 is automatic from the definition of a process state, since every process state is induced by at least one history or history suffix. Statement 2 follows from definitions 2.6.1 and 2.6.3. Let $\mathbf{A}$ be a recurrent process state. Then $G^{-1}(\mathbf{A})$ has positive measure and thus is nonempty. Since $G^{-1}(\mathbf{A})$ contains only good histories, then it contains at least one good history which induces $\mathbf{A}$. ■

Definition 2.6.5. A process state is *transient* if it is reachable and not recurrent. A transient state is said to be *strictly transient* if it is not infinitely preceded.

Definition 2.6.6. A process state is *elusive* if it is unreachable and not recurrent.

Unlike the other kinds of states we have discussed, elusive states can often be ignored. Whereas every reachable state can be induced by at least one word of positive probability and a recurrent state can be induced by a set of histories of positive probability, an elusive state can only be induced by events of zero probability. Thus any countable set of elusive states can be ignored without changing the process' probability measure. We will usually omit elusive states for brevity. However, in some processes the existence and structure of the elusive states is implied by the recurrent states and in others the set of all elusive states is uncountable and has positive measure, so we cannot forget them completely.

Proposition 2.6.4 established that there are five classes of states. This, together with definitions 2.6.5 and 2.6.6, gives us names for them. Table 2.1 summarizes the relevant information about each type. In figure 2.2, we have examples of all types except unreachable recurrent states. In order to present processes which have such states in a reasonable form, we will need to develope more machinery for presenting processes.

Process state type	Reachable	Infinitely Preceded	Recurrent
Strictly Transient	yes	no	no
Infinitely Preceded Transient	yes	yes	no
Reachable Recurrent	yes	yes	yes
Unreachable Recurrent	no	yes	yes
Elusive	no	yes	no

Table 2.1 Summary of process state types.

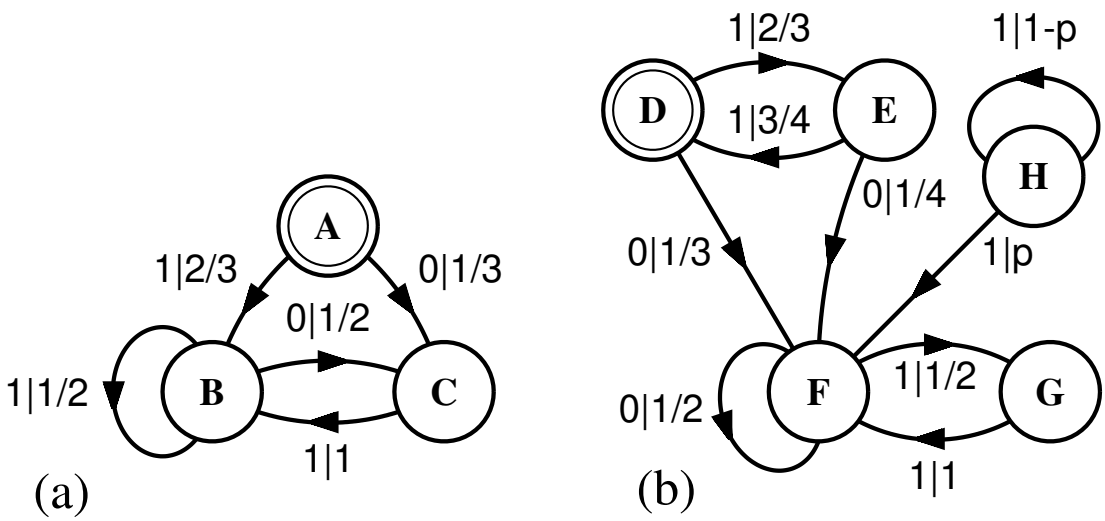

Fig. 2.2 Process state graph representations of two processes. (a) The golden mean process. The process state **A** is reachable and neither infinitely preceded nor recurrent. States **B** and **C** are reachable, infinitely preceded and recurrent. (b) The even process. Here $p = 1 - \frac{\sqrt{2}}{2}$. States **D** and **E** are reachable and infinitely preceded but not recurrent, **F** and **G** are reachable, infinitely preceded and recurrent, and **H** is unreachable, infinitely preceded, and not recurrent. For both of these processes, there may be additional ill-defined states resulting from conditioning on histories in some measure-zero bad set.

2.7 Synchronization

Suppose we are watching the output of the even system, and the history suffix we have seen contains all 1s. This means that our state of knowledge about the future of the process is described by either process state $\mathbf{D}$ or $\mathbf{E}$. It is possible that another observer, who has been watching longer, knows more about the future than we do. In contrast, if the history suffix we have seen contains a 0, then this is not the case. An observer may

have more historical information than we do, but this extra information is irrelevant to the future. In this case we say that we are *synchronized* to the machine.

Definition 2.7.1. A non-null history suffix w is a *synchronizing word* if all good histories and all non-null history suffixes which end in w induce the same process state.

Proposition 2.7.2. If w is a synchronizing word, then w induces a reachable recurrent process state.

Proof. Let $\mathbf{A}$ be the process state induced by w. $\mathbf{A}$ is induced by a history suffix, so it is reachable. Note that w is itself a history suffix that ends in w, so $\mathbf{A}$ is the process state induced by all histories and history suffixes which match w. Further, we know that w is not a null word, and that the set of histories which matches it is a subset of $G^{-1}(\mathbf{A})$. So we have

$$\begin{aligned} \mathbf{P}(\mathbf{A}) &\geq \mathbf{P}(\text{all histories which match } w) \\ &\geq \mathbf{P}(w) > 0. \end{aligned} \tag{2.19}$$

Thus we conclude that $\mathbf{A}$ is recurrent. ■

The converse of Proposition 2.7.2 does not hold. There are process in which nonsynchronizing words induce recurrent states. Also, some processes have no reachable recurrent states, and hence no synchronizing words. These processes either have unreachable recurrent states or have uncountably many elusive states. We will see examples in section 3.5.

3 Hidden Markov Models

In the last chapter, we talked entirely about distributions on sequence space. Although this viewpoint will be necessary for some of our results, processes as we have defined them are not very tractable or structured. A distribution on an infinite set need not lend itself to a finite description, let alone a brief one. In order to do anything concrete, we will need another set of definitions. These are the traditional definitions used in the study of Hidden Markov Models [3,16,17]. In this chapter, we define Hidden Markov Models and then study how they represent processes. We will look at how to represent the process states of a process defined by an HMM. And we will conclude the chapter with a result on the structure of these processes' sets of process states.

The material in sections 3.1, 3.2, and 3.3 is fairly standard in the literature on HMMs, appearing in such works as [3,16]. The contents of section 3.4 has probably all been deduced before. Mixed states, for example, appear in [1], although not with that name. What is new is stating this material in terms of process states. And the material in section 3.6 is entirely the author's though some similar results are known.

We will be careful to keep clear the distinction between the process and the way it's presented to us. By *process*, we will always mean a stationary distribution on a sequence space. When we refer to a finite specification of a process, such as a Hidden Markov Model, we will call it a presentation of a process, or simply a *presentation*.

3.1 Notation for Markov Chains

Before we start on Hidden Markov Models, we will define a Markov Chain. This definition and the following discussion are not intended to be complete; rather, they are intended to introduce the reader to the notation we will be using.

Definition 3.1.1. An n-state *Markov Chain* (MC) is a triple (V, P, π), where V is a finite set of size n, P is an $n \times n$ matrix, and π is a length n row vector, such that

(i) Each row of P has sum one,
(ii) $\sum_i \pi_i = 1$, and
(iii) $\pi P = \pi$.

Elements of V are called states, P is called the transition matrix, and π is called a stationary distribution over the states V.

Note that this definition requires a Markov Chain to have finitely many states. At times in the following, we will discuss both countably and uncountably infinite state Markov Chains, but we will not define them rigorously.

If we let $\mathbb{V}$ be the σ-field defined by the cylinder sets on $V^{\mathbb{Z}}$, then $\left(V^{\mathbb{Z}}, \mathbb{V}\right)$ is a measurable space. We define a distribution $\overline{\mathbf{P}}$ as follows: if $v = v_0 v_1 \dots v_{l-1}$, with all $v_i \in V$, we define

$$\overline{\mathbf{P}}(v) = \pi_{v_0} P_{v_0 v_1} \dots P_{v_{l-2} v_{l-1}}, \tag{3.1}$$

and we define $\overline{\mathbf{P}}(\lambda) = 1$. Equation 2.2 is satisfied trivially. We will verify equation 2.3 and invoke theorem B.1.1 to show that $\overline{\mathbf{P}}$ is a stationary probability distribution. If $z \in V$, then

$$\overline{\mathbf{P}}(\lambda z) = \overline{\mathbf{P}}(z\lambda) = \overline{\mathbf{P}}(z) = \pi_z. \tag{3.2}$$

Thus,

$$\sum_{z \in V} \overline{\mathbf{P}}(z\lambda) = \sum_{z \in V} \overline{\mathbf{P}}(\lambda z) = \sum_{z \in V} \pi_z = 1 = \overline{\mathbf{P}}(\lambda). \tag{3.3}$$

If $v = v_0 \dots v_{l-1}$, we have$\overline{\mathbf{P}}(vz) = \overline{\mathbf{P}}(v) P_{v_{l-1} z}$. Thus

$$\sum_{z \in V} \overline{\mathbf{P}}(vz) = \overline{\mathbf{P}}(v) \sum_{z \in V} P_{v_{l-1} z} = \overline{\mathbf{P}}(v) \cdot 1 = \overline{\mathbf{P}}(v). \tag{3.4}$$

On the other hand,

$$\overline{\mathbf{P}}(zv) = \pi_z P_{z v_0} P_{v_0 v_1} \dots P_{v_{l-2} v_{l-1}}, \tag{3.5}$$

so

$$\sum_{z \in V} \overline{\mathbf{P}}(zv) = \left(\sum_{z \in V} \pi_z P_{z v_0} \right) P_{v_0 v_1} \dots P_{v_{l-2} v_{l-1}}. \tag{3.6}$$

But $\sum_{z \in V} \pi_z P_{z v_0}$ is the v_0 coordinate of πP and $\pi P = \pi$, so

$$\sum_{z \in V} \pi_z P_{z v_0} = \pi_{v_0} \tag{3.7}$$

and

$$\sum_{z \in V} \overline{\mathbf{P}}(zv) = \overline{\mathbf{P}}(v). \tag{3.8}$$

Thus the Markov Chain (V, P, π) defines a process $\left(V^{\mathbb{Z}}, \mathbb{V}, \overline{\mathbf{P}}\right)$.

We conclude this section with at definition which we will need in section 3.2.

Definition 3.1.2. If (V, P, π) is a Markov Chain and $C \subset V$, we say that C is a *recurrent component* of the Markov Chain if:

(i) For all $u, v \in C$, there exists an integer $k > 0$ such that $P^k_{uv} > 0$, and
(ii) For all $u \in C$, for all $v \in V \setminus C$, and for all integers $k > 0$, we have $P^k_{uv} = 0$.

Here, P^k means the kth power of the matrix P.

Definition 3.1.3. A finite Markov Chain is *reducible* if it has more than one recurrent component.

If a Markov Chain is reducible, it is often appropriate to think of it as two or more separate Markov Chains. A Markov Chain which has exactly one recurrent component is said to be *irreducible*.

3.2 Hidden Markov Models

In this section, we will give a definition of a Hidden Markov Model (HMM), and we will show how an HMM specifies a process.

A Hidden Markov Model is a Markov Chain with an associated output mechanism which takes either states or transitions between states to either symbols or distributions on symbols. We will refer to the Markov Chain as the *underlying Markov Chain* of the HMM. We will calculate exclusively with finite presentations — those in which the Markov Chain has finitely many states. However, we will, at times, consider infinite presentations.

Hidden Markov Models appear in the literature in several forms, the most frequent being Functions of a Markov Chain[1] and State-output Hidden Markov Models[16]. These forms are equivalent in the sense that for any HMM in one of these forms, there is an HMM in each of the other forms which defines the same process. The HMMs in this work will be Edge-output Hidden Markov Models, the elements of which are the set of states, the set of symbols, a stationary distribution on those states, and, for each state, a joint distribution on symbols and next states. The following definition formalizes this idea.

Definition 3.2.1. A *Hidden Markov Model (HMM)*is a quadruple $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$, where V and $\mathcal{X}$ are finite sets of sizes $n = |V|$ and $m = |\mathcal{X}|$, $\left\{T^k\right\} = \left\{T^k | k = 0, \ldots, m-1\right\}$ is a set of $n \times n$ matrices, and π is a probability vector with length n. The matrices $\left\{T^k\right\}$ must satisfy

1. For all i such that $0 \leq i \leq n-1$

$$\sum_{j,k} T^i_{ij} = 1, \tag{3.9}$$

2. and for all i,j such that $0 \leq i, j \leq n-1$ and $0 \leq k \leq m-1$,

$$T^k_{ij} \geq 0. \tag{3.10}$$

Finally, π must satisfy

$$\pi_j = \sum_{i,k} \pi_i T^k_{ij}. \tag{3.11}$$

The *underlying Markov Chain* of a Hidden Markov Model is a the Markov Chain $\left(V, \sum_k T^k, \pi\right)$.

Elements of V, called *presentation states*, are the states of the underlying Markov Chain. Elements of $\mathcal{X}$ are called *symbols*, as in chapter 2. Unless we have reason to do otherwise, we will use $V = \{0, 1, \ldots, n-1\}$ or $V = \{A, B, \ldots\}$ and $\mathcal{X} = \{0, 1, \ldots, m-1\}$. The $\left\{T^k\right\}$, called the *joint matrices*, represent a set of joint distributions on next states $j \in V$ and output symbols $k \in \mathcal{X}$ in the following way. If $i, j \in V$ and $k \in \mathcal{X}$ and the Markov Chain is in state i, then the probability that the next symbol emitted will be k and the next state will be j is

$$\mathbf{P}(j, k|i) = T^k_{i,j}. \tag{3.12}$$

The last element of the quadruple is π, which is a *stationary distribution*. Most definitions of HMMs found in the literature have an initial distribution instead of a stationary distribution. The difference is that an initial distribution may be any distribution over the states, whereas the stationary distribution is constrained to satisfy equation 3.11. Using a stationary distribution here makes the resulting process stationary.

If the underlying Markov Chain has a single recurrent component, then π is uniquely determined by the joint matrices. If, however, the underlying Markov Chain has more

than one recurrent component, then π is only partially determined. Choosing a stationary distribution is then tatamount to choosing a distribution over the components.

In addition, we will define a few auxiliary matrices. The *transition matrix* P of a Hidden Markov Model is defined by

$$P_{ij} = \sum_k T_{ij}^k. \tag{3.13}$$

The output matrix B is an $n \times m$ matrix such that B_{jk} gives the probability of emitting the symbol $k \in \mathcal{X}$ while in the state $j \in V$. B is define by

$$B_{ki} = \sum_j T_{ij}^k. \tag{3.14}$$

The conditions imposed on the joint matrices ensure that P and B are stochastic matrices, that is, their rows sums are all equal to 1. Also, we have $\pi P = \pi$, and we can write the underlying Markov Chain of the HMM as (V, P, π).

A warning to readers familiar with state-output Hidden Markov Models defined in terms of transition and output matrices — our choice of notation may be misleading to your intuition. The auxiliary matrices P and B are **not** always sufficient to recover the joint matrices $\left\{T^k\right\}$. For example, if we start with a state-output HMM, the joint matrices can be constructed as $T_{ij}^k = P_{ij}B_{jk}$, and equations 3.13 and 3.14 will be satisfied. That is, if we compute the right hand sides of 3.13 and 3.14, we will recover our original transition and output matrices. But, if we start with a set of joint matrices, compute the transition and output matrices by equations 3.13 and 3.14, and then compute $P_{ij}B_{jk}$, the result need not be the joint matrices. Doubtful readers are encouraged to perform the calculations themselves on the two-state, two symbol process with joint matrices $T^0 = \begin{pmatrix} 0 & 0 \\ 1/2 & 0 \end{pmatrix}$ and $T^1 = \begin{pmatrix} 1/2 & 1/2 \\ 0 & 1/2 \end{pmatrix}$.

In general, a state-output HMM may be built from an edge-output HMM, but the state-output HMM may need to have a greater number of states, because edge-output HMMs have more degrees of freedom per state than state-output HMMs. Given an edge-output HMM $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$, we can construct an equivalent state-output HMM with set of states $U = V \times V$ as follows: if $a, b, c, d \in V$, then we have $(a, b), (c, d) \in U$. Let

$$P_{(a,b),(c,d)} = \begin{cases} \sum\limits_{k \in \mathcal{X}} T_{cd}^k & b = c \\ 0 & b \neq c, \end{cases} \tag{3.15}$$

and let

$$B_{(a,b),k} = \frac{T^k_{a,b}}{\sum\limits_{l \in \mathcal{X}} T^l_{a,b}}. \tag{3.16}$$

3.3 HMMs as Processes

An HMM presentation defines a process. That is, $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ determines a probability distribution $\mathbf{P}$ and thus a process $\mathcal{P} = \left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$. Let us see how this works.

First, we suppose that the presentation's underlying Markov Chain is in the state $i \in V$. Let k be a symbol and $j \in V$ be a presentation state. We want to know $\mathbf{P}(k|i)$, the probability that the next symbol will be k, and $\mathbf{P}(j|i,k)$, the probability that next presentation state will be j if the next symbol is k. These are straightforward to calculate from the presentation.

$$\mathbf{P}(k|i) = \sum_j \mathbf{P}(j,k|i) = \sum_j T^k_{ij} \tag{3.17}$$

$$\mathbf{P}(j|i,k) = \frac{\mathbf{P}(j,k|i)}{\mathbf{P}(k|i)} = \frac{T^k_{ij}}{\sum\limits_l T^k_{il}} \tag{3.18}$$

Next, instead of assuming that the current presentation state is i, that is, $\mathbf{P}(i) = 1$, we assume that it has distribution μ. To calculate the analogous quantities, $\mathbf{P}(k|\mu)$ and $\mathbf{P}(j|k,\mu)$, we start by calculating $\mathbf{P}(j,k|\mu)$. After that, the answers are essentially the same as above.

$$\mathbf{P}(j,k|\mu) = \sum_i \mu_i \mathbf{P}(j,k|i) = \left(\mu T^k\right)_j \tag{3.19}$$

$$\mathbf{P}(k|\mu) = \sum_j \mathbf{P}(j,k|\mu) = \sum_j \left(\mu T^k\right)_j \tag{3.20}$$

$$\mathbf{P}(j|\mu,k) = \frac{\mathbf{P}(j,k|\mu)}{\mathbf{P}(k|\mu)} = \frac{\left(\mu T^k\right)_j}{\sum\limits_j \left(\mu T^k\right)_j} \tag{3.21}$$

If we denote the column vector $(1,\ldots,1)^T$ by $\vec{1}$, we can write $\mathbf{P}(k|\mu) = \mu T^k \vec{1}$.

Now, define a map C_k, which takes distributions μ on the states V to distributions on V, by

$$C_k(\mu) = \mu T^k / \mu T^k \vec{1}. \tag{3.22}$$

We then have $\mathbf{P}(j|\mu, k) = (C_k(\mu))_j$. We think of μ as representing our state of knowledge about the internal state of the process. The C_k should be thought of as update maps: they take a distribution μ at one time and update it to reflect the passage of time and the latest observation k.

Having addressed single symbols, we are ready to address words. We begin with a word w of length two, $w = w_0 w_1$. $\mathbf{P}(w|\mu)$ factors to $\mathbf{P}(w_0|\mu) \cdot \mathbf{P}(w_1|w_0, \mu)$. The first of these terms is a case we have just treated in 3.20. For the second, if we update μ to $C_{w_0}(\mu)$, it reduces to the same case: $\mathbf{P}(w_1|w_0, \mu) = \mathbf{P}(w_1|C_{w_0}(\mu))$. We now expand and simplify,

$$\begin{aligned}
\mathbf{P}(w|\mu) &= \mathbf{P}(w_0|\mu) \cdot \mathbf{P}(w_1|C_{w_0}(\mu)) \\
&= \left(\mu T^{w_0} \vec{1}\right) \left(C_{w_0}(\mu) T^{w_1} \vec{1}\right) \\
&= \left(\mu T^{w_0} \vec{1}\right) \left(\frac{\mu T^{w_0}}{\mu T^{w_0} \vec{1}}\right) \cdot \left(T^{w_1} \vec{1}\right) \\
&= \mu T^{w_0} T^{w_1} \vec{1}
\end{aligned} \tag{3.23}$$

By similar manipulations, we have

$$\begin{aligned}
(C_{w_1} \circ C_{w_0})(\mu) &= C_{w_1} \left(\frac{\mu T^{w_0}}{\mu T^{w_0} \vec{1}}\right) \\
&= \frac{\left(\mu T^{w_0} / \mu T^{w_0} \vec{1}\right) T^{w_1}}{\left(\mu T^{w_0} / \mu T^{w_0} \vec{1}\right) T^{w_1} \vec{1}} \\
&= \frac{\mu T^{w_0} T^{w_1}}{\mu T^{w_0} T^{w_1} \vec{1}}
\end{aligned} \tag{3.24}$$

This extends to words of arbitrary length. If w is a word of length l, then $\mathbf{P}(w|\mu) = \mu T^{w_0} T^{w_1} \ldots T^{w_{l-1}} \vec{1}$ and the updated distribution over the presentation states is

$$\left(C_{w_{l-1}} \circ \ldots \circ C_{w_0} \ldots\right)(\mu) = \frac{\mu T^{w_0} \ldots T^{w_{l-1}}}{\mu T^{w_0} \ldots T^{w_{l-1}} \vec{1}} \tag{3.25}$$

Now, if we use the stationary distribution π in place of the arbitrary distribution μ, we have a stationary process.

Lemma 3.3.1. There is a unique stationary process $\mathcal{P} = \left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$ such that for all words $w = w_0 \dots w_{l-1}$,

$$\mathbf{P}(w) = \pi T^{w_0} \dots T^{w_{l-1}} \vec{1}. \tag{3.26}$$

Proof. We will simply verify equations 2.2 and 2.3 and invoke theorem B.1.1. First, $\mathbf{P}(\lambda) = \pi \vec{1} = \sum_i \pi_i = 1$. This takes care of 2.2. Second, for $z \in \mathcal{X}$,

$$\mathbf{P}(wz) = \pi T^{w_0} \dots T^{w_{l-1}} T^z \vec{1}. \tag{3.27}$$

Thus,

$$\sum_{z \in \mathcal{X}} \mathbf{P}(wz) = \pi T^{w_0} \dots T^{w_{l-1}} \left(\sum_{z \in \mathcal{X}} T^z \right) \vec{1}. \tag{3.28}$$

But the rows of $\sum_z T^z$ sum to one, so $\left(\sum_z T^z \right) \vec{1} = \vec{1}$. Hence,

$$\sum_{z \in \mathcal{X}} \mathbf{P}(wz) = \pi T^{w_0} \dots T^{w_{l-1}} \vec{1} = \mathbf{P}(w). \tag{3.29}$$

Similarly,

$$\sum_{z \in \mathcal{X}} \mathbf{P}(zw) = \pi \left(\sum_{z \in \mathcal{X}} T^z \right) T^{w_0} \dots T^{w_{l-1}} \vec{1}. \tag{3.30}$$

But $\pi \left(\sum_z T^z \right) = \pi$, so

$$\sum_{z \in \mathcal{X}} \mathbf{P}(zw) = \pi T^{w_0} \dots T^{w_{l-1}} \vec{1} = \mathbf{P}(w). \tag{3.31}$$

Thus the hypotheses of theorem B.1.1 are satisfied.■

Definition 3.3.2. The *process defined by an HMM presentation* $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ is the process $\mathcal{P} = \left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$ which assigns the probability $\mathbf{P}(w) = \mathbf{P}(w|\pi)$ for any word w of symbols in $\mathcal{X}$.

Over the course of this dissertation, we will be doing many calculations containing expressions of the form $T^{w_0} \dots T^{w_{l-1}}$. In order to shorten these expressions, we will define the matrix T^w for any word w. If $w = w_0 \dots w_{l-1}$, then we define $T^w = T^{w_0} \dots T^{w_{l-1}}$. For the empty word λ, we define $T^\lambda = I$. Thus, for any pair of words, w and z, we have $T^{wz} = T^w T^z$. In this notation, the probability of a word w is $\mathbf{P}(w) = \pi T^w \vec{1}$.

As we have seen, matrix presentations are convenient for calculation. Intuitive interpretation, on the other hand, is often easier with some other forms of presentation. For this reason, we will introduce a new form of presentation, which we will call a *labeled directed graph*. Examples of labeled directed graph presentations may be found in section 3.5. It is worth noting that, while labeled directed graph presentations are often quite clear, they become less intelligible as the number of edges per state increases. For example, compare figures 3.3 and 3.6 on pages 44 and 46.

We have already seen process state graph presentations in sections 2.5 and 2.6; the presentations we define here are related, but distinct. Here the nodes of a labeled directed graph represent presentation states, and not process states as was the case before. Process state graphs are deterministic — that is, they cannot have two or more edges leaving the same state labeled with the same symbol. Labeled directed graphs do not have this restriction.

A labeled directed graph is a directed graph in which the nodes represent presentation states and the edges represent possible transitions. Each edge is labeled with a symbol and a probability. An edge from state i to state j which is labeled with $k|p$ corresponds to an entry in a joint matrix: $T_{ij}^k = p$. That is, k is a symbol and p is a probability, and whenever the labeled directed graph is in state i, it has probability p of following this edge, and if it does so it will output a k and go to state j. We can translate an HMM into a labeled directed graph by drawing a node for each state of the HMM and an edge for each nonzero T_{ij}^k. Similarly, we can usually translate a labeled directed graph into an HMM. We let V be the set of nodes in the graph and $\mathcal{X}$ be the set of all symbols which are appear on the edges of the graph. For each $i, j \in V$ and $k \in \mathcal{X}$, if there is an edge from state i to state j which is labeled with $k|p$ for some p, then we set $T_{ij}^k = p$, and otherwise we set $T_{ij}^k = 0$. The one piece of an HMM which is not present in a labeled directed graph is the stationary distribution π. If there is only one possible stationary distribution for the set of joint matrices, then the labeled directed graph is a complete presentation, and it defines a process. If there is more than one — that is, if the underlying Markov Chain has several recurrent components or is periodic[13] — then the labeled directed graph does not specify a process.

A given process may have many presentations, and determining whether or not two presentations describe the same process is nontrivial [7,2]. For example, the

$$V = \{0,1,2\},\ \mathcal{X} = \{0,1\},\ \pi_A = (\tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{2})$$
$$T^0 = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},\ T^1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$$

Fig. 3.1 Process "simple nondeterministic source," presentation A.

$$W = \{0,1\},\ \mathcal{X} = \{0,1\},\ \pi_B = (\tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{2})$$
$$U^0 = \begin{pmatrix} 0 & 0 \\ \frac{1}{2} & 0 \end{pmatrix},\ U^1 = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} \end{pmatrix}$$

Fig. 3.2 Process "simple nondeterministic source," presentation B.

presentations in figures 3.1 and 3.2 define the same process. To show that presentations A and B are equivalent, it is sufficient to show that, for every finite word w, $\pi_A T^w \vec{1} = \pi_B U^w \vec{1}$. In this case, it can be done by induction. However such proofs are at best computationally messy and are not very illuminating. In section 4.3, we will develop a systematic approach to equivalence of presentations. We will prove that A and B are equivalent there.

Not all processes can be presented as finite HMMs. For example, consider the *modified nested parentheses process*[18], a process with the alphabet of `(` , `)`, and `!`. (The term *modified* refers to the presence of the `!` symbol.) One way to represent this process is as a single presentation state and a counter which holds a nonnegative integer. If the counter is set to zero, then with probability $\frac{1}{3}$, the machine outputs a `(` and sets the counter to one, and with probability $\frac{2}{3}$ it outputs a `!` and leaves the counter at zero. If the counter is not set to zero, then with probability $\frac{2}{3}$ the machine outputs a `)` and decrements the counter and with probability $\frac{1}{3}$ it outputs a `(` and increments the counter. If the initial value of the counter is drawn from the appropriate distribution, this description defines a (stationary) process. This process always outputs balanced strings of parentheses between any consecutive pair of `!` symbols, and there is no upper bound to the number of levels of nesting. We will prove in section 3.6 that there is no HMM presentation for this process.

Simply stated, in this section we have shown how to get a process from an HMM.

But consider the inverse problem — suppose we have a process, and we want an HMM presentation for it. Because a process can have more than one HMM presentation, we cannot expect a unique answer. And, as the modified nested parentheses process illustrates, we cannot always expect any answer at all. This is a form of the problem of HMM reconstruction, and nothing we have seen here so far suggests a way of approaching it.

Finally, we can define the class of processes which are the subject of this dissertation, stochastic finite automata. A *stochastic finite automaton* (SFA) is a process which has a finite HMM presentation. In section 3.6, we will give a necessary condition for a process to be an SFA. Notably, this condition will, among other things, suggest an approach to HMM reconstruction.

3.4 Mixed States

In section 2.5, we defined process states in rather abstract terms, and in section 3.2 we described HMMs in more concrete terms. In this section, we will bring these threads together and discuss the process states of processes defined by HMM presentations.

Recall that a process state is a conditional future distribution which arises when we condition on a history or a history suffix. Suppose we have a process $\mathcal{P} = \left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$ defined by an HMM presentation $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$. What are the process states for this process?

There are some presentations for which the process states coincide with the presentation states. Such presentations are necessarily *deterministic*. This means that, for any given presentation state $i \in V$ and symbol $k \in \mathcal{X}$ there is at most one presentation state $j \in V$ such that the transition from i to j with symbol k is possible, $T^k_{ij} \neq 0$. If a process has a finite deterministic presentation then it is called a *Stochastic Deterministic Finite Automaton* (SDFA). In this case, the presentation states and process states are similar though they may not coincide. SDFAs are an important class of processes; see [19]. However, typical HMMs are not deterministic and the processes they represent are not SDFAs. It is this case which this section addresses.

We will begin with reachable states, those which result from conditioning on a finite history suffix. Suppose s is a history suffix and w is a next word. We have

$$\mathbf{P}(w|s) = \frac{\mathbf{P}(sw)}{\mathbf{P}(s)} = \frac{\pi T^s T^w \vec{1}}{\pi T^s \vec{1}}. \tag{3.34}$$

(If $\mathbf{P}(s) = 0$, then $\mathbf{P}(w|s)$ is not well defined. We will ignore such s throughout this section.) Since the conditional distribution $\mathbf{P}(\cdot|s)$ is the object we are interested in and w is the argument it takes, we will rewrite this as

$$\mathbf{P}(w|s) = \mathbf{P}(\cdot|s)(w) = \frac{\pi T^s}{\pi T^s \vec{1}} T^w \vec{1} \tag{3.35}$$

Here, $\mathbf{P}(\cdot|s)$ shows up as $\pi T^s / \pi T^s \vec{1}$, which is a distribution on the presentation states.

In fact, distributions over the presentation states are close to being process states. If μ is such a distribution, then $\mathbf{P}(\cdot|\mu)$ is the conditional future distribution given the measure μ, defined by $\mathbf{P}(w|\mu) = \mu T^w \vec{1}$. We will show below that all process states can be represented in this way. If two different history suffixes, s and $\overline{s}$, define the same distribution over presentation states — $\pi T^s / \pi T^s \vec{1} = \pi T^{\overline{s}} / \pi T^{\overline{s}} \vec{1}$ — then clearly $\mathbf{P}(\cdot|s) = \mathbf{P}(\cdot|\overline{s})$, so s and $\overline{s}$ lead to the same process state.

Before we proceed, we will introduce a notational convenience. When we have a row vector μ, we often need to *normalize* it, that is, scale it so that the sum of its components is 1. We have been writing the normalization of μ as $\frac{\mu}{\mu \vec{1}}$. We now define N, the *normalizing function*, which takes row vectors to row vectors, by

$$N(\mu) = \frac{\mu}{\mu \vec{1}}. \tag{3.36}$$

With this, we can write $N(\pi T^w)$ instead of $\pi T^w / \pi T^w \vec{1}$.

Definition 3.4.1. A *mixed state* of a presentation is a distribution over the presentation states.

(The name *mixed state* comes from thinking of mixed states as "mixtures" of presentation states. This is similar to the use of "mixed state" in quantum mechanics. It should be noted that Fraser and Dimitriadis have use the term "mixed state" in connection with HMMs to mean something entirely different [12].)

Mixed states are related to process states, but they are not quite the same. First, there can be mixed states which do not represent any process states. For example, consider

the process presented by the HMM

$$V = \{0,1\}, \mathcal{X} = \{0,1\}, \pi = (\tfrac{1}{2}, \tfrac{1}{2})$$
$$T^0 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, T^1 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}. \tag{3.37}$$

This process has only three process states. (If we have seen any history or history suffix of length 1 or more, then we know the entire past and the entire future almost surely — it is either $\ldots 0.10101\ldots$ or $\ldots 1.01010\ldots$. If not, we are conditioning on λ, and we get the futures $10101\ldots$ and $01010\ldots$ with probability $\frac{1}{2}$ each.) The mixed states corresponding to these process states are $(0,1)$, $(1,0)$, and $(\frac{1}{2},\frac{1}{2})$. The other mixed states do not define process states.

Second, it can happen that two or more different mixed states correspond to a single process state. This can only happen if the presentation in question is not minimal, that is, if it has some redundancy in its states. For example, the process presented by the HMM

$$V = \{0,1\}, \mathcal{X} = \{0,1\}, \pi = (\tfrac{1}{2}, \tfrac{1}{2})$$
$$T^0 = \begin{pmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & 0 \end{pmatrix}, T^1 = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{pmatrix} \tag{3.38}$$

is an elaborate presentation of a fair coin, which has only one process state. The mixed states $(1,0)$ and $(0,1)$, which arise as $N(\pi T^0)$ and $N(\pi T^1)$ respectively, represent the same process state.

Definition 3.4.2. Fix a process and an HMM presentation for it. Let $\mathbf{A}$ be a process state and μ a mixed state. If for all next words w we have $\mathbf{A}(w) = \mu T^w \vec{1}$, then we say that μ is a *mixed state version* of the process state $\mathbf{A}$.

Theorem 3.4.3. Suppose we have a process and an HMM presentation for it. Then every process state, except possibly those in a null set, has a mixed state version.

For a reachable process state $\mathbf{A}$, we have essentially already shown this. If s is a history suffix with $\mathbf{P}(s) > 0$ which induces $\mathbf{A}$, $N(\pi T^s)$ is a mixed state version of $\mathbf{A}$. However, for unreachable states, there is no such simple solution. Most of the rest of this section addresses this issue. The proof of this theorem appears on page 41.

To treat this case, we need to work in a probability space which contains both presentation states and symbols. Begin with our HMM $\left(V, \mathcal{X}, \{T^k\}, \pi\right)$, and its underlying Markov Chain (V, P, π). These define the observation process $\mathcal{P} = \left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$ and the

internal process $\left(V^{\mathbb{Z}}, \mathbb{V}, \overline{\mathbf{P}}\right)$, respectively. We will define the *joint process* of these two to be the process $\mathcal{Q} = \Big(\left(V \times \mathcal{X}\right)^{\mathbb{Z}}, \mathbb{J}, \mathbf{Q}\Big)$ as follows. The alphabet of the joint processes is $V \times \mathcal{X}$ and thus its sequence space is $\left(V \times \mathcal{X}\right)^{\mathbb{Z}}$. Its σ-field is the σ-field generated by the cylinder sets in $\left(V \times \mathcal{X}\right)^{\mathbb{Z}}$. If we have a word $\hat{w} = (v_1, x_1), (v_2, x_2), \ldots, (v_l, x_l)$, we have

$$\mathbf{Q}(\hat{w}) = \pi_{v_1} T^{x_1}_{v_1 v_2} \ldots T^{x_{l-1}}_{v_{l-1} v_l} \left(\sum_{i \in V} T^{x_l}_{v_l i} \right). \tag{3.39}$$

The pair (v, x) corresponds to our original HMM leaving state v and outputting symbol x. Thus $\mathbf{Q}(\hat{w})$ is the probability that the HMM traverses the sequence $v_1, v_2, \ldots, v_l$ of presentation states and, as it does this, emits the word $x_1, x_2, \ldots, x_l$. Specifically, this is the probability that the HMM starts in presentation state v_1, emits x_1 while making a transition to v_2, and then emits x_2 while going to v_3, and so forth. This ends when the HMM emits x_{l-1} during the transition from v_{l-1} to v_l and then emits x_l during a transition to any state. This free choice of the $l+1$th state leads to the sum at the end of equation 3.39. The new process $\mathcal{Q}$ has the HMM presentation $\left(V, V \times \mathcal{X}, \left\{U^k | k \in V \times \mathcal{X}\right\}, \pi\right)$, where if $v \in V$ and $x \in \mathcal{X}$, then $U^{(v,x)}$ is defined by

$$U^{(v,x)}_{ij} = \begin{cases} T^x_{ij} & v = i \\ 0 & v \neq i \end{cases} \tag{3.40}$$

and we can rewrite 3.39 as

$$\mathbf{Q}(\hat{w}) = \pi U^{(v_1, x_1)} \ldots U^{(v_l, x_l)} \vec{1} \tag{3.41}$$

Let $M : V \times \mathcal{X} \to \mathcal{X}$ be the projection map $M(v, x) = x$, and let $M^{\mathbb{Z}} : \left(V \times \mathcal{X}\right)^{\mathbb{Z}} \to \mathcal{X}^{\mathbb{Z}}$ be the projection map on sequence spaces which applies M at each time index: $M^{\mathbb{Z}}(\ldots z_i z_{i+1} \ldots) = (\ldots M(z_i) M(z_{i+1}) \ldots)$. Thus for any subsequence $s = s_a s_{a+1} \ldots s_b$, $s_i \in \mathcal{X}$ when we apply M^{-1} to the cylinder set A_s we get the set of all sequences in $\left(V \times \mathcal{X}\right)^{\mathbb{Z}}$ whose x part matches s,

$$M^{-1}(A_s) = \Big\{ z \in \left(V \times \mathcal{X}\right)^{\mathbb{Z}} | M(z_i) = s_i \text{ for all } i \in a, a+1, \ldots b \Big\}. \tag{3.42}$$

It should be clear that M is a measurable function, and that if $A \in \mathbb{X}$, we have $\mathbf{P}(A) = \mathbf{Q}\big(M^{-1}(A)\big)$.

In some sense, defining joint processes is a more natural way of approaching HMMs, than the path we have taken of defining (symbol) processes first and then introducing

HMMs as ways of representing processes. However, the joint process approach leads one's intuition in a direction other than the one in which this work is going. In particular, the joint process approach does not suggest section 3.6, and in fact could lead one to reject it. This is because introducing HMMs and joint processes first puts presentation states in a more fundamental role than process states. In contrast, the insight which led to section 3.6 resulted in part from observing that process states were actually the more fundamental objects. We will use the joint process only in part of this section.

In section 2.5 we defined R to be the set of words $w \in \mathcal{X}^*$ such that $P(w) \neq 0$. We also defined the set of bad histories $\mathcal{N}$ to be the set of all histories, and we showed that $\mathcal{N}$ is a null set. A history $\mathbf{x}^-$ is in $\mathcal{N}$ if $\lim_{l\to\infty} \mathbf{P}(s|w_l)$ does not exist for some $s \in \mathcal{X}^*$, where w_l is the length l suffix of $\mathcal{N}$. In particular, if $\mathbf{x}^-$ is not in $\mathcal{N}$, we know that every suffix w_l of $\mathbf{x}^-$ lies in R.

Definition 3.4.4. If s is either a history suffix in R or a good history, the mixed state $\eta(s)$ is defined to be that mixed state whose ith coordinate satisfies $(\eta(s))_i = \mathbf{Q}(v_0 = i|s)$ for all $i \in V$. We call $\eta(s)$ the mixed state *induced* by s.

How can we compute induced mixed states? If w is a history suffix in R, we can calculate directly, using equation 3.41 and definition 3.4.4. The answer is far less cumbersome than the calculations needed to produce it, and brings us back to the material of pages 36–36.

$$\mathbf{Q}(v_0 = i|w) = \frac{\mathbf{Q}(i,w)}{\mathbf{Q}(w)} = \frac{\mathbf{Q}(v_0 = i, x_{-l}\dots x_{-1} = w_{-l}\dots w_{-1})}{\mathbf{Q}(x_{-l}\dots x_{-1} = w_{-l}\dots w_{-1})} \tag{3.43}$$

$$\mathbf{Q}(v_0 = i|w) = \frac{\sum_{v_{-l}\dots v_{-1}\in V} \left(\pi U^{(v_{-l},w_{-l})}\dots U^{(v_{-1},w_{-1})} \sum_{x\in\mathcal{X}} U^{(i,x)}\vec{1}\right)}{\sum_{v_{-l}\dots v_{-1}\in V} \left(\pi U^{(v_{-l},w_{-l})}\dots U^{(v_{-2},w_{-1})}\vec{1}\right)} \tag{3.44}$$

$$\mathbf{Q}(v_0 = i|w) = \frac{\pi\left(\sum_{v_{-l}\in V} U^{(v_{-l},w_{-l})}\right)\dots\left(\sum_{v_{-1}\in V} U^{(v_{-1},w_{-1})}\right)\sum_{x\in\mathcal{X}} U^{(i,x)}\vec{1}}{\pi\left(\sum_{v_{-l}\in V} U^{(v_{-l},w_{-l})}\right)\dots\left(\sum_{v_{-1}\in V} U^{(v_{-1},w_{-1})}\right)\vec{1}} \tag{3.45}$$

Note that $\sum_{v \in V} U^{(v,x)} = T^x$ and that $\left(\sum_{x \in \mathcal{X}} U^{(i,x)} \vec{1} \right)_j = \sum_{v \in V, x \in \mathcal{X}} U^{(i,x)}_{j,v} = \delta_{ij}$, which means that $\sum_{x \in \mathcal{X}} U^{(i,x)} \vec{1} = e_i$, the ith standard basis vector. Thus we can write

$$\begin{aligned} \mathbf{Q}(i|w) &= \frac{\pi T^{w_{-l}} \dots T^{w_{-1}} e_i}{\pi T^{w_{-l}} \dots T^{w_{-1}} \vec{1}} \\ &= \frac{\pi T^w e_i}{\pi T^w \vec{1}} = N(\pi T^w) e_i. \end{aligned} \tag{3.46}$$

Thus, the induced mixed state $\eta(w)$ is simply given by $\eta(w) = N(\pi T^w)$.

Before we address the mixed state $\eta(\mathrm{x}^-)$ induced by a history x^-, we need the following theorem, due to technical difficulties of conditioning on sets of measure zero.

Corollary B.2.4. If $\{\mathcal{F}_n\}$ is an increasing sequence of σ-fields and A is an event, then $\mathbf{P}(A|\mathcal{F}_n) \to \mathbf{P}(A|\mathcal{F})$ almost surely, where $\mathcal{F}$ is the smallest σ-field which contains all of the $\mathcal{F}_n$s.

Proposition 3.4.5. For any history x^-, let s_l denote the length l history suffix $x_{-l} \dots x_{-1}$. For almost every x^-, $\eta(s_l) \to \eta(\mathrm{x}^-)$ as $l \to \infty$.

Proof. For each positive integer l, let $\mathcal{F}_l \subset \mathbb{J}$ be the σ-field generated on $(V \times \mathcal{X})^{\mathbb{Z}}$ by history suffixes $w \in \mathcal{X}^*$ of length l, and let $\mathcal{F}_\infty$ be the σ-field generated by the union of the $\mathcal{F}_l$s. Thus $\mathcal{F}_\infty$ is the set of inverse images under M of sets in the history σ-field $\mathbb{H}$ of the process $\mathcal{P}$. Also, let $A_i \subset (V \times \mathcal{X})^{\mathbb{Z}}$ be the set on which $v_0 = i$. Now, applying theorem B.2.4, we get

$$\mathbf{Q}(A_i|\mathcal{F}_l) \to \mathbf{Q}(A_i|\mathcal{F}_\infty) \tag{3.47}$$

almost surely as $l \to \infty$. For a given history $\mathrm{x}^- \notin \mathcal{N}$, and for each positive integer l, let s_l denote the length l history suffix $x_{-l} \dots x_{-1}$. Now,

$$\mathbf{Q}(A_i|\mathcal{F}_l)(\mathrm{x}^-) = \mathbf{Q}(A_i|s_l) = (\eta(s_l))_i \tag{3.48}$$

since we know that $s_l \in R$. Similarly,

$$\mathbf{Q}(A_i|\mathcal{F}_\infty)(\mathrm{x}^-) = \mathbf{Q}(A_i|\mathrm{x}^-) = \left(\eta(\mathrm{x}^-)\right)_i, \tag{3.49}$$

so equation 3.47 becomes $\eta(s_l) \to \eta(\mathrm{x}^-)$ almost surely as $l \to \infty$, for almost every $\mathrm{x}^- \notin \mathcal{N}$, or simply for almost every x^-.■

The next result establishes that $\eta(s)$ contains all the information about the past which is contained in s and which is relevant to the future.

Proposition 3.4.6 Let w be any word in $\mathcal{X}^*$. If s is a history suffix in R, then $\mathbf{P}(w|s) = \eta(s)T^w\vec{1}$. And if x^- is a good history, then $\mathbf{P}(w|s) = \eta(\mathrm{x}^-)T^w\vec{1}$ almost surely.

Proof. If s is a history suffix, we know that

$$\mathbf{P}(w|s) = \frac{\pi T^s}{\pi T^s \vec{1}} T^w \vec{1} = N(\pi T^s) T^w \vec{1}. \tag{3.50}$$

Since $\eta(s) = N(\pi T^s)$, we have $\mathbf{P}(w|s) = \eta(s)T^w\vec{1}$.

For a good history x^-, let $s_l = x_{-l} \ldots x_{-1}$ for each l, and let $\mathcal{F}_l \subset \mathbb{X}$ be the σ-field generated by the history suffixes of length l. In addition, let $A_w \subset \mathcal{X}^{\mathbb{Z}}$ be the cylinder set of sequences which contain w. Now, if we apply theorem B.2.4, we get $\mathbf{P}(A_w|\mathcal{F}_l) \to \mathbf{P}(A_w|\mathcal{F}_\infty)$ almost surely as $l \to \infty$, or equivalently $\mathbf{P}(w|s_l) \to \mathbf{P}(w|\mathrm{x}^-)$.

On the other hand, we know that $\eta(s_l) \to \eta(\mathrm{x}^-)$ almost surely. The function $\mu \to \mu T^w \vec{1}$ is continuous, so $\eta(s_l)T^w\vec{1} \to \eta(\mathrm{x}^-)T^w\vec{1}$ almost surely. And since $\mathbf{P}(w|s_l) = \eta(s_l)T^w\vec{1}$ almost surely, we know that $\mathbf{P}(w|s_l)$ converges almost surely to both $\mathbf{P}(w|\mathrm{x}^-)$ and to $\eta(\mathrm{x}^-)T^w\vec{1}$, so it must be true that $\eta(\mathrm{x}^-)T^w\vec{1} = \mathbf{P}(w|\mathrm{x}^-)$ almost surely. ■

Proposition 3.4.6 directly implies that the past and the future are conditionally independent given the mixed state induced by the past. At last, we can return to mixed state versions of process states and prove theorem 3.4.3.

Proof of theorem 3.4.3 Let $\mathbf{A}$ be a process state for $\mathcal{P} = (\mathcal{X}, \mathcal{X}^{\mathbb{Z}}, \mathbf{P})$. Then there is either a history or a history suffix which induces $\mathbf{A}$. Let s be any such history or history suffix. For all next words w, $\mathbf{A}(w)$ is defined to be $\mathbf{P}(w|s)$ almost surely, and we know that $\mathbf{P}(w|s) = \eta(s)T^w\vec{1}$, so $\mathbf{A}(w) = \eta(s)T^w\vec{1}$. Thus $\eta(s)$ is a mixed state version of $\mathbf{A}$. ■

Finally, with the remainder of this section, we will define a new presentation, called the *mixed state representation* (MSR). If we start with a presentation $(V, \mathcal{X}, \{T^x\}, \pi)$, let $\overline{V}$ be the set of all mixed states $\eta(s)$ which are induced by a history $s \in R$ or a history suffix $\mathrm{x}^- \notin \mathcal{N}$. Elements of $\overline{V}$ are presentation states of the mixed state representation. That is, presentation states $\overline{V}$ of the MSR are mixed states of the

presentation $(V, \mathcal{X}, \{T^x\}, \pi)$. The mixed state representation is another presentation of the process defined by $(V, \mathcal{X}, \{T^x\}, \pi)$. Notably, it may have infinitely many states. It is with this representation in mind that we use the word *state* in the term *mixed state*.

Suppose what we know of the history of our process $\mathcal{P}$ is that the most recent output word was the history suffix w. Then the next symbol will be $x \in \mathcal{X}$ with probability $\mathbf{P}(x|w) = \eta(w)T^x\vec{1}$, and if x is the next symbol, then the known history word becomes wx. Now, we will look at this transition in terms of the mixed states. Since we know that the history suffix is w, we are in mixed state $\eta(w)$. From $\eta(w)$, the next symbol is x with probability $\mathbf{P}(x|\eta(w)) = \eta(w)T^x\vec{1}$, and if x is chosen as the next symbol, then a transition is made to the MSR state $\eta(wx)$.

In order to use mixed states as states, we need to be able to compute $\eta(wx)$ from $\eta(w)$ without using w. Fortunately, this is not difficult to do.

$$\begin{aligned} \eta(wx) &= N(\pi T^w T^x) \\ &= \frac{\pi T^w T^x}{\pi T^w T^x \vec{1}} \\ &= \frac{\pi T^w T^x / \pi T^w \vec{1}}{\pi T^w T^x 1 / \pi T^w \vec{1}}. \end{aligned} \tag{3.51}$$

Thus we have

$$\eta(wx) = \frac{\eta(w)T^x}{\eta(w)T^x\vec{1}} = N(\eta(w)T^x) = C_x(\eta(w)). \tag{3.52}$$

Note that w does not appear except in $\eta(w)$ and $\eta(wx)$.

Now we can define the mixed state representation. As we have stated, its presentation states are elements of $\overline{V}$, mixed states which are induced by histories or history suffixes. We write them as row vectors $\mu = (\mu_1, \ldots, \mu_{|V|})$. Its symbol set, clearly, will be $\mathcal{X}$. Because $\overline{V}$ may be infinite or even uncountable, we cannot define transition matrices, but we can give equivalent information. Given a state $\mu \in \overline{V}$ and a symbol $x \in \mathcal{X}$, if the current state is μ,

(i) the probability that x is emitted is $\mathbf{P}(x|\mu) = \mu T^x\vec{1}$, and
(ii) if x is emitted, the next state is $C_x(\mu) = N(\mu T^x)$.

We will not address here the issue of whether or not a stationary distribution on $\overline{V}$ exists. To use the mixed state presentation to compute the probability of a word w,

assume presentation starts in state $\pi = \eta(\lambda) \in \overline{V}$ and compute

$$\prod_{i=1}^{l} \mu_{i-1} T^{w_i} \vec{1} \tag{3.53}$$

where $l = |w|$, $\mu_0 = \pi$ and $\mu_i = C_{w_i}(\mu_{i-1}) = N(\mu_{i-1}T^{w_i})$ for $i \geq 1$. It can easily be verified that the result of this calculation is equal to the probability $\mathbf{P}(w) = \pi T^{w_1} T^{w_2} \ldots T^{w_l} \vec{1}$ assigned by $\left(V, \mathcal{X}, \{T^k\}, \pi\right)$. The product in equation 3.53 is simply the product of the quantities which the normalizing function N divides out of the μ_i.

Note that the mixed state representation is deterministic. That is, for any MSR state $\mu \in \overline{V}$ and any symbol $x \in \mathcal{X}$, there is a unique MSR state $C_x(\mu)$ to which a transition involving the emission of x is possible. Further, the MSR states are in one-to-one correspondence with the process states, except perhaps for a set of each of measure zero.

In this section, we defined mixed states and showed that they are intimately related to the process states. In fact, the significance of mixed states is that they give us a way of representing the process states.

3.5 Examples

At this point we will digress from the formal development and present several examples in detail. These examples are chosen in part to illustrate the variety of behaviors which are seen in SFAs. The calculations to support the conclusions are not presented here; the reader is encouraged to perform them.

The Golden Mean Process The first example is the *Golden Mean Process* (GMP) which we have already seen in section 2.6. This presentation is deterministic, and the recurrent process states and the presentation states coincide, so GMP is a stochastic deterministic finite automaton.

GMP has two symbols, and its smallest HMM presentation has two states. Its most prominent feature is that its output sequences never contain pairs of consecutive 0s. The reader should be able to verify that $\pi = \left(\frac{2}{3}, \frac{1}{3}\right)$ from the transition matrices T^0 and T^1.

$$\begin{gathered} V = \{\mathbf{B}, \mathbf{C}\},\ \mathcal{X} = \{0, 1\},\ \pi = \left(\tfrac{2}{3}, \tfrac{1}{3}\right) \\ T^0 = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & 0 \end{pmatrix},\ T^1 = \begin{pmatrix} \frac{1}{2} & 0 \\ 1 & 0 \end{pmatrix} \end{gathered} \tag{3.54}$$

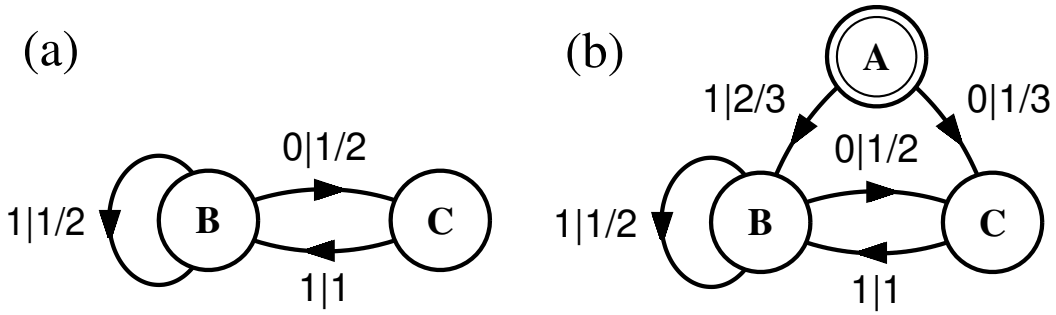

Fig. 3.3 (a) Labeled directed graph presentation of GMP. (b) Process state graph presentation of the process GMP. In this case, the recurrent process states coincide with the presentation states.

This process has three process states. If w is a history or history suffix which ends in 1, it induces process state $\mathbf{B}$. The mixed state it induces is $N(\pi T^w) = (1, 0)$. If w is a history or history suffix which ends in 0, it induces process state $\mathbf{C}$ and mixed state $(0, 1)$. This covers all histories and all history suffixes except λ, which induces process state $\mathbf{A}$, which is transient, and mixed state $\pi = (\frac{2}{3}, \frac{1}{3})$. The probabilities associated with the start state are $\mathbf{P}(1|\lambda) = \pi T^1 \vec{1} = \frac{2}{3}$ and $\mathbf{P}(0|\lambda) = \pi T^0 \vec{1} = \frac{1}{3}$. Similarly, the states to which these transitions are made are identified by comparing mixed states; $C_1(\pi) = (1, 0)$ and $C_0(\pi) = (0, 1)$. These are the mixed states associated to states $\mathbf{B}$ and $\mathbf{C}$, respectively.

The Simple Nondeterministic Source Our next example is the *Simple Nondeterministic Source (SNS)*, which we saw in section 3.2. This process can be represented with only two presentation states, but as we will see shortly, it has infinitely many process states. A two-state HMM presentation is

$$V = \{\mathbf{A}, \mathbf{B}\},\ \mathcal{X} = \{0, 1\},\ \pi = (\tfrac{1}{2}, \tfrac{1}{2}),$$
$$T^0 = \begin{pmatrix} 0 & 0 \\ \frac{1}{2} & 0 \end{pmatrix},\ T^1 = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} \end{pmatrix}. \tag{3.55}$$

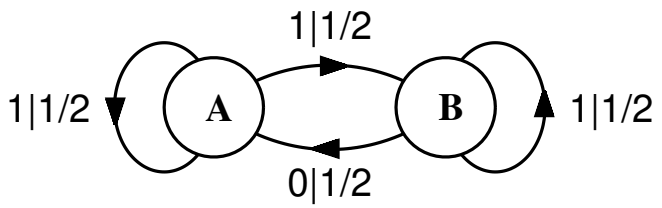

Fig. 3.4 Labeled directed graph presentation for SNS.

Let $w_n = 01^n$, the word consisting of a 0 followed by n 1s, and let $\mathbf{A}_n$ be the process state induced by w_n. The matrix corresponding to w_n is

$$T^{w_n} = T^0 \left(T^1\right)^n = \begin{pmatrix} 0 & 0 \\ 2^{-n-1} & n2^{-n-1} \end{pmatrix}, \tag{3.56}$$

so the mixed state $\mu(w_n)$ corresponding to $\mathbf{A}_n$ is

$$N(\pi T^{w_n}) = N\left(2^{-n-1}, n2^{-n-1}\right) = \left(\frac{1}{n+1}, \frac{n}{n+1}\right). \tag{3.57}$$

The first few of these states are listed in table 3.1. The $\mathbf{A}_n$ are all distinct states, since their mixed state versions are all distinct. In fact, the $\mathbf{A}_n$ comprise all but one of the process states. Also, the word 0 is a synchronizing word, since $\mathbf{P}(\cdot|0) = \mathbf{P}(\cdot|w0)$ for all words w such that $\mathbf{P}(w0) > 0$. We can verify this by calculating $C_0(\pi) = (1, 0)$ and $C_0(\mu) = (1, 0)$ for all μ. Thus all of the w_ns are synchronizing words, and all of the $\mathbf{A}_n$ are reachable recurrent states.

SNS is also an example of a process in which reachable recurrent states are induced by words which are not synchronizing. This precludes the possibility of a converse to proposition 2.6.2, which said that synchronizing words induce reachable recurrent states. The word 11 induces the process state $\mathbf{A}_3$, a reachable recurrent state, also induced by $w_3 = 0111$. However, 11 is not a synchronizing word, because $\mathbf{P}(\cdot|011)$ and $\mathbf{P}(\cdot|111)$ are not equal to $\mathbf{A}_3$.

History or history suffix s	Mixed state $\mu(s)$	Process state	$\mathbf{P}(\text{symbol } 0 \mid w)$
$\ldots 0$	$(1, 0)$	$\mathbf{A}_0$	0
$\ldots 01$	$\left(\frac{1}{2}, \frac{1}{2}\right)$	$\mathbf{A}_1$	$\frac{1}{4}$
$\ldots 011$	$\left(\frac{1}{3}, \frac{2}{3}\right)$	$\mathbf{A}_2$	$\frac{1}{3}$
$\ldots 0111$	$\left(\frac{1}{4}, \frac{3}{4}\right)$	$\mathbf{A}_3$	$\frac{3}{8}$
$\ldots 01111$	$\left(\frac{1}{5}, \frac{4}{5}\right)$	$\mathbf{A}_4$	$\frac{2}{5}$
$\ldots 01^n$	$\left(\frac{1}{n+1}, \frac{n}{n+1}\right)$	$\mathbf{A}_n$	$\frac{1}{2}\frac{n}{n+1}$
infinitely many 1s	$(0, 1)$	$\mathbf{A}_\infty$	$\frac{1}{2}$

Table 3.1 The first few mixed and process states of the "simple nondeterministic source".

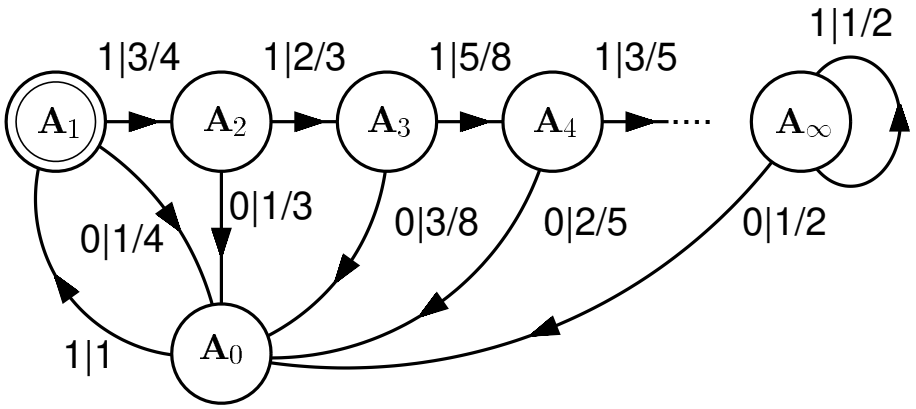

Fig. 3.5 An abbreviated version of the deterministic labeled directed graph presentation for the process "simple nondeterministic source," which has infinitely many process states.

The Cantor process Our third example is the *Cantor* process, which has the following HMM presentation:

$$V = \{\mathbf{A}, \mathbf{B}\},\ \mathcal{X} = \{0, 1\},\ \pi = (\tfrac{1}{2}, \tfrac{1}{2}),$$
$$T^0 = \begin{pmatrix} 0.55 & 0 \\ 0.30 & 0.15 \end{pmatrix},\ T^1 = \begin{pmatrix} 0.15 & 0.30 \\ 0 & 0.55 \end{pmatrix}. \tag{3.58}$$

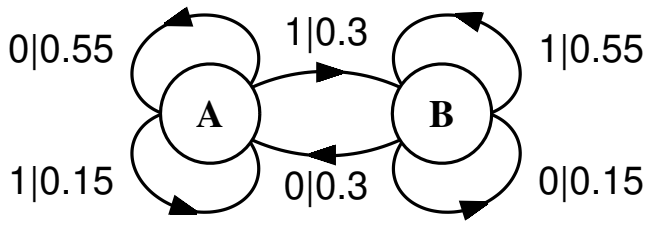

Fig. 3.6 Labeled directed graph presentation for the Cantor process.

Recall that a process state is an equivalence class of histories and history suffixes. For the Cantor process, all of these equivalence classes are trivial: every history and every history suffix induces a future conditional distribution which is different from that generated by every other history and every other history suffix. (Of course, pairs of future conditional distributions exist arbitrarily close to one another.) The result is that the Cantor process has uncountably many elusive process states, one induced by each history. Also, it has countably many strictly transient states, which are in one-to-one correspondence with the history suffixes. Thus, this is an example of a process with no synchronizing words.

Note that

$$N\big((x,1-x)T^1\big) = \left(\frac{3x}{11-2x}, \frac{11-5x}{11-2x}\right) \approx \left(\frac{x}{3}, 1-\frac{x}{3}\right), \text{ and}$$
$$N\big((1-y,y)T^0\big) = \left(\frac{11-5y}{11-2y}, \frac{3y}{11-2y}\right) \approx \left(1-\frac{y}{3}, \frac{y}{3}\right). \tag{3.59}$$

These approximations are exact at $(0,1)$ and $(1,0)$, and are within $\frac{1}{54}$ in between. Thus, if μ is the mixed state induced by a history s, appending a symbol to s corresponds approximately to moving μ two-thirds of the distance to either $(0,1)$ or $(1,0)$, respectively. The mixed states induced by histories form a set similar to the middle-thirds Cantor set, hence the process's name. This may be seen in figure 3.7, which is a plot of the Cantor process's mixed states, all of which lie on the line segment with endpoints $(0,1)$ and $(1,0)$. The mixed states induced by history suffixes lie in the middle of the intervals which are deleted to form the approximate Cantor set. (It is possible to construct an HMM for which the mixed states induced by histories are exactly the middle-thirds Cantor set, but it is degenerate — it is equivalent to a fair coin.)

Fig. 3.7 The mixed states for the process Cantor. Dots are mixed states corresponding to elusive process states. The small vertical lines are the mixed states corresponding to the subset of transient process states.

The Two Biased Coins process The last example we will look at here is the *Two Biased Coins (2BC)* process. The process can be simulated with a pair of biased coins. One of the biased coins is chosen by a flip of a fair coin. The chosen biased coin is then flipped to produce a bi-infinite sequence. Like the above examples, it has the following two-state presentation:

$$V = \{\mathbf{A}, \mathbf{B}\},\ \mathcal{X} = \{0,1\},\ \pi = (\tfrac{1}{2}, \tfrac{1}{2}),$$
$$T^0 = \begin{pmatrix} \frac{3}{4} & 0 \\ 0 & \frac{1}{4} \end{pmatrix},\ T^1 = \begin{pmatrix} \frac{1}{4} & 0 \\ 0 & \frac{3}{4} \end{pmatrix}. \tag{3.60}$$

2BC is a reducible process, as it consists essentially of two processes with no interaction between them; see figure 3.8.

Calculation of 2BC's mixed states is equivalent to using Bayesian methods to infer which of the two biased coins is being flipped. The stationary distribution π is the prior

distribution, and $\mu(w)$ is simply the posterior distribution over the presentation states given the word w. The procedure we use for calculating $\mu(wx)$ from $\mu(w)$ can be viewed as a procedure to dynamically update the posteriors. If w is a word of length $i+j$ consisting of i 0s and j 1s in any order, then $T^w = \begin{pmatrix} \frac{3^i}{4^{i+j}} & 0 \\ 0 & \frac{3^j}{4^{i+j}} \end{pmatrix}$, and resulting mixed state $\mu(w) = N(\pi T^w)$ is $\frac{1}{3^i+3^j}(3^i, 3^j) = \frac{1}{3^{i-j}+1}(3^{i-j}, 1)$. So the mixed state — and the process state — induced by a word depends only on the difference between the number of 0s and the number of 1s in the word. Thus, there are countably infinitely many reachable process states, one for each integer. Some of these process state are portrayed in figure 3.8a.

With finite data, we are never sure which presentation state the process is in, so all reachable process states are transient. Asymptotically, as the length of the word goes to infinity, we can be sure with probability 1 which of the presentations states we are in. Thus, the mixed state versions of the recurrent process states are $(1, 0)$ and $(0, 1)$, Hence there are exactly two recurrent process states, both of which are unreachable and which correspond exactly to the presentation states.

There are also uncountably many histories in which the difference between the number of zeros and the number of ones is bounded, for example $\ldots 010101$. These histories induce elusive states, the total probability of which is 0.

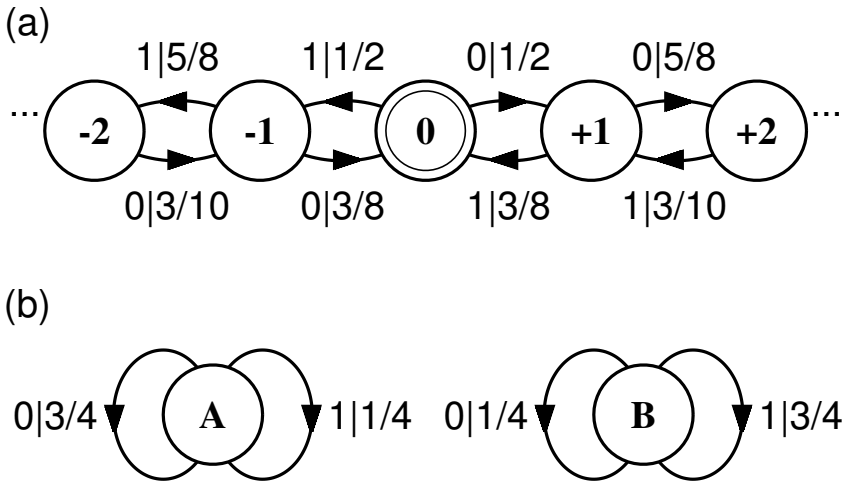

Fig. 3.8 (a) A subset of the transient process states for 2BC. The state induced by a word is determined solely by the number of 0s minus the number of 1s in the word. All of the transient states are infinitely preceded. (b) The recurrent states of 2BC. Both **A** and **B** are unreachable recurrent states. Each connected component of the graph corresponds to a recurrent component of the underlying Markov Chain.

3.6 When Does a Process have an HMM Presentation?

In this section we propose a characterization of when a process is an SFA — that is, when a process has a (finite) HMM presentation — and we show that it is a necessary condition. This characterization is a keystone of this dissertation. It leads almost directly to the reconstruction algorithm of chapter 5. And it follows from the following observation.

If we have a process with an HMM presentation $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$, then a mixed state for that presentation is a distribution on V, or equivalently, a vector of $|V|$ components. This means that all the mixed states lie in the $|V|$-dimensional vector space $\mathbb{R}^{|V|}$. This in turn means that the dimension of the span of the mixed states is less than or equal to $|V| < \infty$. As process states are essentially equivalent to mixed states, we can make a similar statement about the process states.

First, we need to be able to work with process states as elements of a vector space. Let $\mathcal{W}$ be the set of all signed measures on the future space. These include the process states: if $\mathbf{A}$ is a process state, then $\mathbf{A} \in \mathcal{W}$. For any $\mathbf{A}, \mathbf{B} \in \mathcal{W}$ and $c, d \in \mathbb{R}$, we define $c\mathbf{A} + d\mathbf{B}$ as follows. For all future words w, $(c\mathbf{A} + d\mathbf{B})(w)$ is defined to be $c\mathbf{A}(w) + d\mathbf{B}(w)$, so that $\mathcal{W}$ is a vector space. In addition, if $\mathbf{A}, \mathbf{B}$ are probability measures and $c, d \geq 0,\ c + d = 1$, then $c\mathbf{A} + d\mathbf{B}$ is a probability measure.

We now state the main result of this section.

Theorem 3.6.1 Given a process $\mathcal{P}$, let $\mathcal{U}$ be the subspace of $\mathcal{W}$ spanned by the reachable process states. If the $\mathcal{P}$ has an HMM presentation $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$, then $\dim(\mathcal{U}) \leq |V|$.

Before we can readily prove this result, we need to develop the connection between $\mathcal{W}$ and $\mathbb{R}^{|V|}$.

Lemma 3.6.2 Suppose we have $\mathbf{A}, \mathbf{B} \in \mathcal{W}$ and $\mu, \nu \in \mathbb{R}^{|V|}$ such that for all future words w we have $\mu T^w \vec{1} = \mathbf{A}(w)$ and $\nu T^w \vec{1} = \mathbf{B}(w)$. Then for all $c, d \in \mathbb{R}$ and for all future words w, we have $(c\mu + d\nu)T^w \vec{1} = (c\mathbf{A} + d\mathbf{B})(w)$.

Proof. $(c\mu + d\nu)T^w \vec{1} = c\left(\mu T^w \vec{1}\right) + d\left(\nu T^w \vec{1}\right) = c\mathbf{A}(w) + d\mathbf{B}(w) = (c\mathbf{A} + d\mathbf{B})(w)$. ■

For the next lemma, we need some additional notation. We will use $\underline{0}$ to denote the zero vector in $\mathbb{R}^{|V|}$ (Note that $\underline{0}$ is a row vector, in contrast to $\vec{1}$.) Also, we will use $\mathbf{0}$ to

denote the zero measure in $\mathcal{W}$. Thus we have, for all future words w, $\underline{0}T^w\vec{1} = \mathbf{0}(w) = 0$.

Lemma 3.6.3 Suppose we have reachable process states $\mathbf{A}_1, \ldots, \mathbf{A}_l \in \mathcal{W}$, and vectors $\mu^1, \ldots, \mu^l \in \mathbb{R}^{|V|}$ such that μ^i is a mixed state version of $\mathbf{A}_i$ for each $i \in 1, \ldots, l$. If there exist real numbers $c_1, \ldots, c_l$, not all zero, such that $\sum_{i=1}^{l} c_i \mu^i = \underline{0}$, then $\sum_{i=1}^{l} c_i \mathbf{A}_i = \mathbf{0}$.

Proof. For all future words w, we have

$$\left(\sum_{i=1}^{k} c_i \mathbf{A}_i\right)(w) = \left(\sum_{i=1}^{k} c_i \mu^i\right) T^w \vec{1} \tag{3.61}$$

by lemma 3.6.2. However, the right hand side of equation 3.61 is zero by assumption. Thus the left hand side is also zero for all w, and we have

$$\sum_{i=1}^{k} c_i \mathbf{A}_i = \mathbf{0}. \blacksquare \tag{3.62}$$

Proof of theorem 3.6.1 Choose any $|V| + 1$ reachable process states $\mathbf{A}_1, \ldots, \mathbf{A}_{|V|+1}$, and choose $\mu_1, \ldots, \mu_{|V|+1} \in \mathbb{R}^{|V|}$ such that μ_i is a mixed state version of $\mathbf{A}_i$ for each $i \in 1, \ldots, k$. The μ_i are a set of $|V| + 1$ vectors in a $|V|$-dimensional vector space, so they must be dependent. That is, there must exist $c_1, \ldots, c_{|V|+1}$ such that $\sum_{i=1}^{|V|+1} c_i \mu_i = \underline{0}$. Now, by lemma 3.6.3, $\sum_{i=1}^{|V|+1} c_i \mathbf{A}_i = 0_{\mathcal{W}}$, so the $\mathbf{A}_i$s are linearly dependent. Thus, we have shown that a set of linearly independent process states has size at most $|V|$, so the span of the process states is at most $|V|$-dimensional. $\blacksquare$

The following fact about SFAs follows immediately from theorem 3.6.1.

Corollary 3.6.4. Given a process, let $\mathcal{U}$ be the span of its process states. If it has a (finite) HMM presentation, then $\dim(\mathcal{U}) < \infty$.

Proof. For any finite HMM $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$, we have $|V| < \infty$. Thus, using theorem 3.6.1, we have $\dim(\mathcal{U}) \leq |V| < \infty$.$\blacksquare$

As an illustration of the use of corollary 3.6.4, we will now prove the following statement, which was stated without proof at the end of section 3.3.

Proposition 3.6.5. The modified nested parentheses process does not have an HMM presentation.

Proof. Let $w_n = !(^{n-1}$ be the length n word consisting of a ! followed by $n-1$ (s. Similarly, let $s_n =)^{n-1}!$ be the mirror image of w_n. After the word w_n, the counter must be $n-1$. Before the word s_m, the counter must be $m-1$. Thus s_m cannot follow w_n if $m \neq n$:

$$\mathbf{P}(s_m|w_n) = \begin{cases} \left(\frac{2}{3}\right)^n & m = n \\ 0 & m \neq n. \end{cases} \tag{3.63}$$

Now let $\mathbf{A}_n$ be the process state induced by w_n,

$$\mathbf{A}_n(s_m) = \begin{cases} \left(\frac{2}{3}\right)^n & m = n \\ 0 & m \neq n. \end{cases} \tag{3.64}$$

For every n and any for linear combination $c_1, \ldots, c_{n-1}$, we have $(c_1\mathbf{A}_1 + \ldots c_{n-1}\mathbf{A}_{n-1})(s_n) = 0$, while $\mathbf{A}_n(s_n) > 0$. Thus, $\mathbf{A}_n$ is linearly independent of $\mathbf{A}_1, \ldots, \mathbf{A}_{n-1}$. In this way we see that we can construct arbitrarily large, linearly independent sets of process states. Thus we have $\dim(\text{span}\{\mathbf{A}_i | i > 0\}) = \infty$, so this process cannot have an HMM presentation. ■

A related condition for functions of Markov chains was shown by Gilbert [20], and variants appear in [10] and [21]. These conditions are stated in terms of a different context of definitions and terminology, so that their exact relationship to corollary 3.6.4 is difficult to ascertain. The author suspects that if one developed the appropriate machinery to connect these contexts, one would find that the conditions are equivalent.

We have shown that $\dim(\mathcal{U}) < \infty$ is a necessary condition for a process to have an HMM presentation. It is almost a sufficient condition. In order to make this precise, however, we will need to develop a generalization of HMMs. This generalization is the subject of the next chapter.

4 Generalized Hidden Markov Models

At the beginning of the last chapter, we viewed a Hidden Markov Model as a Markov chain with a stochastic output filter, and we only interpreted the presentation states as the states of a Markov Chain. Over the course of that chapter, we used the tools of linear algebra more and more, working with mixed states rather than directly with presentation states. Finally, we introduced an alternative interpretation of the presentation states — that they are basis elements for the set of mixed states, or equivalently, for the set of process states. In this interpretation, the transformation matrices define the process by defining linear transformations on the mixed states. As we will see, this linear algebra interpretation is the more fundamental one.

In this chapter, we will define and use a generalization of Hidden Markov Models in which the presentation states cannot be interpreted as states of a Markov chain, but can only reasonably be interpreted as basis elements. The first use of Generalized Hidden Markov appears to have been as a counter-example in [10]. Several authors have used Generalized Hidden Markovs and similar techniques in recent years, including connection to neural nets developed in [22] and the solution of the problem of HMM equivalence in [7]. Late in this chapter, and in the next one, we will choose a linearly independent basis for the span of the process states, and use these basis vectors as presentation states of a new presentation we construct directly from the process.

4.1 Generalized Hidden Markov Models

When we think of a Hidden Markov Model as an object of linear algebra, it makes sense to consider what happens when we perform a change of basis — a canonical linear algebra operation. And so, after one convenient definition, we will work through a change of basis for a generic HMM.

Definition 4.1.1. A *unit-sum* vector is a vector whose components sum to one. A unit-sum matrix is a matrix whose row vectors are unit-sum vectors.

That is, a row vector v is unit-sum row vector if $v\vec{1} = 1$, and a matrix A is a unit sum matrix if

$$A\vec{1} = \vec{1}. \tag{4.1}$$

Note that the inverse of a unit-sum matrix must also be unit-sum. If we multiply both sides of equation 4.1 by A^{-1}, we get

$$I\vec{1} = A^{-1}\vec{1} \tag{4.2}$$

and, clearly, $I\vec{1} = \vec{1}$. If a unit-sum vector satisfies the additional requirement that all of its components are nonnegative, then it is a *stochastic* vector. Similarly a matrix is stochastic if all of its rows are stochastic vectors.

Suppose we have an HMM $\mathcal{I} = \left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$. A mixed state $\mu = \left(\mu_1, \ldots, \mu_{|V|}\right)$ for this HMM is a stochastic vector in a vector space with basis elements associated to the states $i \in V$: if μ is induced by a history object s, then $\mu_i = \mathbf{P}(i|s)$. The process state associated with μ is a linear combination $\sum_i \mu_i \mathbf{A}_i$ of the conditional future distributions $\mathbf{A}_i = \mathbf{P}(\cdot|i)$. If we let $\mathcal{U}$ be the span of all of the HMM's reachable process states, then $\left\{\mathbf{A}_1, \ldots, \mathbf{A}_{|V|}\right\}$ is the basis we used for $\mathcal{U}$ in section 3.6.

We will now work through a change of basis. We begin by choosing a new basis $\left\{\mathbf{B}_1, \ldots, \mathbf{B}_{|V|}\right\}$ for $\mathcal{U}$ as follows: choose an invertible unit-sum $|V| \times |V|$ matrix M, and for all i let

$$\mathbf{B}_i = \sum_j M_{ij} \mathbf{A}_j. \tag{4.3}$$

Clearly, for all i, $\mathbf{B}_i \in \mathcal{U}$, and because M is invertible, $\left\{\mathbf{B}_1, \ldots, \mathbf{B}_{|V|}\right\}$ is a basis for $\mathcal{U}$.

This change of basis calculation will be facilitated by the following somewhat nonstandard notation. We will write the vectors of the basis in a formal column vector *as if they were scalars*. That is, we define the formal column vectors $\vec{\mathbf{A}}$ and $\vec{\mathbf{B}}$ by

$$\vec{\mathbf{A}} = \begin{pmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_{|V|} \end{pmatrix} \text{ and } \vec{\mathbf{B}} = \begin{pmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_{|V|} \end{pmatrix} \tag{4.4}$$

Thus, we may rewrite 4.3 as

$$\vec{\mathbf{B}} = M\vec{\mathbf{A}}. \tag{4.5}$$

In this notation, if $\mu \in \mathbb{R}^{|V|}$ is a mixed state, then it is a row vector, and it describes the process state $\mu\vec{\mathbf{A}}$. From equation 4.5, we get $M^{-1}\vec{\mathbf{B}} = \vec{\mathbf{A}}$, so $\mu M^{-1}\vec{\mathbf{B}} = \mu\vec{\mathbf{A}}$. Thus if $\nu = \mu M^{-1}$, then $\mu\vec{\mathbf{A}} = \nu\vec{\mathbf{B}}$, so μ and ν describe the same process state in different

coordinate systems. Hence M^{-1} maps $\vec{\mathbf{A}}$–coordinates to $\vec{\mathbf{B}}$–coordinates. Since ν is not a mixed state for any HMM we have defined yet, we will call it a coordinate vector.

Our HMM's stationary distribution π and its transition matrices are all given in $\vec{\mathbf{A}}$–coordinates. What do they look like in $\vec{\mathbf{B}}$–coordinates? π is simply a mixed state, so it transforms to the coordinate vector $\tau = \pi M^{-1}$ as we have indicated above. If ν is a coordinate vector, then $\nu\vec{\mathbf{B}} \in \mathcal{U}$, and νM defines a row vector in $\vec{\mathbf{A}}$ coordinates — a mixed state. We can operate on νM with the operator T^k, and transform the result back to $\vec{\mathbf{B}}$ coordinates with M^{-1}. The result is that the operation $\mu \to \mu T^k$ in $\vec{\mathbf{A}}$ coordinates becomes $\nu \to \nu M T^k M^{-1}$ in $\vec{\mathbf{B}}$ coordinates. That is, the similarity transformation which transforms T^k into $\vec{\mathbf{B}}$ coordinates produces the matrix $U^k = MT^kM^{-1}$.

Now, if we define a set of formal symbols $V' = \{1', 2', \ldots, |V|'\}$, we can construct a quadruple $(V', \mathcal{X}, \{U^k\}, \tau)$. This quadruple looks like an HMM. It may fail to be one, however, because the U^k matrices may have negative entries. Nonetheless, it satisfies the rest of the definition of an HMM. The transition matrices $\{U^k\}$ satisfy $\left(\sum_k U^k\right)\vec{1} = \vec{1}$, since

$$\begin{aligned}\left(\sum_k U^k\right)\vec{1} &= \left(\sum_k MT^kM^{-1}\right)\vec{1} \\ &= M\left(\sum_k T^k\right)M^{-1}\vec{1} \\ &= M\left(\sum_k T^k\right)\vec{1} \\ &= M\vec{1} = \vec{1}.\end{aligned} \tag{4.6}$$

And τ satisfies $\tau = \tau\sum_k U^k$ since

$$\begin{aligned}\tau\sum_k U^k &= \pi M^{-1}\sum_k MT^kM^{-1} \\ &= \pi M^{-1}M\left(\sum_k T^k\right)M^{-1} \\ &= \pi\left(\sum_k T^k\right)M^{-1} \\ &= \pi M^{-1} = \tau.\end{aligned} \tag{4.7}$$

Further, if we manipulate $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$ as if it were an HMM, and calculate $\tau U^w \vec{1}$ for an arbitrary word $w = w_1 \ldots w_l$, we get

$$\begin{aligned} \tau U^w \vec{1} &= \tau U^{w_1} \ldots U^{w_l} \vec{1} \\ &= \left(\pi M^{-1}\right)\left(M U^{w_1} M^{-1}\right) \ldots \left(M U^{w_1} M^{-1}\right) \vec{1} \\ &= \pi T^{w_1} \ldots T^{w_l} M^{-1} \vec{1}, \end{aligned} \tag{4.8}$$

and since M^{-1} is unit-sum, this becomes

$$\tau U^w \vec{1} = \pi T^w \vec{1}. \tag{4.9}$$

Thus, for every word $w \in \mathcal{X}^*$, we get $\mathbf{P}(w) = \tau U^w \vec{1}$. In spite of the fact that it is not an HMM, $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$ defines a process as if it were. We will call it a *Generalized Hidden Markov Model (GHMM)*, following [8].

For instance, consider the following presentation of the Golden Mean Process which we saw in section 3.5:

$$\begin{gathered} V = \{\mathbf{B}, \mathbf{C}\},\ \mathcal{X} = \{0, 1\},\ \pi = \left(\tfrac{2}{3}, \tfrac{1}{3}\right) \\ T^0 = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & 0 \end{pmatrix},\ T^1 = \begin{pmatrix} \frac{1}{2} & 0 \\ 1 & 0 \end{pmatrix} \end{gathered} \tag{4.10}$$

As discussed in section 3.5, the processes states for this HMM are represented by the mixed states $\left(\frac{2}{3}, \frac{1}{3}\right)$, $(1, 0)$ and $(0, 1)$. Let

$$M = \begin{pmatrix} 1 & 0 \\ 2 & -1 \end{pmatrix}, \tag{4.11}$$

which is an invertible, unit-sum matrix. Note that $M^{-1} = M$. Now, when we perform the change of basis, we get

$$\tau = \pi M^{-1} = \left(\tfrac{4}{3}, -\tfrac{1}{3}\right), \tag{4.12}$$

$$U^0 = M T^0 M^{-1} = \begin{pmatrix} 1 & -\frac{1}{2} \\ 2 & -1 \end{pmatrix}, \text{ and} \tag{4.13}$$

$$U^1 = M T^2 M^{-1} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{pmatrix}. \tag{4.14}$$

The new coordinates for the process states, which are the images of the mixed states under multiplication (on the right) by M^{-1}, are $\left(\frac{4}{3}, -\frac{1}{3}\right)$, $(1, 0)$, and $(2, -1)$.

How can we make sense of $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$? If we try to think of $1' \in V'$ as a state in some variation on a Markov chain, it makes no sense at all — this casts $\sum_k U^k$

in the role of a transition matrix, so we find ourselves looking at negative transition probabilities. If we insist on this interpretation, then we must reject the whole idea of GHMMs as absurd. But if we think of $1'$ as an vector in a basis for $\mathcal{U}$, there is no substantial difficulty. A row of U^k simply gives the coordinates of the image of some basis element under a linear mapping, and a negative coordinate is a perfectly sensible thing. Instead of contemplating possible meanings for negative probabilities, we simply stop interpreting the matrix entries U^k_{ij} as probabilities. Essentially, we attribute meaning to the entire matrix U^k — it is a linear map — but not to individual entries in this matrix. (We will continue to use the term *mixed state*, although it is no longer apt.)

Definition 4.1.2. A *Proto-Generalized Hidden Markov Model (Proto-GHMM)* is a quadruple $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$, where V and $\mathcal{X}$ are finite sets, and each T^k is a $|V| \times |V|$ matrix, and the following conditions are satisfied:

1. $\sum_k T^k$ is a unit-sum matrix*,
2. π is a unit-sum vector, and
3. $\pi = \pi \sum_k T^k$.

We would like to have a Proto-GHMM define a process in the same way that an HMM does, but this does not always happen, because of a complication introduced by allowing negative entries in the T^ks. Proto-GHMMs $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and words w exist such that $\pi T^w \vec{1} < 0$. An example is

$$\begin{gathered} V = \left\{0', 1'\right\},\ \mathcal{X} = \{0, 1\},\ \pi = \left(\frac{1}{2}, \frac{1}{2}\right) \\ T^0 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix},\ T^1 = \begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix}, \end{gathered} \tag{4.15}$$

for which $\pi T^0 \vec{1} = -1$. Clearly, then, a Proto-GHMM may fail to define a process. The next two definitions address this problem.

Definition 4.1.3. A Proto-GHMM $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ is *valid* if, for all words $w \in \mathcal{X}^*$, it satisfies $\pi T^w \vec{1} \geq 0$.

---

*The reader may wonder why we require $\sum_k U^k$ to be unit-sum, when we could discard this restriction and have greater generality. This extra generality costs us some convenience. For example, the fact that π and $\sum_k T^k$ are unit-sum guarantees that $\sum_k \mathbf{P}(k) = 1$. We can work around such difficulties, but there is no point — as we will see in chapter 5. Every process which could possibly be represented with this greater generality has a GHMM presentation.

Definition 4.1.4. A *Generalized Hidden Markov Model (GHMM)* is a valid Proto-GHMM.

This definition is precisely what we need in order to have GHMMs represent processes.

Proposition 4.1.5. Every GHMM defines a process.

Proof. We prove this by applying B.1.1, where $f(w) = \pi T^w \vec{1}$. Thus we must verify

1. $f(\lambda) = 1$, and
2. for all words $w \in \mathcal{X}^*$,

$$f(w) = \sum_{z \in \mathcal{X}} f(zw) = \sum_{z \in \mathcal{X}} f(wz). \tag{4.16}$$

Unlike previous applications of B.1.1, here we must also show that $f : \mathcal{X}^* \to [0, 1]$.

First, $f(\lambda) = \pi T^\lambda \vec{1}$, where T^λ is the identity matrix by definition and π is unit-sum. So clearly, $f(\lambda) = 1$. Next, we deal with $f(wz)$:

$$\begin{aligned} \sum_{z \in \mathcal{X}} f(wz) &= \sum_{z \in \mathcal{X}} \pi T^w T^z \vec{1} \\ &= \pi T^w \left(\sum_{z \in \mathcal{X}} T^z \right) \vec{1}. \end{aligned} \tag{4.17}$$

But $\sum_{z \in \mathcal{X}} T^z$ is a unit-sum matrix, so this becomes

$$\sum_{z \in \mathcal{X}} f(wz) = \pi T^w \vec{1} = f(w). \tag{4.18}$$

The other equality in equation 4.16 may be handled by a similar calculation, using $\pi = \pi \sum_{z \in \mathcal{X}} T^z$ in place of the unit-sum property.

Finally, we need to show that for an arbitrary word $w \in \mathcal{X}^*$, $0 \leq f(w) \leq 1$. Half of this is given by validity. Given the properties we have just shown, a simple induction argument establishes that for all l,

$$\sum_{s \in \mathcal{X}^l} f(s) = 1. \tag{4.19}$$

Let $l = |w|$, and rewrite 4.19 as

$$1 = \sum_{s \in \mathcal{X}^l} f(s) = f(w) + \sum_{s \in \mathcal{X}^l,\, s \neq w} f(s). \tag{4.20}$$

Both terms in this sum are nonnegative, so neither can be greater than 1.■

We can characterize the preceding development as follows: A Proto-GHMM is an object which would be an HMM if it didn't have negative entries in its matrices, and a GHMM is a Proto-GHMM which never assigns a negative number to a word, and thus defines a process. That is, we are allowing negative entries in transition matrices, but only when the result works with the procedures we use for HMMs.

Testing the validity of a Proto-GHMM is nontrivial. Consider the obvious, naive algorithm: Take the (countably infinite) list of all words in $\mathcal{X}^*$, and write a loop which computes $\pi T^w \vec{1}$ for every word w on the list. Make the loop halt if this quantity is negative, and continue down the list if it is not. The Proto-GHMM is valid if and only if the loop never halts. Clearly this is not a practical test. One can find improvements to this algorithm which prune this list and thus typically reach invalid conclusions faster. And there are some special cases in which validity can be established — for instance, every HMM is a (valid) GHMM. Nevertheless, the essence of the test in the general case remains the same.

We have now finished defining GHMMs, and we will present some results involving them. The first of these is an extension of theorem 3.6.1 to GHMMs. The proof of 3.6.1 will serve as a proof of 4.1.6 without modification, so we will not give a separate proof here. Recall that $\mathcal{W}$ is the set of all signed measures on the future.

Proposition 4.1.6. Given a process $\mathcal{P}$, let $\mathcal{U}$ be the subset of $\mathcal{W}$ spanned by the reachable process states. If $\mathcal{P}$ has a GHMM presentation $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$, then $\dim(\mathcal{U}) \leq |V|$.

We conclude this section by restating and expanding on the basis change manipulation we performed earlier in this section. We begin with the following result.

Proposition 4.1.7. If $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ is a GHMM and M is an invertible unit-sum matrix, and V' is any set of size $|V|$, then $\left(V', \mathcal{X}, \left\{MT^kM^{-1}\right\}, \pi M^{-1}\right)$ is a GHMM which defines the same process as $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$.

Proof. It may easily be verified that $\left(V', \mathcal{X}, \left\{MT^kM^{-1}\right\}, \pi M^{-1}\right)$ is a Proto-GHMM. And by the same arguments used above for conjugation of HMMs, we know that for any $w \in \mathcal{X}^*$,

$$\tau U^w \vec{1} = \pi T^w \vec{1}. \tag{4.21}$$

The right side of equation 4.21 is always nonnegative, so the left side must be also. Therefore $\left(V', \mathcal{X}, \left\{MT^kM^{-1}\right\}, \pi M^{-1}\right)$ is valid, and thus it is also a GHMM.■

Definition 4.1.8. We say that two Proto-GHMMs $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$ are *conjugate* to each other by an invertible unit-sum matrix M if

1. $|V| = |V'|$,
2. $\tau = \pi M^{-1}$, and
3. for all k, $U^k = MT^kM^{-1}$.

Note that this is a linear conjugacy, which is the only kind of conjugacy we will consider.

Proposition 4.1.7 tells us, then, that if a Proto-GHMM is conjugate to a GHMM, then it is itself a GHMM.

Definition 4.1.9. When two GHMMs define the same process, we say that they are *equivalent*.

Thus the proof of proposition 4.1.7 also shows that if two GHMMs are conjugate, then they are equivalent.

Lemma 4.1.10. If two GHMMs $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$, are conjugate by an invertible unit-sum matrix M, then for all $w \in \mathcal{X}^*$,

$$\tau U^w M = \pi T^w. \tag{4.22}$$

Note that since these two HMMs are equivalent, we know that $\pi T^w \vec{1} = \tau U^w \vec{1}$. So if we divide equation 4.22 by this quantity, we get

$$\frac{\tau U^w}{\tau U^w \vec{1}} M = \frac{\pi T^w}{\pi T^w \vec{1}}, \tag{4.23}$$

which we may recognize as

$$N(\tau U^w) M = N(\pi T^w). \tag{4.24}$$

That is, M takes mixed states to mixed states.

Proof. We are given that $U^0 = MT^0M^{-1}$, $U^1 = MT^1M^{-1}$, and $\tau = \pi M^{-1}$. Multiply by M, and we have $MT^0 = U^0M$, $MT^1 = U^1M$, and $\pi = \tau M$. Thus, for all w,

$$\begin{aligned} \tau U^w M &= \tau U^{w_1} \dots U^{w_l} M \\ &= \tau U^{w_1} \dots U^{w_{l-1}} M T^{w_l} \\ &\vdots \\ &= \tau M T^{w_1} \dots T^{w_l} \\ &= \pi T^w. \quad \blacksquare \end{aligned} \tag{4.25}$$

The converse of proposition 4.1.7 does not hold — there are pairs of GHMMs which are equivalent but not conjugate. This is caused by redundancy — extra states in the presentation. In fact, there are pairs of such presentations which are both HMMs. Consider

$$\begin{aligned} V = \{0,1,2\},\ \mathcal{X} = \{0,1\},\ \pi = \left(\tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{3}\right) \\ T^0 = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \end{pmatrix},\ T^1 = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 \\ \frac{1}{2} & 0 & 0 \end{pmatrix}, \end{aligned} \tag{4.26}$$

and

$$\begin{aligned} V' = \{0', 1', 2'\},\ \mathcal{X} = \{0,1\},\ \pi = \left(\tfrac{2}{9}, \tfrac{1}{3}, \tfrac{4}{9}\right) \\ U^0 = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \end{pmatrix},\ U^1 = \begin{pmatrix} 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 \\ \frac{1}{2} & 0 & 0 \end{pmatrix}. \end{aligned} \tag{4.27}$$

These are both presentations of the Golden Mean Process, which are redundant in different ways. The reader may see this by noticing that states 0 and 2 — and $0'$ and $2'$ — have the same future conditional distributions. Thus, 0 and 2 — and $0'$ and $2'$ —can be merged.

To show that these presentations are not conjugate, we will show that there exists no invertible unit-sum matrix A which takes πT^w to τU^w for all $w \in \mathcal{X}^*$. Lemma

w	πT^w	τU^w
0	$\left(0, \frac{1}{3}, 0\right)$	$\left(0, \frac{1}{3}, 0\right)$
01	$\left(0, 0, \frac{1}{3}\right)$	$\left(0, 0, \frac{1}{3}\right)$
011	$\left(\frac{1}{6}, 0, 0\right)$	$\left(\frac{1}{6}, 0, 0\right)$
0111	$\left(\frac{1}{12}, 0, 0\right)$	$\left(0, 0, \frac{1}{12}\right)$

Table 4.1 The actions of the HMMs given in equations 4.26 and 4.27 on selected words.

4.1.10 tells us that each row of table 4.1 constrains the matrix A. These constraints are incompatible; the first three rows imply that A is the identity matrix, and the fourth implies that it is not. Thus, no suitable matrix A exists, so these presentations are not conjugate.

4.2 Redundancy and Linear Algebra

In the last section we saw an example in which two equivalent presentations may fail to be conjugate to one another, because they are redundant in different ways. We will now study redundancy, and then return in the next section to GHMM equivalence. The methods we will develop here give us a new way of describing the essential information in a GHMM. In this new form, we will be able to identify and factor out redundancy, which is the key to resolving the equivalence and minimization problems.

Vector Spaces We begin by identifying two vector spaces $\mathcal{H}$ and $\mathcal{F}$, which we will call the history and future spaces. Given a GHMM $\left(V, \mathcal{X}, \{T^k\}, \pi\right)$, let $\mathcal{H}$ be the span of the set of all mixed states:

$$\mathcal{H} = \text{span}\{N(\pi T^w) | w \in \mathcal{X}^*\}. \tag{4.28}$$

In a complementary fashion, let

$$\mathcal{F} = \text{span}\left\{T^s \vec{1} | s \in \mathcal{X}^*\right\}. \tag{4.29}$$

Elements of $\mathcal{H}$ are linear combinations of mixed states, which are row vectors, and elements of $\mathcal{F}$ are column vectors. Just as $N(\pi T^w)$ contains all the information about the history suffix w that is relevant to the future, $T^s \vec{1}$ contains all the information about the future that is relevant to the past. Implicit in this is that π and $\vec{1}$ play analogous roles which is suggested by the identities $\pi \sum_k T^k = \pi$ and $\sum_k T^k \vec{1} = \vec{1}$. Just as we may use πT^w to calculate the conditional distributions on the future induced by w, we may use $T^s \vec{1}$ to calculate the conditional distributions on the past induced by s. Thus, we may think of $T^s \vec{1}$ as a backward analog of a process state, and we may think of any $f \in \mathcal{F}$ as a linear combination of these.

Let $h = N(\pi T^w)$ and $f = T^s \vec{1}$. If we take the product of these, we get $hf = \mathbf{P}(s|w)$. In considering the product hf, we may think of h as the linear functional and f as the

operand, or vice versa. In general, if $h \in \mathcal{H}$ and $f \in \mathcal{F}$, then we may think of h as a map $h : \mathcal{F} \to \mathbb{R}$ defined by $h(f) = hf$, and we may think of f as a map $f : \mathcal{H} \to \mathbb{R}$ given by $f(h) = hf$. Thus $\mathcal{H}$ and $\mathcal{F}$ are almost each other's dual spaces.

But $\mathcal{H}$ and $\mathcal{F}$ may not be each other's duals. If there is redundancy in the presentation states — that is, if there are distinct mixed states which induce the same conditional future — then it may happen that $\mathcal{H}$ and $\mathcal{F}$ have different dimensions, and there may be nonzero $h \in \mathcal{H}$ for which $hf = 0$ for all $f \in \mathcal{F}$.

Let

$$K_{\mathcal{F}} = \{h \in \mathcal{H} | \text{ for all } f \in \mathcal{F},\ hf = 0\}, \tag{4.30}$$

and similarly,

$$K_{\mathcal{H}} = \{f \in \mathcal{F} | \text{ for all } h \in \mathcal{H},\ hf = 0\}. \tag{4.31}$$

That is, $h \in \mathcal{H}$ is in $K_{\mathcal{F}}$ if it is in the kernel of every $f \in \mathcal{F}$. If $h = \pi T^w$ is a row vector induced by a word w and $h \in K_{\mathcal{F}}$, we usually have $h = (0, \ldots, 0)$ and $\mathbf{P}(w) = 0$. In an HMM, this is the only way a row vector induced by a word may fall in $K_{\mathcal{F}}$. But differences between mixed states may lie in $K_{\mathcal{F}}$. Suppose two words w_1 and w_2 induce the same process state, $\mathbf{P}(\cdot|w_1) = \mathbf{P}(\cdot|w_2)$, and let $h_1 = N(\pi T^{w_1})$ and $h_2 = N(\pi T^{w_2})$. Then for all words s,

$$h_1 T^s \vec{1} = \mathbf{P}(s|w_1) = \mathbf{P}(s|w_2) = h_2 T^s \vec{1}. \tag{4.32}$$

Because every $f \in \mathcal{F}$ is a linear combination of column vectors $T^s\vec{1}$, this implies $h_1 f = h_2 f$ for all $f \in \mathcal{F}$, or $h_1 - h_2 \in K_{\mathcal{F}}$. If we know that the current history suffix is either w_1 or w_2, but we do not know which one, then our finding out which one does not improve our ability to predict the future. And this works backward, too — if $h_1 - h_2 \in K_{\mathcal{F}}$, then w_1 and w_2 induce the same process state. $K_{\mathcal{F}}$ contains exactly those vectors which are irrelevant to the future of the process. For this reason we say that $K_{\mathcal{F}}$ consists of redundancy.

Similar statements are true about $K_{\mathcal{H}}$. If, and only if, $f_1, f_2 \in \mathcal{F}$ and $hf_1 = hf_2$ for all $h \in \mathcal{H}$, then $f_1 - f_2 \in K_{\mathcal{H}}$, and the distinction between f_1 and f_2 is independent of the history of the process.

Now we can eliminate this redundancy — factor it out, so to speak — by working with the quotient spaces $\mathcal{H}/K_{\mathcal{F}}$ and $\mathcal{F}/K_{\mathcal{H}}$ in place of $\mathcal{H}$ and $\mathcal{F}$. As the following

lemma shows, $\mathcal{H}/K_{\mathcal{F}}$ is in one-to-one correspondence with $\mathcal{U}$ the span of the reachable process states, so it contains no redundancy.

Lemma 4.2.1. *$\mathcal{H}/K_{\mathcal{F}}$ is isomorphic to $\mathcal{U}$.*

Proof. Let ψ be the quotient map $\psi : \mathcal{H} \to \mathcal{H}/K_{\mathcal{F}}$, and let $M : \mathcal{H} \to \mathcal{U}$ be defined as follows: if $h \in \mathcal{H}$, then $M(h)$ is the signed measure on the future space given by $M(h)(s) = hT^s\vec{1}$ for all $s \in \mathcal{X}^*$. Now we can define our isomorphism $\phi : \mathcal{H}/\mathcal{K}_{\mathcal{F}} \to \mathcal{U}$. If $g \in \mathcal{H}/K_{\mathcal{F}}$, choose $h \in \psi^{-1}(g)$ and define $\phi(g) = M(h)$. The value of $\phi(g)$ does not depend on our choice of h, because if $h_1, h_2 \in \psi^{-1}(g)$, then $h_1 - h_2 \in K_{\mathcal{F}}$. This implies that $h_1T^s\vec{1} = h_2T^s\vec{1}$ for all s, so $M(h_1) = M(h_2)$.

Because ψ and M are linear, ϕ must be linear. To show that it is an isomorphism, we must show that it is injective and surjective. The first of these is trivial: suppose $\phi(g_1) = \phi(g_2)$. If we choose $h_1 \in \psi^{-1}(g_1)$ and $h_2 \in \psi^{-1}(g_2)$, we have $M(h_1) = M(h_2)$, which is equivalent to $h_1T^s\vec{1} = h_2T^s\vec{1}$ for all s. Thus $h_1 - h_2 \in K_{\mathcal{F}}$, so $g_1 = g_2$.

Next we show that ϕ is surjective. The definition of $\mathcal{U}$ implies that there must exist words $w_1, \ldots, w_n$ such that the process states $\mathbf{A}_1 = \mathbf{P}(\cdot|w_1), \ldots, \mathbf{A}_n = \mathbf{P}(\cdot|w_n)$ form a linearly independent basis for $\mathcal{U}$. That is, there are no real numbers $a_1, \ldots, a_n$, such that

$$(a_1\mathbf{A}_1 + \ldots + a_n\mathbf{A}_n)(s) = 0 \text{ for all } s. \tag{4.33}$$

For $i = 1, \ldots, n$, let $g_i = \psi(N(\pi T^{w_i}))$. For all i and for all s, we have

$$\begin{aligned} \phi(g_i)(s) &= M(N(\pi T^{w_i}))(s) \\ &= \mathbf{P}(s|w) \\ &= \mathbf{A}(s), \end{aligned} \tag{4.34}$$

so $\phi(g_i) = \mathbf{A}_i$. Thus if $g_1, \ldots, g_n$ were linearly dependent, $\mathbf{A}_1, \ldots, \mathbf{A}_n$ would have to be linearly dependent as well. So $g_1, \ldots, g_n$ are a linearly independent basis for some subspace of $\mathcal{H}/K_{\mathcal{F}}$ and ϕ is a map which takes this basis to a basis for $\mathcal{U}$. This means that ϕ is an isomorphism from this subspace onto $\mathcal{U}$. But ϕ is injective and takes all of $\mathcal{H}/K_{\mathcal{F}}$ to $\mathcal{U}$, hence this subspace is all of $\mathcal{H}/K_{\mathcal{F}}$.■

An element of $\mathcal{H}/K_{\mathcal{F}}$ is a set of row vectors, differences between which lie in $\mathcal{K}_{\mathcal{F}}$. Similarly an element of $\mathcal{F}/K_{\mathcal{F}}$ is a set of column vectors which is parallel to $K_{\mathcal{H}}$. What does a linear functional — a real valued linear function — on $\mathcal{H}/K_{\mathcal{F}}$ look like? It is

simply a linear functional on $\mathcal{H}$ which is constant along directions which lie in $K_{\mathcal{F}}$. Every $f \in \mathcal{F}$ is a linear functional on $\mathcal{H}$ which is zero on all of $K_{\mathcal{F}}$, so every $f \in \mathcal{F}$ is a linear functional on $\mathcal{H}/K_{\mathcal{F}}$. Note that $f_1, f_2 \in \mathcal{F}$ represent the same functional on $\mathcal{H}$ — and thus on $\mathcal{H}/K_{\mathcal{F}}$ — if and only if f_1 and f_2 differ by an element of $K_{\mathcal{H}}$. So all of the column vectors in any $e \in \mathcal{F}/K_{\mathcal{H}}$ represent the same linear functional on $\mathcal{H}/K_{\mathcal{F}}$. So we may say that e is that functional. And if $g \in \mathcal{H}/K_{\mathcal{F}}$, we have $ge = hf$ for any $h \in \psi^{-1}(g)$ and for any $f \in \mathcal{F}$ taken by the quotient map to e. Likewise, each $g \in \mathcal{H}/K_{\mathcal{F}}$ is a unique linear functional on $\mathcal{F}/K_{\mathcal{H}}$. So $\mathcal{H}/K_{\mathcal{F}}$ and $\mathcal{F}/K_{\mathcal{H}}$ like $\mathcal{H}$ and $\mathcal{F}$, consist of elements of each other's dual spaces. But unlike elements of $\mathcal{H}$ and $\mathcal{F}$, elements of $\mathcal{H}/K_{\mathcal{F}}$ and $\mathcal{F}/K_{\mathcal{H}}$ contain no redundancy.

Lemma 4.2.2. $\mathcal{H}/K_{\mathcal{F}}$ and $\mathcal{F}/K_{\mathcal{H}}$ are each other's dual spaces.

Proof. We have seen that every $e \in \mathcal{F}/K_{\mathcal{H}}$ is a unique linear functional on $\mathcal{H}/K_{\mathcal{F}}$. Thus, to show that $\mathcal{F}/K_{\mathcal{H}}$ is the dual of $\mathcal{H}/K_{\mathcal{F}}$, we need only show that $\mathcal{F}/K_{\mathcal{H}}$ contains the entire dual space rather than a proper subspace. Similarly, we must show that $\mathcal{H}/K_{\mathcal{F}}$ contains the entire dual of $\mathcal{F}/K_{\mathcal{H}}$.

Let n be the dimension of $\mathcal{H}/K_{\mathcal{F}}$, and let n be the dimension of $\mathcal{F}/K_{\mathcal{H}}$. Let $g_1, \ldots, g_n$ be a linearly independent basis for $\mathcal{H}/K_{\mathcal{F}}$ and let $e_1, \ldots, e_m$ be a linearly independent basis for $\mathcal{F}/K_{\mathcal{H}}$. Finally, let A be the $n \times m$ matrix with entries $A_{ij} = g_i e_j$. Suppose a linear combination $a_1 g_1 + \ldots a_n g_n$ is taken to zero by all of $e_1, \ldots, e_m$. It must be taken to zero by every $e \in \mathcal{F}/K_{\mathcal{H}}$, and hence $(a_1 g_1 + \ldots a_n g_n)f = 0$ for every $f \in \mathcal{F}$. This is possible only if $a_1 g_1 + \ldots a_n g_n = 0$, so the rows of A are linearly independent and A has rank n. A similar argument shows that A has rank m, so $n = m$. Hence $\mathcal{H}/K_{\mathcal{F}}$ and $\mathcal{F}/K_{\mathcal{H}}$ have the same dimension, so neither can be a proper subspace of the other's dual space. Each must be the dual space of the other. ■

Bases We will now move on to more concrete and more readily manipulated forms of these vector spaces. In particular, we will be working with bases for subspaces of $\mathcal{H}$ and $\mathcal{F}$ that are isomorphic to $\mathcal{H}/K_{\mathcal{F}}$ and $\mathcal{F}/K_{\mathcal{H}}$, respectively. In addition, we will want to be able to refer to our basis vectors in a way that does not depend on any basis derived from a presentation. Thus we will choose basis vectors which are associated with words.

Definition 4.2.3. A *wordlist* W of length l is a finite list (ordered set) of words $w_1, \ldots, w_l \in \mathcal{X}^*$.

Given a wordlist W of length l and a GHMM $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$, we define the $l \times |V|$ matrix H as follows: the ith row of H is $N(\pi T^{w_i})$. We call W the *history wordlist* and H the *history matrix*. Similarly, given a wordlist S of length m, which we will call the *future wordlist*, we define the *future matrix* F to be the $|V| \times m$ matrix whose ith column is $T^{s_i}\vec{1}$. Note that H and F are functions of W and S, respectively. We will sometimes write $H(W)$ and $F(S)$ to avoid ambiguity. We will use $\overline{H}$ and $\overline{F}$ to denote the span of the rows of H and the columns of F, respectively.

Because we want $\overline{H}$ to contain a representation of every process state, we are interested in wordlists which induce a sufficiently large basis.

Definition 4.2.4. A history wordlist W is *sufficient* for a given GHMM, or simply sufficient if the span of the rows of $H(W)$ satisfies

$$\text{span}\left\{\overline{H} \cup K_{\mathcal{F}}\right\} = \mathcal{H}. \tag{4.35}$$

Similarly, a future wordlist S is sufficient if the span of the columns of $F(S)$ satisfies $\text{span}\left\{\overline{F} \cup K_{\mathcal{H}}\right\} = \mathcal{F}$.

This definition means that a history wordlist, for example, is sufficient if the rows of H, mapped into $\mathcal{H}/K_{\mathcal{F}}$ by the quotient map ψ, form a basis for $\mathcal{H}/K_{\mathcal{F}}$.

It is possible for a sufficient wordlist to contain words which are not needed for sufficiency. As we will see shortly, removing such words is desirable.

Definition 4.2.5. A history wordlist is *minimal* if it is sufficient and it has length $l = \dim\{\mathcal{H}\} - \dim\{K_{\mathcal{F}}\}$. Similarly, a future wordlist is minimal if it is sufficient and it has length $m = \dim\{\mathcal{F}\} - \dim\{K_{\mathcal{H}}\}$.

If W and S are minimal history and future wordlists, then $\overline{H}$ and $K_{\mathcal{F}}$ are complimentary subspaces of $\mathcal{H}$, and $\overline{F}$ and $K_{\mathcal{H}}$ are complimentary subspaces of $\mathcal{F}$.

Proposition 4.2.6. If W is a minimal history wordlist, then $\overline{H}$ is isomorphic to $\mathcal{H}/K_{\mathcal{F}}$. Similarly, if S is a minimal future wordlist then $\overline{F}$ is isomorphic to $\mathcal{F}/K_{\mathcal{H}}$. In both cases, the restriction of the quotient map — to $\overline{H}$ and $\overline{F}$ as appropriate — is a suitable isomorphism.

Proof. If W is a wordlist of length l, then $\dim\{\overline{H}\} \leq l$. But if W is sufficient, then $\overline{H}$ and $K_{\mathcal{F}}$ together span $\mathcal{H}$, so

$$\dim\{\overline{H}\} + \dim\{K_{\mathcal{F}}\} \geq \dim\{\mathcal{H}\}. \tag{4.36}$$

If W is also minimal, this implies $\dim\{\overline{H}\} \geq l$, so we have $\dim\{\overline{H}\} = l$ and

$$\dim\{\overline{H}\} + \dim\{K_{\mathcal{F}}\} = \dim\{\mathcal{H}\}. \tag{4.37}$$

Thus the rows of H must be linearly independent. Further, $\overline{H}$ must be independent of $K_{\mathcal{F}}$, so we can write

$$\mathcal{H} = \overline{H} \oplus K_{\mathcal{F}}. \tag{4.38}$$

Thus $\overline{H}$ is complementary to $K_{\mathcal{F}}$, and the restriction of the quotient map to $\overline{H}$ is an isomorphism between $\overline{H}$ and $\mathcal{H}/K_{\mathcal{F}}$. A similar proof holds for S, $\overline{F}$, and $\mathcal{F}/K_{\mathcal{H}}$.■

Proposition 4.2.7. If W and S are minimal wordlists, then

1. $|W| = |S|$,
2. rank(H) = rank(F),
3. dim$(\overline{H})$ = dim$(\overline{F})$, and
4. $\overline{H}$ and $\overline{F}$ are each other's dual spaces.

Proof. $\overline{H}$ is isomorphic to $\mathcal{H}/K_{\mathcal{F}}$, and $\overline{F}$ is isomorphic to $\mathcal{F}/K_{\mathcal{H}}$. Lemma 4.2.2 tells us that $\mathcal{H}/K_{\mathcal{F}}$ and $\mathcal{F}/K_{\mathcal{H}}$ are dual to each other, and we know that these isomorphisms preserve the products of elements. This $\overline{H}$ and $\overline{F}$ must be each other's dual spaces, which proves (4). This, in turn, implies (1), (2), and (3).■

We began this section by defining the vector spaces $\mathcal{H}$ and $\mathcal{F}$ so that elements of $\mathcal{H}$ represent conditional distributions on the future, and elements of $\mathcal{F}$ represent conditional distributions on the past. But $\mathcal{H}$ and $\mathcal{F}$ contain subspaces of redundancy, $K_{\mathcal{F}}$ and $K_{\mathcal{H}}$. Now, if W and S are sufficient, we have vector spaces $\overline{H}$ and $\overline{F}$, the elements of which still encode the same set of conditional distributions, and we have bases H and F for $\overline{H}$ and $\overline{F}$, respectively. In addition, if W and S are minimal, $\overline{H}$ and $\overline{F}$ contain no redundancy and H and F are linearly independent bases.

Note that we have now shown that if our history wordlist is minimal, $\overline{H}$ is isomorphic to $\mathcal{U}$. In particular, we have a natural map from one to the other: If $\mathbf{A} \in \mathcal{U}$, then there

is a unique $h \in \overline{H}$ such that for all $s \in \mathcal{X}^*$, we have $\mathbf{A}(s) = hT^s\vec{1}$. That is, if h is a mixed state, it is the mixed state representation for $\mathbf{A}$. (This h may be found by the techniques used in the proof of lemma 4.2.1, which involve building a basis of process states and a corresponding basis of mixed states induced by the same words.) Thus $\overline{H}$ may be thought of as a version of $\mathcal{U}$ consisting of tangible vectors of real numbers rather than the more abstract signed measures on an infinite sequence space.

As the following proposition establishes, if W and S are sufficient, then the matrices H and F, like the vector spaces $\mathcal{H}$ and $\mathcal{F}$, can tell us whether or not a distinction is independent of the past or of the future. And if W and S are minimal, then H and F contain no redundancy, in the sense that no row vector in the span of H is independent of the future, and likewise for F. We will work with minimal wordlists whenever we can because the absence of redundancy makes it easier to determine whether or not two processes states are distinct.

Proposition 4.2.8. If W and S are sufficient, $f \in \mathcal{F}$, and $h \in \mathcal{H}$, then

1. $f \in K_{\mathcal{H}}$ if and only if $Hf = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$, and
2. $h \in K_{\mathcal{F}}$ if and only if $hF = (0, \ldots, 0)$.

Proof. If $f \in K_{\mathcal{H}}$ and h is a row of H, then $h \in \mathcal{H}$, so $hf = 0$. If $f \notin K_{\mathcal{H}}$, then there is some $h \in \mathcal{H}$ such that $hf \neq 0$. Any vector in $\mathcal{H}$ can be written as $h = cH + k$, a linear combination of the rows of H plus some $k \in K_{\mathcal{F}}$. Thus we have

$$0 \neq hf = cHf + kf, \tag{4.39}$$

where we know that $kf = 0$. So this becomes $0 \neq cHf$, which implies $Hf \neq \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$. This proves (1), and a virtually identical argument proves (2)■

Recall that each row of H is associated with a particular word: the ith row of H is the mixed state $N(\pi T^{w_i})$, where w_i is the ith element of W. Similarly, the jth column of F is $T^{s_j}\vec{1}$, where s_j is the jth element of S. Thus HF is the matrix of conditional probabilities given by

$$(HF)_{ij} = N(\pi T^{w_i})T^{s_j}\vec{1} = \mathbf{P}(s_j|w_i). \tag{4.40}$$

Also, because W and S have the same length, HF is square.

Corollary 4.2.9. If W and S are minimal, then HF is a nonsingular matrix. Conversely, if W and S are sufficient and one or both of W and S is not minimal, then HF is singular.

Proof. Suppose v is a row vector such that $vHF = (0, \ldots, 0)$. Then $(vH)F = (0, \ldots, 0)$, so $vH \in K_{\mathcal{F}}$. But $vH \in \overline{H}$, and since W is minimal, $\overline{H} \cap K_{\mathcal{F}} = \{(0, \ldots, 0)\}$. Thus we have $vH = (0, \ldots, 0)$. The rows of H are linearly independent, so we must have $v = (0, \ldots, 0)$. That is, if a linear combination of the rows of HF is the zero row, then all of the coefficients of the linear combination are zero. Hence the rows of HF are linearly independent. And HF is square, so it is nonsingular.

Conversely, assume that W is not minimal. The case in which S is not minimal may be treated similarly. Let W' be a minimal wordlist which is a subset of W, and let H' be the history matrix induced by W'. Now we consider two cases. In the first, the rows of H span the same space as the rows of H'. This means that the dimension of $\overline{H}$ must be less than the number rows in H. Thus the rows of H must be linearly dependent, which implies that the rows of HF are linearly dependent. In the second, $\overline{H}$ is larger than the span of the rows of H'. But the span of the rows of H' is isomorphic to $\mathcal{H}/K_{\mathcal{F}}$. Any larger subspace of $\mathcal{H}$ must contain nonzero vectors which lie in $K_{\mathcal{F}}$ and which are sent to zero by F. Thus the row span of H is reduced by multiplication by F, which means that the rows of HF are linearly dependent.■

Wordlists In the next section, we will use H and F extensively, as they will be our primary tools for solving the equivalence and minimization problems. In the remainder of this section, we will discuss the construction of these matrices and the construction of the wordlists on which they depend.

The next lemma is a minor fact which we will need in section 4.3.

Lemma 4.2.10. If S is sufficient and $h \in K_{\mathcal{F}}$, then for any k in the alphabet $\mathcal{X}$, $hT^k \in K_{\mathcal{F}}$.

Proof. Our assumption that $h \in K_{\mathcal{F}}$ implies that for all words s, $hT^s\vec{1} = 0$. This means that for any k, for all words s, $hT^{ks}\vec{1} = 0$, simply because ks is a word. Thus we have $\big(hT^k\big)T^s\vec{1} = 0$ for all words s, which implies that $hT^k \in K_{\mathcal{F}}$.■

Two facts make the construction of sufficient wordlists feasible. First, $\mathcal{H}$ is a vector space spanned by the mixed states, which are vectors of length $|V|$, so a basis for $\mathcal{H}$ — or any subspace of $\mathcal{H}$ — need not have more than $|V|$ elements. Similarly, a basis for $\mathcal{F}$ need not have more than $|V|$ elements. So wordlists never need to be longer than $|V|$ in order to be sufficient. Second, we have the following fact of linear algebra.

Lemma 4.2.11. If $u \in$ span$\{v_1, \ldots, v_n\}$, and A is any matrix such that the product uA is defined, then $uA \in$ span$\{v_1A, \ldots, v_nA\}$.

Proof. There must exist numbers $c_1, \ldots, c_n$ such that $u = c_1v_1 + \ldots + c_nv_n$. But then $uA = c_1v_1A + \ldots + c_nv_nA$.■

We construct sufficient wordlists using the following algorithm, which is used for a somewhat different purpose in [8].

Algorithm 4.2.12. Let Q be a queue — a first-in, first-out list — Q of words, and let and let W be a list of words. Queue Q will store a list of words which the algorithm has determined it must examine, and W will store the developing wordlist.

1. Initialize Q to contain only the word λ, and initialize W to be empty.
2. Take a word z from the tail of Q and test whether of not $N(\pi T^z)$ lies in

$$\text{span}\{N(\pi T^w) | w \in W\} = \text{span}\{\text{rows of } H(W)\}. \tag{4.41}$$

3. If it does, discard z and skip forward to step 6. (Otherwise, continue with step 4.)
4. Add z to the wordlist W.
5. For each $x \in \mathcal{X}$, add the word zx to the head of Q.
6. If Q is not empty, go back to step 2.
7. Stop. Q is empty, and W contains the completed wordlist.

Algorithm 4.2.12 builds only history wordlists, but a virtually identical algorithm builds future wordlists.

Proposition 4.2.13. The wordlists constructed by algorithm 4.2.12 are sufficient. In fact, for these wordlists, $\mathcal{H} = \overline{H}$.

Proof. We need to show that if $h \in \mathcal{H}$, then $h \in \overline{H}$. Every $h \in \mathcal{H}$ is a linear combination of vectors of the form $N(\pi T^w)$ for some $w \in \mathcal{X}^*$, so it will suffice to consider vectors

of that form. Suppose there exists a word y such that $N(\pi T^y) \notin \overline{H}$. Let z be the longest prefix of y such that $N(\pi T^z) \in \overline{H}$. There is at least one such prefix, namely λ. Let x be the symbol which follows z in y — that is, choose $x \in \mathcal{X}$ so that zx is a prefix of y. Thus $N(\pi T^{zx})$ is not in $\overline{H}$.

Now, we have chosen z so that $N(\pi T^z)$ is in $\overline{H} = \text{span}\{N(\pi T^w)|w \in W\}$, but $N(\pi T^{zx})$ is not. By lemma 4.2.11, we know that $N(\pi T^z)T^x$ is in the span of $\{N(\pi T^w)T^x|w \in W\}$, or equivalently,

$$N(\pi T^{zx}) \in \text{span}\{N(\pi T^{wx})|w \in W\}. \tag{4.42}$$

Because the algorithm added each w to W, we know that it added the words wx to the queue. Thus the vectors $N(\pi T^{wx})$ were subsequently tested against the developing basis and added to the basis if it did not already span them. Hence we know that for all $w \in W$, the vectors$N(\pi T^{zx})$ lie in $\overline{H}$, so $N(\pi T^{zx})$ is a linear combination of vectors which are known to be in $\overline{H}$, thus it is itself in $\overline{H}$. This is a contradiction, hence no words y can exist.■

Sufficient future wordlists may be constructed by an essentially identical process.

Lemma 4.2.14. A GHMM $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ has sufficient wordlists W and S, every word of which has length less than $|V|$.

Proof. The construction we have just given has the property that if w is not added to W, then no words of the form wz can be added to W. Thus if w is in W, every prefix of w is in W as well, including the length zero word λ. This means that if a word of length l is added to a wordlist W, then W has at least one element of each of the lengths $0, 1, \ldots, l$ and thus contains at least $l + 1$ elements. Note that the wordlists constructed here never have more than $|V|$ elements because rows of H are linearly independent and span a subset of $\mathbb{R}^{|V|}$. Thus a word of length $|V|$ or more can never be added to W.

A virtually identical proof holds for S.■

Once we have sufficient wordlists, we can get minimal wordlists simply by extracting appropriate subsets. Suppose we have history and future wordlists W and S, we take $H(W)F(S)$ and delete every row which is linearly dependent on those which precede

it. The words associated with the remaining rows form a new wordlist W', and the remaining rows themselves form $H(W')F(S)$.

The rows of $H(W')F(S)$ are now linearly independent, which means the rows of $H(W')$ are independent. Moreover, because we only deleted linearly dependent rows, the row span of $H(W')F(S)$ is the same as that of $H(W)F(S)$. The properties of F are such that we can be sure $H(W')$ has the appropriate span. As we will show next, the resulting wordlist W' is minimal. The analogous construction, deleting columns instead of rows, builds us a minimal future wordlist S'.

Proposition 4.2.15. Let W and S be sufficient wordlists. If W' is a subset of W such that

1. the row span of $H(W')F(S)$ is the same as that of $H(W)F(S)$, and
2. the rows of $H(W')F(S)$ are linearly independent,

then W' is minimal. Similarly, if S' is a subset of S such that

1. the column span of $H(W)F(S')$ is the same as that of $H(W)F(S)$, and
2. the columns of $H(W)F(S')$ are linearly independent,

then S' is minimal.

Proof. As usual, we have separate statements about the past and the future, and we will only prove the one about the past, as essentially the same argument will serve for the future. We will not need to refer to $F(S')$, and will use $F = F(S)$.

We will first show that W' is sufficient, that is, that the rows of $H(W')$, together with $K_{\mathcal{F}}$, span $\mathcal{H}$.

Let $h_1, \ldots, h_n$ be the rows of $H(W)$. If $h \in \mathcal{H}$, then there exists a vector $a = (a_1, \ldots, a_n)$ such that

$$\begin{aligned} h &= a_1 h_1 + \ldots + a_n h_n + k \\ &= aH(W) + k \end{aligned} \tag{4.43}$$

for some $k \in K_{\mathcal{F}}$. Multiplying on the right by F, we have $hF = aH(W)F + kF$. We know that $kF = 0$, so hF is in the span of the rows of $H(W)F$. But the rows of $H(W')F$ have the same span, so there must exist some vector c such that $hF = cH(W')F$. This is equivalent to $(h - cH(W'))F = 0$, which means that $h - cH(W')$ is in $K_{\mathcal{F}}$. Thus

$h = cH(W') + k'$ for some $k' \in K_{\mathcal{F}}$. Thus $K_{\mathcal{F}}$ and the rows of $H(W')$ span $\mathcal{H}$, so W' is sufficient.

Showing that W' is minimal is now trivial. The rows of $H(W')F$ are linearly independent and multiplying by F cannot eliminate any linear dependency which is in the rows of $H(W')$. So the rows of $H(W')$ are linearly independent, and W' is minimal.■

In this section, we have studied the relationship between the past and the future in terms of linear algebra. As part of this study, we introduced and defined H and F, and showed how to construct them and their associated wordlists. Next, we will begin to use them to identify and eliminate redundancy from GHMM presentations.

4.3 Equivalence and Minimization of GHMMs

This section addresses two problems. The first of these, which we discussed in section 4.1, is the *identifiability* problem: When are two GHMMs equivalent? That is, when do two GHMMs define the same process? The second is the *minimization* problem: Given a GHMM, what GHMM is as small as possible — that is, has as few presentation states as is possible — but is equivalent to the given one? These two questions are closely related, and have been studied for HMMs for some time. The identifiability problem is the more famous of the two and was posed by Blackwell and Koopmans in 1957 and solved by Ito et al in 1992, by a methods similar to the one presented here.[2,7] The minimization problem is nearly solved in the same paper, and was completed by Vijay Balasubramanian.[8] The details of the method presented here are the work of the author. Notably, the standard presentation, which is important here and again in chapter 5, does not appear in any previous paper, though related presentations have appeared before beginning with [20].

Suppose we have a GHMM presentation $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ for a process $\mathcal{P}$ and wordlists W and S such that H and F are invertible. (This can only happen if there is no redundancy, that is, if $K_{\mathcal{F}}$ and $K_{\mathcal{H}}$ are trivial.) We are now going to define a sort of a canonical presentation for $\mathcal{P}$, which we will call the *standard presentation* for $\mathcal{P}$, which in this case is conjugate to $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ by H.

If we conjugate T^k by H, we get $B^k = HT^kH^{-1}$, and if we write H^{-1} as $F(HF)^{-1}$, we have

$$B^k = HT^kF(HF)^{-1}. \tag{4.44}$$

We will call the matrices B^k the *standard transition matrices* for the process $\mathcal{P}$ given W and S. As we know, each $(HF)_{ij}$ is simply $\mathbf{P}(s_j|w_i)$. Similarly, for all $k \in \mathcal{X}$ and for all $i, j \in V$,

$$\begin{aligned} \left(HT^kF\right)_{ij} &= N(\pi T^{w_i})T^kT^{s_j}\vec{1} \\ &= \mathbf{P}(ks_j|w_i). \end{aligned} \tag{4.45}$$

In words, HT^kF exhibits the action of the map T^k in terms of the bases — the rows of H and the columns of F — we have developed for the past and the future. And both $(HF)^{-1}$ and HT^kF are entirely determined by probabilities of words — by the process, rather than by the presentation. This fact is key to the algorithms of this section and the next chapter.

The same is true of the initial vector

$$\gamma = \pi H^{-1} = \pi F(HF)^{-1}, \tag{4.46}$$

since $(\pi F)_j = \mathbf{P}(s_j)$. We will refer to γ as the *standard initial vector* for $\mathcal{P}$ given W and S. Thus γ and B^k depend only on the wordlists and the process, and not on the GHMMs themselves.

The following definition assembles the standard transition matrices and the standard initial vector into a presentation. We may choose any set of size $|V|$ as the set of presentation states. It will be convenient to choose W because, as we will see shortly, the presentation states are associated with the words in W. Also, recall corollary 4.2.9, which tells us that HF is invertible.

Definition 4.3.1. If $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ is a GHMM presentation for the process $\mathcal{P}$ and W and S are wordlists such that the matrix HF is invertible, then we define the *standard presentation* for $\mathcal{P}$ given W and S to be

$$\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right), \tag{4.47}$$

where $B^k = HT^kF(HF)^{-1}$ and $\gamma = \pi H^{-1} = \pi F(HF)^{-1}$.

The following result establishes that if H and F are invertible, the standard presentation for $\mathcal{P}$ is in fact a presentation, and that it is a presentation for $\mathcal{P}$. In 4.3.10, we will establish that the standard presentation is a presentation for $\mathcal{P}$whenever HF is invertible, that is, whenever the standard presentation is defined.

Lemma 4.3.2. Let $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ be a GHMM presentation for the process $\mathcal{P}$ and let W and S be wordlists such that the matrices H and F are invertible. Then the standard presentation for $\mathcal{P}$ given W and S is a GHMM presentation for $\mathcal{P}$.

Proof. The standard presentation $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$ is conjugate to $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ by H. Given this fact, proposition 4.1.7 tells us that $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$ is a GHMM and that it is equivalent to $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$.■

Suppose we have a second GHMM $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$ which is equivalent to our first GHMM, $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$. If we take its history and future matrices H' and F' with respect to the same wordlists W and S, then we must have $H'F' = HF$, $H'U^kF' = HT^kF$, and $\tau F' = \pi F$. The for both presentations generate the same γ and the same set of B^ks and so they generate the same standard presentation. In this sense, the standard presentation plays a role similar to that of a canonical form. But because the standard presentation depends on the wordlists W and S, there is no single, absolute standard presentation. This is why we call it the *standard* presentation and not the *canonical* presentation.

In other words, suppose we have two equivalent GHMMs, and we have wordlists such that both GHMMs' history and future matrices are invertible. Then they must produce the same γ and B^k, and therefore they must both be conjugate to the standard presentation. We know that if two GHMMs are both conjugate to a third GHMM, then they must be equivalent. So using standard presentations shows promise of resolving the identifiability problem. The approach above for constructing the standard transition matrices will not work in general, because the assumption that H and F are invertible may fail. However, there is a generalization of the standard presentation which we can always construct, and so we can give a new solution to the identifiability problem using this generalization.

Theorem 4.3.3. Suppose we are given two GHMMs $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and

$\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$, and let W and S be history and future wordlists which are sufficient for both of them. Let $\mathbf{P}$ be the probability measure induced by $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and let $\mathbf{Q}$ be that induced by $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$. The GHMMs are equivalent — that is, $\mathbf{P} = \mathbf{Q}$ — if and only if all of the following hold:

1. for all $w_i \in W$ and $s_j \in S$, $\mathbf{P}(s_j|w_i) = \mathbf{Q}(s_j|w_i)$,
2. for all $w_i \in W$, $s_j \in S$, and $k \in \mathcal{X}$, $\mathbf{P}(ks_j|w_i) = \mathbf{Q}(ks_j|w_i)$, and
3. for all $s_j \in S$, $\mathbf{P}(s_j) = \mathbf{Q}(s_j)$.

We will prove this by showing that the probability of any word is determined by these few probabilities, these few conditional probabilities, and the information that W and S are sufficient. Thus the entire process is determined by these same few pieces of information. At this point, it is worth recalling Bayes rule — $\mathbf{P}(s|w) = \mathbf{P}(ws)/\mathbf{P}(w)$ — because it tells us that we can compute the necessary conditional probabilities from the (non-conditional) probabilities of all words w_i, s_j, w_is_j, and w_iks_j for all $w_i \in W$, $s_j \in S$, and $k \in \mathcal{X}$. We can ignore the possibility that $\mathbf{P}(w_i) = 0$ for some i, and hence that $\mathbf{P}(s_j|w_i)$ will not be well-defined, because the row of H corresponding to that w_i must lie in $K_{\mathcal{F}}$. Such a row is never needed in the basis.

The proof itself, which begins on page 79, uses a number of lemmas, most of which are proved by calculation and use of the properties of H and F. These lemmas develop a generalization of the standard presentation. Note that we are using W as a set of presentation states. Recall that we concluded in chapter 3 that the role of presentation states was to serve as basis vectors for the space containing the mixed states. Here, we use this the other way, and having chosen a basis for the mixed states, we will use the elements of that basis as presentation states. Arguably, the presentation states should be labeled with the mixed states, rather than the strings in W which induce those mixed states, but we will use the strings themselves for brevity and clarity.

Whenever two GHMMs share an alphabet†, it is always possible to find wordlists which are sufficient for both of them. If W_1 and W_2 are sufficient for $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$, respectively, then the wordlist $W = W_1 \cup W_2$, with words in any fixed order, will be sufficient for both $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$.

†Two GHMMs which have distinct alphabets always represent different processes. In some applications, however, it may be desirable to map one alphabet onto another so as to sidestep this fact.

When W and S are not minimal, and thus HF is not invertible, we can no longer define the standard transition matrices B^k. It is still possible, however, to define a form of the standard presentation, though we can no longer define it as simply as we can if W and S are minimal.

Definition 4.3.4. Given a GHMM presentation $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ of a process $\mathcal{P}$ and sufficient wordlists W and S, we say that $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$ is a *quasi-presentation* if

1. for each $k \in \mathcal{X}$, B^k satisfies $B^k HF = HT^k F$, and
2. γ satisfies $\gamma HF = \pi F$.

We refer to B^k and γ as a *quasi-transition matrix* and a *quasi-initial vector* respectively.

If HF is invertible, then there is a unique quasi-presentation for a process $\mathcal{P}$ given W and S, and it is the standard presentation. Otherwise, quasi-presentations are not unique. Lemma 4.3.5 shows that quasi-presentations exist.

Lemma 4.3.5. If W and S are sufficient, then for each k there exists a quasi-transition matrix B^k such that

$$B^k HF = HT^k F, \tag{4.48}$$

and there exists a quasi-initial vector γ such that $\gamma HF = \pi F$.

Proof. Let $h_i = N(\pi T^{w_i})$ be the ith row of H. Then the ith row of HT^k is $h_i T^k = N(\pi T^{w_i}) T^k$, which must lie in $\mathcal{H}$. The rows of H and elements of $K_{\mathcal{F}}$ span $\mathcal{H}$, so there is some $v \in K_{\mathcal{F}}$ and some $a \in \mathbb{R}^n$ such that $h_i T^k = aH + v$. This means that $h_i T^k F = aHF$. Let the ith row of B^k be a.

Likewise, π is in $\mathcal{H}$, so we can find $\gamma \in \mathbb{R}^n$ and $v \in K_{\mathcal{F}}$ such that $\gamma H + v = \pi$, and hence $\gamma HF = \pi F$.■

The next lemma depends on the normalization of the rows of H. In fact it is the reason we defined H — and indeed, $\mathcal{H}$ — using $N(\pi T^w)$ instead of the simpler πT^w.

Lemma 4.3.6. The matrix $\sum_{k \in \mathcal{X}} B^k$ and the vector γ are both unit-sum.

Proof. Because $\vec{1}$ is an element of $\mathcal{F}$, there exists an $a \in \mathbb{R}^n$ and an $u \in K_{\mathcal{H}}$ such that $\vec{1} = Fa + u$. Thus $H\vec{1} = HFa$, and since the rows of H are mixed states and so must

be unit-sum, we have $\vec{1} = HFa$. Similarly, for all $k \in \mathcal{X}$, we can write

$$HT^k\vec{1} = HT^kFa + HT^ku \tag{4.49}$$

But each row of HT^k is in $\mathcal{H}$, so $HT^ku = 0$ and we have $HT^k\vec{1} = HT^kFa$. Now, $\sum_k T^k$ is unit-sum, so if we sum on k, this becomes $\vec{1} = H\sum_k T^kFa$.

Summing 4.48 on k gives us

$$\left(\sum_{k\in\mathcal{X}} B^k\right)HF = H\left(\sum_{k\in\mathcal{X}} T^k\right)F. \tag{4.50}$$

If we multiply on the right by a, we can substitute $\vec{1}$ for HFa and for $H\sum_k T^kFa$, and we have $\left(\sum_k B^k\right)\vec{1} = \vec{1}$.

Similarly, $\gamma HF = \pi F$ becomes $\gamma HFa = \pi Fa$, which in turn becomes $\gamma\vec{1} = 1$.■

Lemma 4.3.6 establishes that $\left(W, \mathcal{X}, \{B^k\}, \gamma\right)$ satisfies conditions 1 and 2 of the three conditions of definition 4.1.2, which defines a Proto-GHMM. The next lemma tells us that it may fail to satisfy the final condition — $\gamma\sum_k B^k$ may differ from γ, but only by a vector in $K_{\mathcal{F}}$.

Lemma 4.3.7. For some vector r such that $rHF = 0$, γ satisfies $\gamma\sum_k B^k = \gamma + r$.

Proof. Because γ is known to satisfy $\gamma HF = \pi F$, there is a $v \in K_{\mathcal{F}}$ such that $\gamma H = \pi + v$. Thus,

$$\begin{aligned}\gamma H\sum_k T^k &= \pi\sum_k T^k + v\sum_k T^k \\ &= \pi + v'\end{aligned} \tag{4.51}$$

for some $v' \in K_{\mathcal{F}}$. Similarly, 4.50 tells us that

$$H\sum_k T^k = \sum_k B^kH + A \tag{4.52}$$

for some matrix A, all rows of which lie in $K_{\mathcal{F}}$. When we substitute the right-hand side of equation 4.52 into equation 4.51, we have $\gamma\sum_k B^kH = \pi + v' - \gamma A$, from which the substitution of $\gamma H - v$ for π gives us

$$\gamma\sum_k B^kH = \gamma H + v' - \gamma A - v. \tag{4.53}$$

Multiplying by F, we have

$$\gamma \sum_k B^k HF = \gamma HF. \tag{4.54}$$

Thus, there is some row vector r such that $rHF = 0$ and $\gamma \sum_k B^k = \gamma + r$.■

We would like lemma 4.3.7 to have established that $\gamma \sum_k B^k = \gamma$, because that would have completed the verification that $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$ is a Proto-GHMM. However, this is not always the case — a quasi-presentation is not always a Proto-GHMM. If HF is invertible — that is, if W and S are minimal — then $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$ is the standard presentation. In this case, we do have a Proto-GHMM, which we may prove by multiplying equation 4.54 by $(HF)^{-1}$. We will soon show that the standard presentation is, in fact, a GHMM equivalent to $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$, and we will use this fact later in this section when we address the minimization problem.

In the next lemma, which is the key step in the proof of theorem 4.3.3, we establish that the quasi-initial vector γ and the quasi-presentation matrices B^k can reproduce the probabilities of words given by the initial vector π and the transition matrices T^k. Here, we begin using the quasi-transition matrices as transition matrices, with the convention that if $x = x_1 \ldots x_n$ is a word, then $B^x = B^{x_1} \cdots B^{x_n}$.

Lemma 4.3.8. For all $x \in \mathcal{X}^*$,

1. $HT^x F = B^x HF$,
2. $\pi T^x F = \gamma B^x HF$, and
3. $\pi T^x \vec{1} = \gamma B^x \vec{1}$.

Proof. We will prove by induction on the length of x. If the length is one, then 1 is equivalent to $B^k HF = HT^k F$, which B^k satisfies by definition.

If x has length greater than one, then we can write $x = yk$ for $k \in \mathcal{X}$ and y the prefix of x with length $|x| - 1$. We will assume, as our induction assumption, that $HT^y F = B^y HF$. This means that there is some matrix A such that

$$HT^y = B^y H + A, \tag{4.55}$$

and all rows of A lie in $K_{\mathcal{F}}$. Now, multiply on the right by T^k and we have

$$HT^y T^k = B^y HT^k + AT^k. \tag{4.56}$$

Because B^k is defined to satisfy $B^k HF = HT^k F$, we know that $HT^k = B^k H + C$ for some matrix C, all rows of which lie in $K_{\mathcal{F}}$. Making this substitution for HT^k on the right-hand side of 4.56 and writing $C' = C + AT^k$, we have

$$HT^y T^k = B^y B^k H + C'. \tag{4.57}$$

Note that $C'F$ is a matrix of zeros. Thus, if we multiply equation 4.57 by F, we have

$$HT^x F = B^x HF, \tag{4.58}$$

which proves 1.

Note that we have shown that

$$HT^x = B^x H + A \tag{4.59}$$

for some matrix A such that AF is a matrix of zeros. Multiplying by γ and then substituting $\pi + v$ for γH gives us

$$\pi T^x + vT^x = \gamma B^x H + \gamma A, \tag{4.60}$$

for some $v \in K_{\mathcal{F}}$. Note that both vT^x and γA lie in $K_{\mathcal{F}}$. Now, if we multiply by F, we get

$$\pi T^x F = \gamma B^x HF, \tag{4.61}$$

thus proving 2.

And finally, if we multiply equation 4.60 by $\vec{1}$, we have

$$\pi T^x \vec{1} = \gamma B^x H\vec{1}. \tag{4.62}$$

Because H is a unit-sum matrix, equation 4.62 proves 3.■

We have now defined quasi-presentations, proven that they exist, and proven that they determine the probabilities of all words. Having done so, we are ready to prove theorem 4.3.3.

Proof of Theorem 4.3.3. If two GHMMs $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$ are equivalent, then they agree on the probabilities of all words, and thus an all conditional

probabilities. The interesting part of the theorem is that if W and S are sufficient for both $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$, and if these two GHMMs agree on the probabilities $\mathbf{P}(s|w)$, $\mathbf{P}(ks|w)$, and $\mathbf{P}(s)$ for all $w \in W$, $s \in S$, and $k \in \mathcal{X}$, then $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$ are equivalent. We will prove this by constructing a quasi-presentation for $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and showing that it is also a quasi-presentation for $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$.

Let H_1 and F_1 be the history and future matrices for $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and let H_2 and F_2 be the history and future matrices for $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$. Let us recall the hypotheses of this theorem.

1. For all $w_i \in W$ and $s_j \in S$, $\mathbf{P}(s_j|w_i) = \mathbf{Q}(s_j|w_i)$,
2. For all $w_i \in W$, $s_j \in S$, and $k \in \mathcal{X}$, $\mathbf{P}(ks_j|w_i) = \mathbf{Q}(ks_j|w_i)$, and
3. For all $s_j \in S$, $\mathbf{P}(s_j) = \mathbf{Q}(s_j)$.

First, $\mathbf{P}(s_j|w_i) = (H_1F_1)_{ij}$, and $\mathbf{Q}(s_j|w_i) = (H_2F_2)_{ij}$, so 1 is equivalent to $H_1F_1 = H_2F_2$. Second, $\mathbf{P}(ks_j|w_i) = \left(H_1T^kF_1\right)_{ij}$, and $\mathbf{Q}(ks_j|w_i) = \left(H_2U^kF_2\right)_{ij}$, so 2 can be written as follows: for all k, $H_1T^kF_1 = H_2U^kF_2$. Last, $\mathbf{P}(s_j) = (\pi F_1)_j$, and $\mathbf{Q}(s_j) = (\tau F_2)_j$, so 3 becomes $\pi F_1 = \tau F_2$. Thus, what we need to show is that these three facts — $H_1F_1 = H_2F_2$, $H_1T^kF_1 = H_2U^kF_2$ for all k, and $\pi F_1 = \tau F_2$ — together imply that $\mathbf{P}(x) = \mathbf{Q}(x)$ for any $x \in \mathcal{X}^*$, where $\mathbf{P}(x) = \pi T^x\vec{1}$ and $\mathbf{Q}(x) = \tau U^k\vec{1}$. Let γ be any solution to $\gamma H_1F_1 = \pi F_1$, and for all k, let B^k be any solution to $B^kH_1F_1 = H_1T^kF_1$. Then $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$ is a quasi-presentation for the process represented by $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$. Lemma 4.3.8 now tells us that for any $x \in \mathcal{X}^*$ $\pi T^x\vec{1} = \gamma B^x\vec{1}$.

But we know that $H_1F_1 = H_2F_2$ and $\pi F_1 = \tau F_2$, so γ satisfies $\gamma H_2F_2 = \tau F_2$. Similarly, each B^k satisfies $B^kH_2F_2 = H_2U^kF_2$. Thus, $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$ is also a quasi-presentation for $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$. So lemma 4.3.8 tells us that for all x, $\tau U^x\vec{1} = \gamma B^x\vec{1}$, and therefore that for all x, $\tau U^x\vec{1} = \pi T^x\vec{1}$.■

Corollary 4.3.9. If $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$ are two GHMMs which assign the same probabilities to all words of length less than $2n$, neither of which has more than n states, then they are equivalent.

Proof. Lemma 4.2.14 tells us that we can find wordlists W and S in which all of the words have length at most $n-1$. For such wordlists, all words of the form w_i, s_j, $w_i s_j$, and $w_i k s_j$ have lengths less than $2n$, so $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$ must agree on all $\mathbf{P}(s_j|w_i) = \mathbf{P}(w_i s_j)/\mathbf{P}(w_i)$, $\mathbf{P}(k s_j|w_i) = \mathbf{P}(w_i k s_j)/\mathbf{P}(w_i)$, and $\mathbf{P}(s_j)$. Thus, by theorem 4.3.3, they are equivalent.■

With this machinery in hand, we can resolve the minimization problem.

Theorem 4.3.10. Given a GHMM $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$, let $\mathbf{P}$ be the process it represents, and let W and S be minimal wordlists. Then the standard presentation $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$ for given W and S is a GHMM, and it is equivalent to $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$. Furthermore, no GHMM exists with fewer than $|W|$ states which is equivalent to $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$.

Proof. As noted on page 78, when HF is invertible, $\gamma = \sum_k B^k \gamma$, and the standard presentation is a Proto-GHMM. We established in lemma 4.3.8 that $\gamma B^x \vec{1} = \pi T^x \vec{1}$ for all $x \in \mathcal{X}^*$. This proves both that it is a GHMM and that it is equivalent to $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$.

Because the rows of HF are linearly independent, and these rows consist of conditional future probabilities, we know that the process states $\mathbf{P}(\cdot|w_i)$ are linearly independent. Thus the span of the reachable process states has dimension at least $|W|$. In fact, if we combine lemma 4.2.1 and proposition 4.2.6, we have proven that its dimension is exactly $|W|$. If a GHMM has fewer than $|W|$ states, then it induces a process for which the span of the reachable process states has dimension less than $|W|$, and thus it cannot be equivalent to $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$.■

We conclude this section with a result — the existence of conjugacies — which we promised in section 4.1. This was first shown — for functions of finite Markov Chains — by Gilbert [20].

Proposition 4.3.11. Let $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$ be two minimal, equivalent GHMMs — that is, $|V| = |V'| = \dim(\mathcal{U})$, where $\mathcal{U}$ is the span of the reachable process states. Then there exists a matrix M such that $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$ are conjugate by M.

Proof. Let W and S be minimal wordlists for $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$, and let H_1 and F_1 be the associated history and future matrices. Because W and S are minimal, $H_1 F_1$

must be invertible. We know that $|V| = \dim(\mathcal{U})$, and for minimal W we know that $|W| = \dim(\mathcal{U})$. But H_1 has $|W|$ rows and $|V|$ columns, so it must be square. Likewise, F_1 must be square and both H_1 and F_1 must be invertible.

Let H_2 and F_2 be the history and future matrices for $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$. Then the process states $\mathbf{P}(\cdot|w_i)$ form a linearly independent basis for $\mathcal{U}$, hence the rows of H_2 must be linearly independent. Thus H_1, F_1, and H_2 are full rank square matrices, and must be invertible. We also know that $H_1F_1 = H_2F_2$ is invertible, so $F_2 = H_2^{-1}H_1F_1$ must be invertible.

Now, we calculate. We have $H_2U^kF_2 = H_1T^kF_1$ and $\pi F_1 = \tau F_2$, so

$$U^k = H_2^{-1}H_1T^kF_1F_2^{-1} \tag{4.63}$$

and $\tau = \pi F_1F_2^{-1}$. And if we let $M = H_2^{-1}H_1$, M is a unit-sum matrix. And $H_1F_1 = H_2F_2$ implies $H_2^{-1}H_1F_1F_2^{-1} = I$, so $M^{-1} = F_1F_2^{-1}$. So equation 4.63 may be written $U^k = MT^kM^{-1}$. Moreover, we have $\tau = \pi F_1F_2^{-1} = \pi M^{-1}$. Thus $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ and $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$ are conjugate.■

We began this section by defining a GHMM to be a representation of a process similar to an HMM, but with negative entries allowed in its transition matrices. We studied the vector spaces of conditional distributions on the future induced by history words and of conditional distributions on the past induced by future words. We found bases for these vector spaces in terms of these same words. These bases were instrumental in resolving the identifiability and minimization problems, and they led us to the standard presentation. We have shown that the standard presentation is a GHMM presentation for a process, and we have observed that it is determined entirely by probabilities of words. In the next chapter, we will use this fact to construct presentations directly from probabilities of words — that is, directly from the process.

5 Reconstruction

The subject of this chapter is the *reconstruction problem*, which has two versions. In both, the objective is to construct a presentation for some process given certain information about that process. In the first, which we will call *reconstruction from probabilities*, the given information is the probability the process assigns to every word in $\mathcal{X}^*$. In the second, which we will call *reconstruction from a sample*, the given information is a sample of the process's output. We will use this sample data solely to estimate probabilities of words, so reconstruction from a sample may be viewed as a form of reconstruction from probabilities in which the probabilities are only approximately known. Alternatively, reconstruction from probabilities may be thought of as an idealized form of reconstruction from a sample, in which the sample is infinitely large. In both versions, we make the assumption that the span of the process states is finite-dimensioned. In fact, we will show how to construct a GHMM presentation for any process which satisfies this condition.

A substantial body of research has accumulated around the problem of reconstruction from a sample for HMMs, for example [23–26]. Most of it involves versions of an algorithm known as forward-backward or Baum-Welsh [17,24]. These are forms of the expectation-maximization (EM) algorithm [27]. With all of these algorithms, a number of structural assumptions are required — the number of states and some choice about what transitions will be allowed to occur (for example, all may be allowed). Then a random initial presentation which satisfies the structural assumptions is chosen. The various algorithms then implicitly assume it describes the sample to some degree, and iteratively adjust the parameters while leaving the structure fixed.

In contrast, we will present a solution to the problem of reconstruction from probabilities, which appears not to have been studied before. We will follow this with an adaptation of this solution to reconstruction from a sample. The resulting reconstruction algorithm does not require its user to choose a number of states or a structure of allowed transitions, nor does it depend on any ability of random HMMs to describe the sample. Instead, it operates by estimating conditional distributions (that is, process states) from the sample and then constructs GHMM transition matrices directly from these conditional distributions.

This algorithm is new, and it is the work of the author. It should be noted, however, that other elements of this algorithm have been used before. Gilbert [20] and Dharmadhikari [10,28], consider the rank of the matrix of probabilities we call HT^kF. Given a function of a Markov chain in a certain class, Gilbert constructs a new function of a Markov chain which is conjugate to the original. His technique, like the one used here, derives a set of presentation states form a matrix of word probabilities. Crutchfield [11] uses estimated probabilities to reconstruct presentations of stochastic deterministic finite automata from samples. And the QL algorithm is a standard technique in numerical linear algebra [29].

5.1 Constructing a Presentation for a Process

The key observation in section 4.3 was this: the matrices HF and HT^kF, for all k, do not depend on a presentation. We can rewrite theorem 4.3.10 as follows: if a process $\mathcal{P} = \left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$ has a presentation for which W and S are minimal wordlists, then $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$ is a presentation for $\mathcal{P}$. That is, we can construct a presentation for $\mathcal{P}$ almost without referring to a preexisting presentation — if we know the probabilities of the right words, we can compute all the components of $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$ from those probabilities using 4.3.10. But so far, we still need a preexisting presentation from which to construct the wordlists. To solve the problem of reconstruction from probabilities, then, we need to be able to build minimal wordlists from probabilities — that is, from a process without referring to a presentation.

It turns out that it is possible to construct suitable wordlists from probabilities of words, but no finite algorithm can do so correctly in every case. The reason is simple: a finite algorithm can only examine the probability of a finite number of words. If we build a presentation from the probabilities of a finite number of words, then it is always possible that there is a word, which was not examined, whose probability is not correctly extrapolated from the words which were examined by the presentation. Thus, the solution to the problem of reconstruction from probabilities must have an element which is either infinitary or nonconstructive. Fortunately, there is an algorithm, built on our theoretical framework, that works well in practice as we will see in the next section. And we can find suitable wordlists directly given any upper bound on the maximum word length, such as that which can be derived from the number of presentation states.

In chapter 4, we built our history wordlists so that the set of mixed states induced by their words spanned $\mathcal{H}/K_{\mathcal{F}}$. Recall that $\mathcal{U}$ is defined as the span of the recurrent process states, without reference to a presentation, and that $\mathcal{U}$ is isomorphic to $\mathcal{H}/K_{\mathcal{F}}$. To build history wordlists without referring to presentations, we will choose words that induce a set of process states spanning $\mathcal{U}$. Describing the future wordlists in an analogous fashion is a little more difficult because we do not have a concept analogous to "state" that refers to something induced by the future.* If we recall the definition of a reachable process state — a conditional distribution on the future which is induced by a history word — then the appropriate analog is clear: a conditional distribution on the past which is induced by a future word. Such a conditional distribution would be a process state if we were to reverse the direction of time, so we will call it a *reverse state*. The future wordlists we wish to build, then, consist of words which induce a set of reverse states which has the same span as the set of all reverse states. It will not be necessary to actually write reverse states in any calculations, so we will not define notation for them.

At this point, we shift to more nearly concrete objects. We need to choose an order on $\mathcal{X}^*$ — the order we choose does not matter, so long as it is fixed. A natural choice is to put shorter words before longer words, and put words of the same length in lexicographical order by some order on $\mathcal{X}$. When $\mathcal{X} = \{0, 1\}$, and 0 precedes 1, we have

$$\lambda, 0, 1, 00, 01, 10, 11, 000, \ldots \tag{5.1}$$

Let q_i be the ith word in this ordering, then we have a one-to-one correspondence between $\mathbb{N}$ and $\mathcal{X}^*$.

We can now define the infinite matrix P, whose entries are indexed by $\mathbb{N} \times \mathbb{N}$. Let $w = q_i$, $s = q_j$, and

$$P_{ij} = \begin{cases} \mathbf{P}(s|w) & \mathbf{P}(w) \neq 0 \\ 0 & \mathbf{P}(w) = 0. \end{cases} \tag{5.2}$$

For convenience and clarity, we will use q_i in place of i itself in subscripts of P and write $P_{w,s}$ instead of P_{ij}. Each row of P is labeled with a history word, each column with a future word, and each entry (except for those in all-zero rows) is a conditional probability. The top row, with history word λ, contains the unconditioned probabilities $\mathbf{P}(w|\lambda) = \mathbf{P}(w)$. So in principle the top row alone can generate the whole matrix. Each

*There is a time-symmetric development of our approach that we have chosen not to present here.

row contains a complete description of a reachable process state, in the form of all the conditional probabilities on future words. Each column contains a complete description of a reverse state, although some renormalization is needed to put the column into the right form. Thus, any word w identifies both a unique row $P_{w,\cdot}$ and a unique column $P_{\cdot,w}$. We will denote the row associated with w by $r(w)$ and the column associated with s by $c(s)$.

The following definition is analogous to definition 4.2.4 of sufficient wordlists for a GHMM.

Definition 5.1.1. A history wordlist W is *sufficient for a process* if the rows identified by the words in W span all the rows of P. Similarly, a future wordlist S is sufficient for a process if the columns identified by the words in S span all the columns of P.

Now, a pair of wordlists W and S identifies a submatrix of P in a natural way: if $W = \{w_1, \ldots, w_n\}$ and $S = \{s_1, \ldots, s_k\}$, we define the submatrix G of P by $G_{ij} = P_{w_i,s_j}$, so that $G_{ij} = \mathbf{P}(s_j|w_i)$ for all $i = 1, \ldots, n$ and $j = 1, \ldots, l$. That is, we pick out the elements of P which are both in the rows identified by W and in the columns identified by S. If we had a presentation for $\mathcal{P}$ and we built the wordlists W and S and the matrices H and F for that presentation, we would find that $G = HF$ (see equation 4.40). In the same manner, we define a submatrix C^k of P for each $k \in \mathcal{X}$ by picking out the rows of P corresponding to words in W and the columns corresponding to words ks_j for $s_j \in S$. That is, C^k is defined by $C^k_{ij} = P_{w_i,ks_j} = \mathbf{P}(ks_j|w_i)$. Thus, if we had a presentation for $\mathcal{P}$, we would have $C^k = HT^kF$.

In corollary 4.2.9, we established that if W and S are minimal for a GHMM, then HF is invertible. And we know that if either W or S is sufficient but not minimal, then HF is larger in size — but not in rank — than it would be if W and S were minimal. Thus if W and S are sufficient, HF is invertible if and only if W and S are both minimal. We will use the analogous version of this for the following definition.

Definition 5.1.2. A pair of wordlists W and S are *minimal for a process* $\mathcal{P}$ if they are sufficient for $\mathcal{P}$ and the matrix G they define is invertible.

The next few results establish that sufficiency and minimality for a process have the properties we will need in theorem 5.1.8, which is the main result of this section. These

properties are roughly analogous to the properties of sufficient and minimal wordlists for a GHMM, though we will not attempt to draw precise analogies.

We are interested in the number of linearly independent rows and columns of P. If P were finite, we would refer to its rank. Thus we give the following definition for rank in this infinite context.

Definition 5.1.3. The *rank* of P is defined to be the supremum of the ranks of the finite submatrices of P.

Lemma 5.1.4. The rank of P is equal to the dimension of the span of $\mathcal{U}$.

Because of the close connection between the rows of P and the reachable process states, this is almost automatic. The only difficulties are in dealing with the infinitely many columns in those rows.

Proof. Let $n = \dim(\mathcal{U})$, which we will assume for the moment is finite. We can choose $\{w_1, \ldots, w_n\}$ such that if $\mathbf{A}_i = \mathbf{P}(\cdot|w_i)$ for $i = 1, \ldots, n$, the process states $\mathbf{A}_1, \ldots, \mathbf{A}_n$ span $\mathcal{U}$. Then for all $w \in \mathcal{X}^*$, there are numbers $a_1, \ldots, a_n$ such that

$$\mathbf{P}(\cdot|w) = a_1\mathbf{A}_1 + \ldots + a_n\mathbf{A}_n. \tag{5.3}$$

so for all s, $\mathbf{P}(s|w) = \sum_i a_i\mathbf{A}_i(s)$. This implies that the row $r(w)$ can be written as $r(w) = a_1 r(w_1) + \ldots + a_n r(w_n)$. So the rows $r(w_1), \ldots, r(w_n)$ form a basis for the rows of P. Thus, any collection of more than n rows is linearly dependent, and the same must be true for the rows of submatrices. This shows that rank$(P) \leq n$.

Conversely, because $\mathbf{A}_1, \ldots, \mathbf{A}_n$ are linearly independent and the row $r(w_i)$ determines the distribution $\mathbf{A}_i = \mathbf{P}(\cdot|w_i)$, $r(w_1), \ldots, r(w_n)$ must be linearly independent. So the row space of P has dimension n. What remains to be shown is that P has a finite submatrix of rank n.

For any word s, let $c'(s)$ be the length n column vector consisting of those elements of $c(s)$ which lie in rows $r(w_1), \ldots, r(w_n)$; that is,

$$c'(s) = \begin{pmatrix} \mathbf{P}(s|w_1) \\ \vdots \\ \mathbf{P}(s|w_n) \end{pmatrix} = \begin{pmatrix} P_{w_1,s} \\ \vdots \\ P_{w_n,s} \end{pmatrix}. \tag{5.4}$$

We will call $c'(s)$ a *subcolumn* of P. Choose a wordlist S such that $c'(s_1), \ldots, c'(s_l)$ is a linearly independent basis for the span of all subcolumns $c'(s)$. Let G be the $n \times l$

submatrix of P given by W and S. The columns of G are exactly the subcolumns $c'(s_1), \ldots, c'(s_l)$, so G has rank l. We have shown the rank of G must be less than n, so $l \leq n$.

We now define the subrow r' in a manner analogous to the definition of a subcolumn. The subrow $r'(s)$ is the length l row vector $(P_{s_1,w}, \ldots, P_{s_l,w})$. We will reserve the terms *subrow* and *subcolumn* for these subsets of P, and we will use the terms *row vector* and *column vector* for other row and column vectors, including linear combinations of subrows and subcolumns.

Suppose that $l < n$. Then there exists a row vector $(a_1, \ldots, a_n)$, not all components of which are zero, such that the product $(a_1, \ldots, a_n)G$ is a linear combination of subrows which is the zero row vector. Now, for any $s \in \mathcal{X}^*$, the subcolumn $c'(s)$ is a linear combination of $c'(s_1), \ldots, c'(s_l)$, so if the subrow $r'(w_i)$ is zero, the ith element of each $c'(s_j)$ must be zero, and so the ith element of every subcolumn must be zero. That is, if a subrow is zero, the corresponding entire row must be zero. The same must be true for linear combinations of subrows: for any s the element in column s of the infinite vector $a_1 r(w_1) + \ldots + a_n r(w_n)$ is a linear combination of elements of the row vector $(a_1, \ldots, a_n)G$. Thus, if $(a_1, \ldots, a_n)G$ is all zeros, then every element of $a_1 r(w_1) + \ldots + a_n r(w_n)$ is a linear combination of zeros, and so

$$a_1 P_{w_1,s} + \ldots + a_n P_{w_n,s} = 0, \tag{5.5}$$

for all words s.

But the rows $r(w_1), \ldots, r(w_n)$ have been shown to be linearly independent, so $a_1 r(w_1) + \ldots + a_n r(w_n)$ cannot be equal to a row of zeros. Thus, there is a word s such that

$$a_1 P_{w_1,s} + \ldots + a_n P_{w_n,s} \neq 0. \tag{5.6}$$

This is a contradiction. Therefore, $n = l$ and G is a square submatrix of P which has rank n.†

†We may interpret this in the following way: the process states of the forward and time-reversed processes have span sets of the same dimension. Thus, the minimal GHMM presentations for the forward and time-reversed processes have the same number of presentation states.

Lastly, if $n = \dim(\mathcal{U})$ is infinite, then for any m we can find linearly independent process states $\mathbf{A}_1, \ldots, \mathbf{A}_m$. Proceeding as above, we can construct a submatrix G or P with rank m. Thus there is no upper bound to the ranks of the submatrices of P, so the rank of P is infinite.■

The first part of the following result has essentially already been proven. The second part establishes that minimal wordlists exist.

Proposition 5.1.5. For any process $\mathcal{P}$, let n be the rank of P. If n is finite, then

1. There exists an invertible $n \times n$ submatrix of P, and
2. If G is any such matrix, then the wordlists W and S that produce it are minimal for $\mathcal{P}$.

Proof. In the proof of lemma 5.1.4, we constructed an $n \times l$ matrix G which had rank l, and showed than $l = n$. Thus G is a square matrix of full rank, and must be invertible, which proves part 1. Note that the invertibility of G is also part of what it means to be minimal for a process. Thus, to prove part 2, we need only show that W and S are sufficient, because G is invertible. That is, we need only show that $r(w_1), \ldots, r(w_n)$ span all rows of P, and that $c(s_1), \ldots c(s_n)$ span all columns of P. As in the proof of lemma 5.1.4, we will use $c'(s)$ to denote the subcolumn of P,

$$c'(s) = \begin{pmatrix} P_{w_1,s} \\ \vdots \\ P_{w_n,s} \end{pmatrix}. \tag{5.7}$$

Thus, $c'(s_j)$ is the jth column of G. Likewise, we will use $r'(w)$ to denote the subrow $r'(w) = (P_{ws_1}, \ldots, P_{ws_n})$, so that $r'(w_i)$ is the ith row of G.

Let w be any word not in W. The set of subrows $\{r'(w),\ r'(w_1), \ldots, r'(w_n)\}$ contains $n+1$ vectors of length n, hence they are not linearly independent. But $r'(w_1), \ldots, r'(w_n)$ are linearly independent, so $r'(w)$ must be a linear combination of them. That is, there is a vector a such that $aG = r'(w)$. Because G is invertible, we have $a = r'(w)G^{-1}$; that is, there is exactly one such a.

Now let s be any word not in S, and consider the $(n+1) \times (n+1)$ submatrix M of P given by

$$M = \begin{pmatrix} G & c'(s) \\ r'(w) & P_{ws} \end{pmatrix}. \tag{5.8}$$

We know that rank$(P) = n$, so M must be singular. The first n rows of M are linearly independent, so the last row must be a linear combination of them. That is, there must be a vector b such that $bG = r'(w)$ and $bc'(s) = P_{w,s}$. Now we have just shown that there is a unique vector a such that $aG = r'(w)$, and this a clearly does not depend on s. So we must have $a = b$, and $ac'(s) = P_{w,s}$. Furthermore, this must hold for all s. Thus the entire row $r(w)$ of P satisfies

$$r(w) = a_1 r(w_1) + \ldots + a_n r(w_n), \tag{5.9}$$

so we have shown that all rows of P lie in the span of $r(w_1), \ldots, r(w_n)$.

A similar argument shows that the columns $c(s_1), \ldots, c(s_n)$ span all columns of P.■

The next lemma completes the development of minimal wordlists for a process.

Lemma 5.1.6. Let $\mathcal{P}$ be any process such that the rank n of P is finite. If W and S are minimal wordlists for $\mathcal{P}$, then both W and S have length n.

Proof. If W and S are minimal wordlists for $\mathcal{P}$, then G is an invertible submatrix of P. The rank of G is at most n, and so W and S (which must have the same length because invertible matrices must be square) have length at most n. Furthermore, there are sets of n rows which are linearly independent, all of which must lie in the span of the rows identified by W. So W must have length n.■

We need one more lemma before we are ready for theorem 5.1.8. This one is a calculation, most of which appeared in the proof of proposition 5.1.5.

Lemma 5.1.7. Given a process $\mathcal{P}$ and a matrix G defined by minimal wordlists, for any $w, s \in \mathcal{X}^*$, we have

$$r'(w)G^{-1}c'(s) = \mathbf{P}(s|w). \tag{5.10}$$

Proof. As in the proof of proposition 5.1.5, let

$$M = \begin{pmatrix} G & c'(s) \\ r'(w) & P_{w,s} \end{pmatrix}. \tag{5.11}$$

M must be singular in such a way that there is a vector a satisfying

1. $aG = r'(w)$, and
2. $ac'(s) = P_{w,s}$.

And because G is invertible, we must have $a = r'(w)G^{-1}$. Thus we have

$$r'(w)G^{-1}c'(s) = P_{w,s} = \mathbf{P}(s|w). \blacksquare \tag{5.12}$$

Note that if we let $w = x$ and $s = \lambda$, lemma 5.1.7 gives us $r'(x)G^{-1}c'(\lambda) = \mathbf{P}(\lambda|x)$. But $c'(\lambda) = \vec{1}$ and $\mathbf{P}(\lambda|x) = 1$ for any x, so for all x we have

$$r'(x)G^{-1}\vec{1} = 1. \tag{5.13}$$

We have now completed the minor results concerning minimal wordlists for processes, and are ready to state and prove the main result of this section. We will use δ_i to represent a vector with a one in the ith position and zeros in all other positions. This symbol represents a row vector when it appears to the left of a matrix and a column vector when it appears to the right of a matrix.

Recall that we have defined C^k by $C^k_{ij} = P_{w_i,ks_j}$, and that if we have arbitrary wordlists and any presentation for $\mathcal{P}$, we have $G = HF$ and $C^k = HT^kF$. Note that C^k satisfies the following:

$$\begin{aligned} C^k_{ij} &= \mathbf{P}(ks_j|w_i) \\ &= \mathbf{P}(k|w_i)\mathbf{P}(s_j|w_ik) \\ &= \mathbf{P}(k|w_i)P_{w_ik,s_j}. \end{aligned} \tag{5.14}$$

so the ith row of C^k satisfies

$$\delta_i C^k = \mathbf{P}(k|w)r'(w_ik) \tag{5.15}$$

Theorem 5.1.8. If $\mathcal{P}$ is any process for which the dimension of the span of $\mathcal{U}$ is finite, and W and S are minimal wordlists for $\mathcal{P}$, then $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$ is a GHMM presentation for $\mathcal{P}$, where $B^k = C^kG^{-1}$ and $\gamma = (\mathbf{P}(s_1), \ldots \mathbf{P}(s_n))G^{-1}$.

Proof. To prove that $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$ is a presentation for $\mathcal{P}$, we must show $\sum_k B^k$ is unit–sum and that $\gamma B^x\vec{1} = \mathbf{P}(x)$ for all $x \in \mathcal{X}^*$. The first of these is a calculation. By manipulating the definition of the B^ks, we get

$$\left(\sum_k B^k\right)\vec{1} = \left(\sum_k C^k\right)G^{-1}\vec{1}. \tag{5.16}$$

We will work with the ith row alone, and use equation 5.15:

$$\delta_i \left(\sum_k B^k \right) \vec{1} = \delta_i \left(\sum_k C^k \right) G^{-1} \vec{1} = \sum_k \mathbf{P}(k|w_i) r'(w_i k) G^{-1} \vec{1}. \tag{5.17}$$

Applying equation 5.13, we get

$$\begin{aligned} \delta_i \left(\sum_k B^k \right) \vec{1} &= \sum_k \mathbf{P}(k|w_i) \\ &= 1. \end{aligned} \tag{5.18}$$

Since this is true for each i, we have shown that $\sum_k B^k$ is unit-sum: $\left(\sum_k B^k \right) \vec{1} = \vec{1}$.

Showing that

$$\gamma B^x \vec{1} = \mathbf{P}(x) \tag{5.19}$$

holds for all x is more involved. We will prove this by proving that for all x,

$$\gamma B^x = \mathbf{P}(x) r'(x) G^{-1}. \tag{5.20}$$

From this multiplying on the right by $\vec{1}$ and applying equation 5.13 gives us equation 5.19.

We will establish equation 5.20 by inductively concatenating symbols to form an arbitrary word x. The base case is trivial — B^γ is the identity matrix, $\mathbf{P}(\lambda) = 1$, and $\gamma = r' G^{-1}$ by definition. So what remains to be shown is the induction case: given a word x and a symbol k, assume that $\gamma B^x = \mathbf{P}(x) r'(x) G^{-1}$ and prove that $\gamma B^x B^k = \mathbf{P}(xk) r'(xk) G^{-1}$.

Consider the jth coordinate of $\mathbf{P}(k|x) r'(xk)$: we have

$$\begin{aligned} \mathbf{P}(k|x) r'(xk) \delta_j &= \mathbf{P}(k|x) P_{xk,s_j} \\ &= \mathbf{P}(k|x) \mathbf{P}(s_j|xk) \\ &= \mathbf{P}(k s_j|x). \end{aligned} \tag{5.21}$$

Replacing the right hand side using lemma 5.17, we have

$$\mathbf{P}(k|x) r'(xk) \delta_j = r'(x) G^{-1} c'(k s_j). \tag{5.22}$$

Now, note that $C^k_{ij} = P_{w_i, k s_j}$ is the ith row of $c'(k s_j)$, so that for all i,

$$\delta_i c'(k s_j) = C^k_{ij} = \delta_i C^k \delta_j. \tag{5.23}$$

Thus, we have $C^k\delta_j = c'(ks_j)$, and equation 5.22 becomes

$$\mathbf{P}(k|x)r'(xk)\delta_j = r'(x)G^{-1}C^k\delta_j. \tag{5.24}$$

This holds for all j, so we have

$$\mathbf{P}(k|x)r'(xk) = r'(x)G^{-1}C^k. \tag{5.25}$$

Next we multiply both sides by $\mathbf{P}(x)$ on the left and G^{-1} on the right and then simplify:

$$\begin{aligned} \mathbf{P}(x)\mathbf{P}(k|x)r'(xk)G^{-1} &= \mathbf{P}(x)r'(x)G^{-1}C^kG^{-1} \\ \mathbf{P}(xk)r'(xk)G^{-1} &= \mathbf{P}(x)r'(x)G^{-1}B^k. \end{aligned} \tag{5.26}$$

Finally, we use the induction hypothesis $\gamma B^x = \mathbf{P}(x)r'(x)G^{-1}$ and we get

$$\mathbf{P}(xk)r'(xk)G^{-1} = \gamma B^xB^k, \tag{5.27}$$

and the proof is complete.■

We summarize the constructive portion of the preceding development as follows.

Algorithm 5.1.9. Let $\mathcal{P} = \left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$ be a process satisfying the condition that the span of its process states is finite-dimensional. A GHMM presentation for $\mathcal{P}$ may be constructed by the following steps.

1. Construct the infinite matrix P with entries $P_{w,s} = \mathbf{P}(s|w)$.
2. Find a nonsingular minor G of P such that no other minor of P has rank greater than the rank of G.
3. Build the history wordlist $W = \{w_1, \ldots, w_n\}$ by defining w_i to be the word associated with the ith row of P. Build the future wordlist $s = \{s_1, \ldots, s_n\}$ by defining s_j to be the word associated with the jth column of P.
4. For each $k \in \mathcal{X}$, construct the matrix C^k with entries $C^k_{ij} = \mathbf{P}(ks_j|w_i)$.
5. For each $k \in \mathcal{X}$, compute $B^k = C^kG^{-1}$.
6. Compute $\gamma = (\mathbf{P}(s_1), \ldots, \mathbf{P}(s_n))G^{-1}$.

As we will see in section 5.2, a reconstruction program based on this theoretical framework has been developed. When it is given a set of word probabilities produced by a GHMM it reliably constructs a GHMM which accurately reproduces those word probabilities.

We conclude this section with the converse of theorem 4.16, which says that every process that has a GHMM presentation satisfies $\dim(\mathcal{U}) < \infty$. And if conjecture 6.1.1 is true, it is also a converse of theorem 3.6.1, which says the same thing for HMMs.

Corollary 5.1.10. Every process such that the dimension of the span of its reachable process states is finite has a GHMM presentation.

Proof. Let $\mathcal{P}$ be any process for which the span of $\mathcal{U}$ is finite dimensional. Lemma 5.1.4 tells us that P has finite rank, and proposition 5.1.5 then establishes that finite minimal wordlists exist. Theorem 5.1.8 gives us a GHMM presentation in terms of these wordlists.■

Together, theorem 4.16 and corollary 5.1.10 prove the following characterization of the class of processes represented by GHMMs.

Theorem 5.1.11. A process has a GHMM presentation if and only if the dimension of the span of its reachable process states is finite.

5.2 Reconstruction from a Sample

Suppose we are given a finite sequence of symbols and we are told that it is a sample of output from a process $\mathcal{P}$. How can we construct a presentation for this process? This is the problem of *reconstruction from a sample*. We will consider samples which consist of a single sample sequence of length L, such as

$$01101\ldots10, \tag{5.28}$$

and also samples which consist of several sample sequences of total length L, like

$$11110\ldots11,\ 0111\ldots0,\ \text{and}\ 010110\ldots001. \tag{5.29}$$

It should be apparent to the reader that this problem is of a different character than the problem of reconstruction from probabilities. Any given finite sample could have been generated by any one of an infinite number of processes, so the problem cannot be solved in any absolute sense. The best we can possibly do is to give a presentation for a process which would be likely to generate this sample, and consider this presentation to represent a new process that is an approximation to $\mathcal{P}$. Thus, we

must heed statistical issues such as variances of parameter estimates, and how much data is available. Further, because this is a question which can be asked for real data in a practical setting, we will be interested in the computational issues of operation counts and storage needs. These issues are discussed in subsections 2 and 3, following the description of the reconstruction algorithm.

The Algorithm Our approach to reconstruction from a sample estimates conditional probabilities of various words from the relative frequencies of those words in the data. It then assembles these probabilities into a truncated (finite size) estimate $\hat{P}$ of P. From here we will proceed as in algorithm 5.1.9, substituting $\hat{P}$ for P and adapting the algorithm so that it works with the finite size and imperfect estimation of $\hat{P}$.

Our first task, then, is to construct $\hat{P}$, for which we need an estimator and a pair of wordlists. Let r and r' be the lengths of the longest history and future words which we will consider, respectively. Define a *cutpoint* to be a position between to consecutive symbols in a sample sequence which is at least r symbols from the beginning and r' symbols from the end of the sample sequence. That is, a cutpoint is a time at which we know the immediate history and future words of lengths at least r and r', respectively. For any pair of words w and s, let b be the number of cutpoints in all sample sequences preceded by w, and let a be the number of cutpoints preceded by w and followed by s. Our estimate for $\mathbf{P}(s|w)$, which we will denote $\pi(s|w)$, is given by $\pi(s|w) = a/b$. We use the conventions that $\pi(s|w) = 0$ if $b = 0$ and that if $w = \lambda$, b is the number of cutpoints in the sample. This is simply a frequency substitution estimate, and it has mean $\mathbf{P}(s|w)$. For any sample, and for any choice of r, the estimates $\pi(s|w)$ for different pairs of words w and s are consistent with each other. (For example, for any w, s, and x, $\pi(ws|x) = \pi(w|x) \cdot \pi(s|w)$.)

The wordlists we will use to construct $\hat{P}$ will not be minimal in any sense. We will make them as large as possible to make sure they are sufficient. We are limited by our data in what words we can include. (If we fix b, the variance of $\pi(s|w)$ can be shown to be $\mathbf{P}(s|w)(1 - \mathbf{P}(s|w))/b$, so the estimate is of little value if b is too small. If a is too small, on the other hand, the standard deviation becomes large relative to a/b even if b is large.) Thus, we will need to select a large set of words which occur reasonably often. The precise method by which we do this may be rather ad-hoc, as

all reasonable methods will produce similar sets of words. This is because they will all include short words with well estimated probabilities and will exclude long words with poorly estimated probabilities. The essential requirement is that a balance must be struck between making the wordlists large and making the variances of the estimates small. For simplicity, we will use the following heuristic: let l be the largest natural number such that K times the total number of words of length $2l+1$ that occur in the sample is less than the total number of cutpoints in the sample, for some fixed constant K. This means that those words of length $2l+1$ that have at least one occurrence in the sample occur an average of more than K times. We choose both of our wordlists to be the set of all words of length l or less that occur in the sample. Let Y denote this set.

Now, with our wordlists chosen, we construct the $|Y| \times |Y|$ matrix $\overline{P}$ from history wordlist Y and future wordlist Y just as we constructed G from the minimal wordlists W and S in section 5.1. That is, let $\overline{P}$ consist of one row and one column for each word in Y, so that for all $w, s \in Y$, $\overline{P}$ has an element $\overline{P}_{w,s} = P_{w,s} = \mathbf{P}(s|w)$. $\overline{P}$ contains those unknown true probabilities we want to estimate. We define $|Y| \times |Y|$ matrix $\hat{P}$ by replacing each conditional probability $\overline{P}_{w,s} = \mathbf{P}(s|w)$ with its estimate $\hat{P}_{w,s} = \pi(s|w)$ for all $w, s \in Y$. Thus, $\hat{P}$ is a stochastic matrix which may be thought of as an estimate of $\overline{P}$.

Hopefully $\hat{P}$ is large enough, by which we mean that Y contains sufficient history and future wordlists for the process $\mathcal{P}$. If it is not, then the presentation we produce will represent only a poor approximation to $\mathcal{P}$, and the available data is probably insufficient to induce a good approximation. In this case, the estimates in $\hat{P}$ have sufficiently small variances, but some essential behavior of the process cannot be deduced from the probabilities we have estimated. If we make Y larger, the probabilities we attempt to estimate capture the essential behavior, but our estimates might have variance large enough to make them meaningless. If we followed section 5.1 exactly, the next step would be to find a minor G of $\hat{P}$ which has the same rank as $\hat{P}$. Because we have chosen Y as large as possible, we expect it to contain minimal wordlists W and S as proper subsets, and we expect $\hat{P}$ to have a rank much less than its size.

However, this is probably not the case. If Y is big enough, $\overline{P}$ will have a rank much less than its size; but $\hat{P}$ will not. Because each $\hat{P}_{w,s}$ is a random variable with nonzero variance, $\hat{P}$ is a random matrix close to the singular matrix $\overline{P}$. Generically,

such a matrix will be nonsingular, but ill-conditioned. We want G to have the same rank as $\overline{P}$ and to be well-conditioned; because the rank of $\overline{P}$ is unknown, we will make G as large a well-conditioned submatrix of $\hat{P}$ as possible. For this reason, the problem of finding a suitable G is itself an estimation problem.

We solve it as follows. We decompose $\hat{P}$ by the QL algorithm with pivoting. The QL decomposition is related by transposes to the more familiar QR decomposition, and both are standard techniques in numerical linear algebra [29]. This algorithm builds a basis for the row space of $\hat{P}$, starting with an empty basis, and adding one vector at a time. At each step, it computes the distance from each row vector to the subspace spanned by the developing basis. It selects the row with the greatest distance, and adds a vector derived from it to the basis, in such a way that the basis now spans the selected row. As the reader may deduce, each row is selected at most once and the distances for the selected rows decrease as more rows are selected.

As a by-product, the QL algorithm lists the rows of the matrix in the order in which they were selected, and it gives the distance computed for each row at the time it was selected. These distances tell us how significant each row is and thus, how significant each basis vector is. We choose a threshold θ for these distances and discard the rows with distances less than this threshold. These discarded rows with their small distances make $\hat{P}$ ill-conditioned rather than singular, and their contributions are likely to result from stochastic (sample) variation rather than reflecting the true probabilities in $\overline{P}$. How we choose this threshold θ is necessarily somewhat arbitrary. We use a second heuristic that attempts to draw the threshold just above the largest distance resulting from the variance of the estimates $\hat{P}_{w,s}$.

We build our history wordlist W by collecting the words that induce the rows we keep. This step dictates the number $n = |W|$ of presentation states in the presentation we will reconstruct. Finally, we apply the QR algorithm (related to QL by transposes) to this edited matrix and select the columns with the n largest distances. The result is that we are left with a square matrix G and a wordlist S. The construction of G guarantees that it is invertible and well-conditioned.

From here, we construct the matrices C^k such that $C^k_{ij} = \pi(ks_j|w_i)$ and let $B^k = C^k G^{-1}$ as in section 5.1. And if we let $p_i = \pi(s_i|\lambda)$ for each word s_i in the future wordlist S, we can set $\gamma = (p_1, \ldots, p_n)G^{-1}$. The same calculations we used in the proof

of theorem 5.1.8 show that $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$ satisfies all the necessary conditions and is in fact a Proto-GHMM. This finishes the construction of the presentation, which we summarize as follows.

Algorithm 5.2.1. If we are given one or more sample sequences of output from a process $\mathcal{P} = \left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$ for which $\mathbf{P}$ is unknown, we may construct a Proto-GHMM $W, \mathcal{X}\left\{B^k\right\}, \lambda$ which approximately represents $\mathcal{P}$ by the following steps.

1. Construct a matrix of estimated word probabilities. This may be done as follows.
   a. Fix a number K and choose the largest l such that each word of length $2l+1$ occurs an average of K times.
   b. Let Y be the set of all words of length l or less.
   c. For each $w, s \in Y$, compute
   $$\hat{P}_{w,s} = \frac{\text{the number of times } ws \text{ occurs}}{\text{the number of times } w \text{ occurs}}. \tag{5.30}$$
2. Find a nonsingular, well-conditioned minor G of P. One way to do this is as follows.
   a. Perform a QL decomposition of $\hat{P}$ to compute a significance for each row.
   b. Fix a threshold θ and discard all rows of $\hat{P}$ with significance less than θ. Let n be the number of rows remaining.
   c. Perform a QR decomposition of the remainder of $\hat{P}$ and select the n columns with the highest significance. These subcolumns of $\hat{P}$ comprise the matrix G.
3. Let the history wordlist W be the set of words associated with the rows of G, and let the future wordlist S be the set of words associated with the columns of G.
4. For each $k \in \mathcal{X}$, construct the matrix C^k with entries $C^k_{ij} = \pi(ks_j|w_i) = \hat{P}_{w_i, ks_j}$
5. For each $k \in \mathcal{X}$, compute $B^k = C^k G^{-1}$.
6. For each $s_i \in S$, compute an estimate $\pi(s_i)$ of $\mathbf{P}(s_i)$.
7. Compute $\gamma = (\pi(s_1), \ldots, \pi(s_n))G^{-1}$.

This computes all the parts of a Proto-GHMM $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$. If this Proto-GHMM is valid, it is a GHMM presentation for a process which has word probabilities close to those of $\mathcal{P}$.

This construction is not completely satisfactory, however. We cannot assert that $\left(W, \mathcal{X}, \left\{B^k\right\}, \gamma\right)$ is a GHMM because we have not shown that it is valid. In fact it

is possible to construct a sample from which a reconstruction yields an invalid Proto-GHMM. As mentioned in section 4.1, determining whether or not a given presentation is valid is quite difficult. We will address this further in section 6.2.

Statistical Considerations Algorithm 5.1.1 produces a Proto-GHMM as a function of a sample. If the sample is a random sample of the output from a process $\mathcal{P}$, the sample is a random variable, and so the Proto-GHMM is also a random variable. Thus the reconstruction algorithm, together with $\mathcal{P}$ and the length of the sample, induce a distribution on the space of Proto-GHMMs.

We would like to be able to describe this distribution, but doing so is quite complicated. One of the difficulties is describing the distribution of $\hat{P}$. Each entry $\pi(s|w)$ in $\hat{P}$ has the form $\frac{X}{Y}$, where X appears to have a binomial distribution, with parameter $\theta = \mathbf{P}(s|w)$ and Y trials, and Y is the random variable describing the number of occurrences of the word w. However, X may not be binomial, because occurrences of s may not be independent of previous occurrences. Similarly, the distribution of Y cannot be described easily. Moreover, entries of $\hat{P}$ are not independent. Although it may appear that the joint distributions of some groups of entries may be a function of a multinomial distribution, this is not the case. Our "trials" are not independent, as they come from examining pieces of the sample sequences, and these pieces may overlap. For example, if the alphabet is $\mathcal{X} = \{0, 1\}$ and the sample consists of a single sequence, then the numbers of occurrences of 01 and 10 may differ by at most one. This complicates any description of the distribution of $\hat{P}$. In addition, even the size of $\hat{P}$ depends on the sample. Although it may be possible to characterize the distribution of $\hat{P}$ in a tractable way, doing so is beyond the scope of this dissertation.

If we were able to express this distribution reasonably, we would manipulate it further in order to derive the distribution it induces on the set of all Proto-GHMMs. To do this, we would look at the output of the QL algorithm as a random variable and examine its distribution. Continuing in this way, we would derive distributions for G, the C^ks, and eventually for γ and the B^ks. However, we cannot proceed with this program because we cannot describe the distribution of $\hat{P}$. Most of the steps would be laborious, and probably not very interesting.

The distribution of the output of QL raises two interesting questions in mathematical statistics. First, given a random matrix $\hat{A}$ which is the sum of an unknown singular matrix A and a random matrix (with zero mean and some assumptions about its variance), how can we best estimate the rank of A? Second, how ill-conditioned can A be — or how small must the variance of the random matrix be — so that we can reliably estimate the rank of A?

Computational Issues The computational issues for the reconstruction algorithm are the numbers of operations and the storage requirements of this algorithm. As is common practice, we will be concerned only with how these quantities scale — their *order* — and not with precise estimation.

First, we need notation for the parameters of the problem. We have used L as the length of the sample, and $m = |\mathcal{X}|$ as the size of the alphabet. Let Y be the large wordlist used to construct $\hat{P}$, and let N be the number of words in Y — this means that $\hat{P}$ is $N \times N$. Let l be the length of the longest word in Y, and let n be the number of words in the minimal wordlists we derive. These parameters are not all independent; we must have $n < N$ and may reasonably expect, assuming the entropy rate of the process is close to the maximum possible, that $l \approx \log_m N$.

To construct the estimates, we need to count the number of occurrences of each word of the form ws for w and s in Y. To do so, we step through the sample. At each step, and for each length up to $2l$, we identify the word of that length which starts at our present location in the sample, and add one to that word's count. (This technique requires minor adjustments involving the ends of sample sequences to produce the estimates exactly as we defined them in terms of cutpoints on page 95.) The number of locations is essentially L, and the number of words at each location is $2l$. The work to be done for each word may be organized so that it has constant time. Thus, the number of operations required to do the estimation has order $\mathcal{O}(lL)$.

The storage requirements of estimation are simple. To estimate $\hat{P}$, we must store one counter for each of N^2 words. In addition, the matrices C^k depend on the counts for words of the form wks, with w and s in Y and $k \in \mathcal{X}$. It is convenient to do the necessary counts for words of length up to $2l+1$ at the same time. This does not change the order of the number of operations, and it increases the storage to $\mathcal{O}\big(N^2 m\big)$.

Constructing minimal wordlists, and a basis for the mixed states, is the next stage of reconstruction. The QL algorithm we use to do this requires $\mathcal{O}(N^3)$ operations and $\mathcal{O}(N^2)$ storage. In practice, this is the most time-consuming part of the algorithm. Computing G^{-1} takes $\mathcal{O}(n^3)$ operations, which is dominated by N^3 and so may be ignored.

Finally, the algorithm constructs the matrices C^k and performs the multiplications $C^k G^{-1}$. There are m of these multiplications, each a product of $n \times n$ matrices, so this stage requires $\mathcal{O}(n^3 m)$ operations and $\mathcal{O}(n^2 m)$ memory. This last quantity is dominated by the storage requirements of the estimation stage, and we will ignore it. Thus, the entire reconstruction algorithm requires

$$\mathcal{O}(lL + N^3 + n^3 m) \tag{5.31}$$

operations and $\mathcal{O}(N^2 m)$ memory.

Of the parameters in expression 5.31, only L and m are known in advance. The number of states n is difficult to estimate — indeed, a good part of the work of the reconstruction algorithm is in estimating it. However, m is usually small enough and n is usually sufficiently smaller than N that $n^3 m$ is small compared to N^3. Thus, the number of operations does not scale strongly with the eventual number of presentation states.

With the heuristic we used to choose Y, the remaining two parameters are related by $l \approx \log N$. This will necessarily be true of any method for choosing Y, since the number of possible words of length l or less is $(m^{l+1} - 1)/(m - 1) \approx m^l$. Thus, l changes slowly compared to N. Because of this and because expression 5.31 depends linearly on l but on the cube of N, we see that N is far more important to the scaling of the operation count.

The heuristic for Y (page 96) gives us a way to estimate l and N. It chooses l so that the words of length $2l+1$ occur an average of K times. There are about L opportunities for such words to occur, so there must be about L/K such words. Assuming all m^{2l+1} possible words of length $2l + 1$ appear in the sample, we have $m^{2l+1} \approx L/K$, or

$$m^l \approx \left(\frac{L}{K}\right)^{\frac{l}{2l+1}} \approx \sqrt{\frac{L}{K}}. \tag{5.32}$$

Taking the base m of both sides gives us the approximation $l \approx \log_m \sqrt{L/K}$. Assuming all possible words of length l occur in the sample, the left-hand side of equation 5.32 is

an estimate of the N produced by the heuristic, so we have $N \approx \sqrt{L/K}$. With these substitutions, the expression 5.31 becomes

$$\mathcal{O}\left(L \log_m \frac{L}{K} + \left(\frac{L}{K}\right)^{\frac{3}{2}} + n^3 m\right) \tag{5.33}$$

and we see that the reconstruction algorithm takes order $L^{\frac{3}{2}}$ operations. Similarly, it requires memory of $\mathcal{O}(mL/K)$.

We conclude this computational analysis a few final comments. First, recall that there is more than one possible choice of the heuristic for selecting Y. The conclusion we have just reached, that the reconstruction algorithm takes time of order $L^{\frac{3}{2}}$, depends explicitly on the choice. Other choices may give higher or lower exponents, and may make the algorithm as a whole better or worse. We have no reason to believe that the heuristic we have used is a particularly good one.

Second, it is worth mentioning the time and memory required by the forward-backward algorithm because the reconstruction algorithm will inevitably be compared to it. The forward-backward algorithm operates by repeatedly making small adjustments to a presentation. It needs $\mathcal{O}(n^2 L)$ operations per iteration and uses $\mathcal{O}(nL + n^2 m)$ storage overall. There are variants that take fewer operations, but these do not appear to be widely used [30,31]. The number of states initially selected is likely to be larger than the n estimated by the reconstruction algorithm because this number must be chosen *a priori* and because it is usually necessary to use more states than are mathematically needed in order to get an HMM which describes the data well. Additionally, the forward-backward algorithm requires an unspecified number of iterations to converge. The author knows no way to estimate how this number of iterations scales with the size of the data set or the complexity of the process.

Finally, the reconstruction algorithm, as presented here, should be thought of as a mathematical draft, rather than a finished program. At this time it has been implemented, but the implementation cannot be considered an optimal coding. There has not been a systematic search either for a good heuristic for Y or for the heuristic used to choose the threshold θ. And there has not been any systematic comparison of the results of this algorithm to those of the forward-backward algorithm.

When the existing implementation is given actual probabilities (that is, we apply QL to $\overline{P}$ instead of $\hat{P}$) for a process defined by a GHMM, it reliably produces a minimal

GHMM which is, to machine precision, equivalent to the original. When it is given a sufficiently large sample of the output from a GHMM, it typically produces a GHMM that represents a process with word probabilities similar to those of the original GHMM. It is prone to two types of failures, however. In one, it produces a GHMM with an excessive number of states because the threshold θ is too low. In the other, it produces an invalid Proto-GHMM. The reconstruction algorithm needs more work on the implementation details in order to make it widely usable, but the overall framework has a solid theoretical grounding and may, in time, replace the forward-backward algorithm.

6 Conclusions and Further Directions

In the preceding chapters, we have introduced process states, described the process states of an HMM, and shown that the span of an HMM's process states is a finite-dimensional vector space. We have introduced GHMMs, shown how to tell when pairs of GHMMs represent the same process, and shown how to construct a minimal GHMM equivalent to a given one. Finally, we have given procedures for constructing a GHMM presentation for a process either from word probabilities or from a sample of output. We will conclude this dissertation by making some observations and by suggesting some directions for further investigation. We will discuss the viewpoint which led to this work, the problem of GHMM validity, and the question of how to find an HMM that is equivalent to a given GHMM. We will end by discussing the implications of this dissertation for the broader field of modeling complex systems.

6.1 Process states and presentations

The results of this dissertation have followed from a few ideas. The first of these is that the process is more fundamental than the presentation which represents it. This idea is present in the dynamical systems literature; see [32,33,11]. The second is that a process has states which are inherent to it, and distinct from the states of any presentation. The third is a question: What are the process states for a process in terms of a presentation that defines it? And the fourth is the observation that if a process has an HMM presentation, then its process states lie in a finite-dimensional vector space. Together, these ideas lead to a viewpoint that is useful for the study of the processes generated by HMMs.

Some previous work on HMMs has defined the processes represented by HMMs, usually in discussing the problem of HMM equivalence [2]. However, these works have kept the focus on the HMMs' presentations themselves, and have worked with processes very little. In this dissertation, on the other hand, the focus is on the processes, and HMMs are of interest primarily as convenient representations of processes. We justify this shift in focus, and the accompanying shift in viewpoint, with the assertion that processes are the more fundamental objects. The process, not the presentation, is almost always the object of the real focus. (There may be a few applications in which a

phenomenon being studied has a known structure, and the presentation states may be chosen so as to have some inherent reality of their own. But in any other use of HMMs, the presentation states themselves have no meaning and the HMM is being used solely as a representation of a process.)

A similar contrast between this dissertation and previous work on HMMs can be made for process states and presentation states. In previous work on HMMs, the word *state* is used exclusively to refer to presentation states. Much of this work uses the vectors we have named *mixed states* [1,3] without referring to them as states of any sort. (The terms *mixed state*, *process state*, and *presentation state* were coined by the author for this work, so they could not have been used in previous works. Mixed states have not been named; process states have been called *causal states* [11], and presentation states have simply been called *states*.) In some works — for example, [1] — it is clear that the authors were aware that mixed states render the future conditionally independent of the past.

Beginning with the process, we are led to define the process state, because process states are inherent to the process, and presentation states are not. And later, when we introduce HMMs, it is natural to ask what their process states look like. With this background, the mixed states become objects with meaning — conditional future distributions — instead of merely being intermediate results in a computation. Thus HMMs, which we use as a convenient way to represent processes, have given us a convenient way to represent process states. From this we see that an HMM's process states lie in a finite-dimensional vector space, and we see the presentation states in their true role as basis vectors for this space.

This viewpoint can lead us in other directions as well. Process states are useful for calculating various statistics. The entropy of a process, for instance, may readily be computed from an HMM presentation by use of mixed states, but not directly from the presentation states. This was done in [1,34,35]

The author has work in progress concerning the computation of other statistics — notably statistical complexity and excess entropy [9,11] — and a classification of processes with HMM presentations. It appears that this perspective will be a useful one for any research involving HMMs.

6.2 Generalized Hidden Markov Models

Many of the results in this dissertation are stated for Generalized Hidden Markov Models instead of Hidden Markov Models. For several reasons, however, the reader may prefer to work with HMMs. Certainly HMMs are more familiar, and the reader is likely to be more comfortable with them than with GHMMs because of their negative entries. There is no question of validity with HMMs — every HMM is valid. Also, HMMs can be interpreted by examining the transition matrices, though such interpretation may be suspect unless equivalent HMMs receive similar interpretations. And finally, there are questions which make sense for HMMs but not for GHMMs, such as, What presentation state is the HMM most likely to be in at time t? (The meaningfulness of the answer is questionable, as discussed above, unless individual states have intrinsic meanings.)

We ask the reader to work with GHMMs for the following reasons. First, when the presentation states are correctly understood as the basis elements for the space containing the process states, the entries in the transition matrices are understood to be coordinates and not probabilities. With this in mind, restricting vectors to the positive cone of the space, in which all coordinates are positive, is unnatural and arbitrary. Second, the results of chapters 4 and 5 use the tools of linear algebra. Use of these tools becomes much more difficult if it is necessary to stay entirely in the positive cone. Furthermore, it is possible — though no examples are known to the author — that there are processes which may be represented with fewer states as GHMMs than as HMMs, or that processes exist that can be represented as GHMMs but not as HMMs.

Determining whether or not a Proto-GHMM is valid — given $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ such that π and $\sum_k T^k$ are unit-sum, is $\pi T^x \vec{1} \geq 0$ for all words x? — seems to be a hard problem. If we simply generate random output, negative "probabilities" of symbols usually show up quickly or not at all. But the absence of negative "probabilities" up to any finite time does not prove that the Proto-GHMM is valid.

We might try the following naive algorithm for testing validity. If we order the set of all finite words, we can compute the probability of each word in turn. When we reach a negative probability, we conclude the Proto-GHMM is invalid and halt. But if it is valid, the process never halts.

The question of validity can be phrased in the following way. Does the set of

reachable process states intersect the set of vectors v for which $vT^k\vec{1} < 0$ for any symbol k? The latter of these sets is a finite union of half-spaces, so the answer does not change if we replace the former set with its convex hull. This suggests the following approach: we take a queue of mixed states, initially containing only the stationary vector π and a convex set S, initially empty. For each vector v that we remove from the queue and for each symbol k, we compute the "probability" $vT^k\vec{1}$ and halt if it is negative. Then we generate the mixed state $u = N\big(vT^k\big)$ which is the result if the process is in the process state corresponding to v and then emits k. If u is not in S, we replace S with the convex hull of $S \cup \{u\}$ and add u to the queue. We continue in this way until the queue is empty.

In essence, this algorithm constructs a subset of the set of all words such that if none of the words in the subset has negative "probability," then the Proto-GHMM is valid. However, this algorithm has the same problem as the naive algorithm. For some GHMMs it will never stop because this subset is still infinite — the queue may never be empty. For the simple nondeterministic source we saw in section 3.5, for example, there is an infinite sequence of words all of which lie outside the convex hull of all the preceding words.

No finite method for generating the convex hull of the set of reachable process states is known, though it may be possible to develop such a method based on linear programming. The work of Heller, which gives results for functions of finite Markov Chains in terms of convex polygonal cones, may contain part of a solution [21]. Indeed, it is not known whether or not this set can always be finitely described. Thus, we do not know of a practical method for testing whether or not a Proto-GHMM is valid.

6.3 Converting GHMMs to HMMs

There is another aspect of our understanding of GHMMs which is unsatisfactory. If we have a GHMM, is there an HMM which is equivalent to it? If so, is there an equivalent HMM that has the same number of states as the GHMM? These questions are open, but in light of the present understanding, we offer the following conjectures.

Conjecture 6.3.1. Given a GHMM, there is an HMM that is equivalent to it.

Conjecture 6.3.2. Given a GHMM, there is an HMM that is equivalent to it with the same number of presentation states.

For a slightly different formulation of GHMMs — which output from states instead of from transitions — Balasubramanian [8] has asserted that conjecture 6.3.1 holds but conjecture 6.3.2 does not. A similar assertion is implied in [7], but neither paper gives any proof or any counterexamples that apply to GHMMs as defined here. Darmadhikari and Heller state results for "regular" functions of a Markov Chain that the author has not applied to GHMMs [10,21]. One or both of these results may show that conjecture 6.3.2 is false. The author of this dissertation expects that both conjectures hold, but has not found a proof.

Whether or not conjectures 6.3.1 and 6.3.2 hold, there are GHMMs for which equivalent HMMs exist. We can construct such GHMMs by conjugating HMMs, or by reconstructing from probabilities. When an equivalent HMM exists, how can we find it? Because this is a matter of converting from a generalized HMM to an ordinary HMM, we will call this the *degeneralization problem*.

There is reason to believe that the degeneralization problem is nontrivial. Suppose, for instance, we have an algorithm that will degeneralize any GHMM for which an equivalent HMM exists. If we apply this algorithm to an invalid Proto-GHMM, it must fail. If we apply it to an arbitrary Proto-GHMM, and it succeeds, we have shown that the Proto-GHMM is valid. Thus, if we have a degeneralization algorithm we have a way to establish the validity of a substantial collection of Proto-GHMMs. Furthermore, if conjecture 6.3.1 is true, the only way this degeneralization algorithm can fail is if the Proto-GHMM is invalid. In this case, our degeneralization algorithm serves as a general test which determines the validity of Proto-GHMMs.

We can give a necessary and sufficient condition for a GHMM to have a HMM equivalent to it, but this condition is so close to restating the definition that it has little value. Given a GHMM, there is an HMM equivalent to it if and only if there exists a convex set C in the space containing the mixed states with a finite number of vertices such that

1. the initial distribution π is in C, and

2. for all vertices v and for all symbols k the vector vT^k lies in the convex hull of $C \cup \left\{\vec{0}\right\}$.

If C exists, then the vertices of C are the presentation states of an HMM equivalent to the given GHMM. As with validity, the problem is finding the convex set.

No degeneralization algorithm exists at present, but one possible approach is known. If we conjugate by a carefully chosen matrix, we can make any particular entry in any given transition matrix nonnegative, with the possible cost of making some other entry negative. It may be possible to choose a matrix that makes all entries simultaneously nonnegative. The task of finding such a matrix may be approached as a maximization task: maximize the sum of the negative entries of the matrices AT^kA^{-1} over the space of $n \times n$ stochastic invertible matrices A.

General multidimensional maximization techniques fail to find suitable matrices even when they are known to exist. This probably occurs because these techniques search for local maxima and the space of invertible matrices is disconnected. However, it may be possible to develop a global technique specialized to the form of this specific problem.

Further, and of particular interest if conjecture 6.3.1 holds but conjecture 6.3.2 fails, it is possible to add states to a GHMM by an operation similar to conjugation. If v is a stochastic row vector of length n, and h is an arbitrary column vector of length n, consider the $n \times (n+1)$ matrix

$$A = (I - hv \quad h) \tag{6.1}$$

and the $(n+1) \times n$ matrix

$$B = \begin{pmatrix} I \\ v \end{pmatrix}. \tag{6.2}$$

Note that both A and B are stochastic and their product is $AB = I$. If $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$ is a GHMM and we let $\tau = \pi A$, and for all k we let $U^k = BT^kA$, then for any suitable V', $\left(V', \mathcal{X}, \left\{U^k\right\}, \tau\right)$ is an $n+1$–state GHMM which is equivalent to $\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$. Thus, if we cannot find an n–state HMM equivalent to our original GHMM, we can search for one with more than n states. A solution to the degeneralization problem may well emerge from these techniques.

6.4 Reconstruction

The reconstruction algorithm is the crowning achievement of this dissertation. The algorithm is novel and it is not a variation on an older algorithm. It operates directly, without requiring an initial configuration and without making iterative adjustments to the model. The connection between the source data and the resulting Proto-GHMM through the probability estimates is natural and clear.

The practical implementation of the reconstruction algorithm (that builds GHMMs from samples) shows considerable promise, but needs further development in order to be widely useful. As discussed at the end of section 5.2, this work includes looking for better heuristics, considering other possible estimators, a proper statistical study, and general fine-tuning.

6.5 Last remarks

The problem of HMM (or GHMM) reconstruction may be thought of as a "complex estimation problem" or a problem of "model inversion with uncertainty." That is, we want to take a random sample and build a model from it. But what we have is a class of models and a method of generating random samples from a model in this class. However, the method of generating samples, while not complicated, is involved enough that there is not a practical way to "invert" it. There are many model classes to which this description applies, including a number which have applications: Hidden Markov Models, neural networks, and a number of less well known model classes used in pattern recognition.

In recent years, there has been a proliferation of work attempting to solve these problems by iterated improvement. Forward-backward and back propagation, for HMMs and neural nets, respectively, are probably the oldest of these. Many of these approaches use versions of the expectation-maximization (EM) algorithm, which is a general algorithm that may be specialized to address many situations. Others use stochastic optimization algorithms, such as simulated annealing and genetic algorithms. Most of these approaches have similar failings: they get stuck in local optima, they may depend strongly on the initial (random) model, and they often require larger models than is appropriate to get a decent answer. Nonetheless, these approaches are being used because these algorithms

provide a way to get some sort of answer to questions that previously would have been intractable. It is not inappropriate to describe these methods as crude tools by which one can bring a computer to bear on modelling problem.

This dissertation has taken a different approach. We began by attempting to better understand HMMs on a theoretical level. This led to an attempt to characterize the class of processes which could be represented by HMMs. With the better theoretical understanding we had gained, a new approach to the reconstruction problem became accessible. Although it remains to be seen how the reconstruction algorithm will serve in practice, there is reason to believe that in time it may replace forward-backward as the main method by which HMMs are constructed from the sample data.

It is the author's contention that this experience is applicable to other problems of model inversion with uncertainty. It is undoubtedly easier to implement an iterative improvement algorithm than to do this sort of theoretical study, so iterative improvement schemes may remain useful in the study of new model classes. After a model class has proved its utility, however, it is valuable to gain the theoretical understanding necessary to develop more direct algorithms.

Appendix A Notation

Processes

$\mathcal{X}$	Alphabet (set of symbols), canonically $\{0, 1, \ldots, m-1\}$.
$\mathcal{X}^{\mathbb{Z}}$	The set of bi-infinite sequences of symbols in $\mathcal{X}$.
$\mathcal{X}^-$, $\mathcal{X}^+$	The history and future sequence spaces, which are sets of semi-infinite sequences.
x	A bi-infinite sequence, that is, an element of $\mathcal{X}^{\mathbb{Z}}$.
x^-, x^+	Semi-infinite history and future sequences, which are elements of $\mathcal{X}^-$ and $\mathcal{X}^+$ respectively.
x_i	ith element in sequence x, x^+, or x^-.
$\mathcal{X}^*$	The set of all (finite-length) words of symbols in $\mathcal{X}$.
w, s	Words in $\mathcal{X}^*$ or subsequences, especially history suffixes, which are subsequences with end time -1, or next words, which are subsequences with end time 0.
$\lvert w \rvert$	The length of a word or subsequence w.
A_w	The set of (bi-infinite) sequences which match the word w.
$\mathbb{X}$	The σ-field on $\mathcal{X}^{\mathbb{Z}}$ generated by the cylinder sets.
$\mathbb{F}$	The future σ-field, subset of $\mathbb{X}$.
$\mathbb{H}$	The history σ-field, subset of $\mathbb{X}$.
$\mathbf{P}$	A probability distribution on measureable space $(\mathcal{X}^{\mathbb{Z}}, \mathbb{X})$.
$\mathcal{P}$	A process, which is a measure space $(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P})$ in which $\mathbf{P}$ is stationary.
$\mathcal{N}$	The set of bad histories, on which we do not condition.
R	The set of non-null history suffixes — that is, the set of words with positive probability — on which conditioning is well defined.
$\mathbf{P}(\cdot\lvert s)$	The conditional distribution on the future induced by a word s. If $s \in R$, $\mathbf{P}(\cdot\lvert s)$ is a reachable process state.
$\mathbf{P}(\cdot\lvert \mathrm{x}^-)$	The conditional distribution on the future induced by a history x^-. If $s \notin \mathcal{N}$, $\mathbf{P}(\cdot\lvert \mathrm{x}^-)$ is an infinitely preceded process state.
$\mathbf{A}$	A process state, which is a conditional distribution on the future induced by conditioning on a history, a history suffix, or both.

Hidden Markov Models

V	Set of presentation states or Markov chain states.
$\|V\|$	The size — that is, number of elements — of V.
π	The initial distribution of a Markov chain or an HMM, which we always take to be stationary.
P	The transition matrix of a Markov Chain (V, P, π).
i, j	Indices which refer to specific presentation states.
k	An index which refers to a symbol in the alphabet $\mathcal{X}$.
T^k	A joint matrix of a Hidden Markov Model, also sometimes referred to as a transition matrix. Note that the superscript k is an index, not an exponent.
T^k_{ij}	An entry in a joint matrix: if the HMM is in the presentation state i, T^k_{ij} is the probability that the HMM will make a transition to presentation state j and emit the symbol k.
T^w	For any word $w = w_1 \dots w_n$, T^w is the product $T^{w_1} T^{w_2} \cdots T^{w_n}$.
$\left(V, \mathcal{X}, \left\{T^k\right\}, \pi\right)$	A Hidden Markov Model. The of matrices $\left\{T^k\right\}$ contains one matrix for each symbol $k \in \mathcal{X}$.
B	The output matrix for an HMM which emits symbols from states rather than from transitions. Such an HMM may be converted to joint matrix form by assigning $T^k_{ij} = P_{ij} B_{jk}$.
$\vec{1}$	A column vector containing all 1s. Size is implied by context.
N	The normalization operator: if v is a row vector, $N(v)$ is v multiplied by a scalar so that its entries sum to 1.
$\eta(s)$, $\eta(\mathrm{x}^-)$	The mixed state induced by a history suffix $s \in R$ or a history $\mathrm{x}^- \in \mathcal{N}$. For s, we have $\eta(s) = N(\pi T^s)$.
μ, ν	Mixed states of an HMM, which are row vectors satisfying $\mu\vec{1} = 1$.
$\mathcal{W}$	The space of all signed measures on the future.
$\mathcal{U}$	The span of the reachable process states, which is a subspace of $\mathcal{W}$.

Generalized Hidden Markov Models

M	Conjugation matrix, which is an invertible unit-sum matrix.
$\mathcal{H}$, $\mathcal{F}$	History and future vector spaces, which are spaces of row and column vectors.
$K_{\mathcal{F}}$	The subspace of $\mathcal{H}$ consisting of all vectors which are sent to zero by multiplication on the right by every vector in $\mathcal{F}$.
$K_{\mathcal{H}}$	The subspace of $\mathcal{F}$ consisting of all vectors which are sent to zero by multiplication on the left by every vector in $\mathcal{H}$.
H	A matrix, the rows of which are mixed states and form a basis for $\mathcal{H}/K_{\mathcal{F}}$.
F	A matrix, the columns of which have the form $T^s\vec{1}$ and form a basis for $\mathcal{F}/K_{\mathcal{H}}$.
W, S	History and future wordlists. The rows of H are the mixed states induced by the words in W, and the columns of F are the vectors $T^s\vec{1}$ for the words $s \in S$.
B^k	A joint matrix for the standard presentation. B^k solves $HT^kF = B^kHF$.
γ	The initial vector of the standard presentation. γ solves $\pi F = \gamma HF$.

Reconstruction

q_i	The ith word in a fixed ordering of $\mathcal{X}^*$.
P	The $\mathbb{N} \times \mathbb{N}$ matrix with entries $P_{ij} = \mathbf{P}(q_j \mid q_i)$.
$\overline{P}$	The $\lvert W' \rvert \times \lvert S' \rvert$ truncation of P containing those rows and columns which correspond to words in the large wordlists W' and S' respectively.
$\pi(s \mid w)$	The frequency-count estimate of $\mathbf{P}(s \mid w)$ estimated from sample data.
$\hat{P}$	A $\lvert W' \rvert \times \lvert S' \rvert$ approximation to $\overline{P}$ estimated from sample data.
$r(w)$	The row of P which corresponds to the word w.
$c(s)$	The column of P which corresponds to the word s.
$r'(w)$	The sub-row of $r(w)$ containing only those columns corresponding to words in S
$c'(s)$	The sub-column of $c(s)$ containing only those rows corresponding to words in W.
G	The submatrix of P containing those rows and columns which correspond to words in the wordlists W and S respectively. G is normally chosen to be invertible.
C^k	The $\lvert W \rvert \times \lvert S \rvert$ matrix with entries $C^k_{ij} = \mathbf{P}(ks_j \mid w_i)$.
B^k	A joint matrix of the reconstructed presentation, given by $B^k = C^k G^{-1}$

Appendix B Selected Probability Theory

This appendix outlines selected elements of probability theory that are used in this dissertation. For a thorough presentation of this material, see any text on the subject, for example [14,13].

B.1 Kolmogorov's Extension Theorem and Process Existence

The purpose of this section is to prove the following theorem, which is our tool for showing that processes exist.

Theorem B.1.1. Given a map $f : \mathcal{X}^* \to [0,1]$ statisfying

1. $f(\lambda) = 1$, and
2. For all words $w \in \mathcal{X}^*$, $f(w) = \sum_{z \in \mathcal{X}} f(zw) = \sum_{z \in \mathcal{X}} f(wz)$,

there is a unique (stationary) process $\mathcal{P} = \left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$ such that for all $w \in \mathcal{X}^*$, $\mathbf{P}(w) = f(w)$.

This result is derived from Kolmogorov's extension theorem, which appears in the literature in several forms. None of the forms the author has seen, however, can be transformed into the form we need without an unreasonable amount of manipulation, thus this section. We will use $\mathcal{R}^n$ and $\mathcal{R}^{\mathbb{N}}$ to refer to the Borel σ-fields on $\mathbb{R}^n$ and $\mathbb{R}^{\mathbb{N}}$, respectively.

Theorem B.1.2. (Kolmogorov's Extension Theorem [13 p. 428]) Suppose that we are given probability measures μ_n on $(\mathbb{R}^n, \mathcal{R}^n)$ that are consistent; that is,

$$\mu_{n+1}((a_1, b_1] \times \ldots \times (a_n, a_n] \times \mathbf{R}) = \mu_n((a_1, b_1] \times \ldots \times (a_n, a_n]). \tag{B.1}$$

Then there is a unique probability measure $\overline{\mathbf{P}}$ on $\left(\mathbb{R}^{\mathbb{N}}, \mathcal{R}^{\mathbb{N}}\right)$ with

$$\overline{\mathbf{P}}(x | x \in (a_i, b_i], 1 \leq i \leq n) = \mu_n((a_1, b_1] \times \ldots \times (a_n, b_n]). \tag{B.2}$$

There are three diferences between the probability measure $\left(\mathbb{R}^{\mathbb{N}}, \mathcal{R}^{\mathbb{N}}, \overline{\mathbf{P}}\right)$ shown to exist by Kolmogorov's Extension Theorem and a stationary process $\left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$. We will need to surmount all three to prove theorem B.1.1. First, $\mathbb{R}^{\mathbb{N}}$ is a product of copies of the real numbers and $\mathcal{X}^{\mathbb{Z}}$ is a product of copies of the finite discrete set $\mathcal{X}$. We will

deal with this by an injective map $g : \mathcal{X} \to \mathbb{R}$. Second, and most troublesome, elements of $\mathbb{R}^{\mathbb{N}}$ are semi-infinite sequences and elements of $\mathcal{X}^{\mathbb{Z}}$ are bi-infinite sequences. Our trick for working around this difficulty makes use of a bijective map $h : \mathbb{N} \to \mathbb{Z}$, and involves considering the integers in the order $0, 1, -1, 2, -2, \ldots$. This has an unfortunate effect on the readability of the proofs. Last, $\overline{\mathbf{P}}$ need not be stationary nor does it need to satisfy any similar condition. We will use Kolmogorov's Extension Theorem to show that $\mathbf{P}$ exists, and then prove separately that it is stationary.

Before we begin, we will introduce several functions and some notation. First, we define g to be any injective map $g : \mathcal{X} \to \mathbb{R}$. It will not matter what the images of particular symbols $x \in \mathcal{X}$ are, as long as they are different.

Second, we define the bijective map $h : \mathbb{N} \to \mathbb{Z}$ mentioned above by

$$h(n) = \begin{cases} n/2 & n \text{ even} \\ (1-n)/2 & n \text{ odd.} \end{cases} \tag{B.3}$$

Its inverse is given by

$$h^{-1}(y) = \begin{cases} 2y & y > 0 \\ 1 - 2y & y \leq 0. \end{cases} \tag{B.4}$$

In effect, h alternately returns positive and negative integers. It maps $1, 2, 3, 4, 5$ to $0, 1, -1, 2, -2$ respectively, and h^{-1} maps $-2, -1, 0, 1, 2$ to $5, 3, 1, 2, 4$ in that order. Several more functions are defined in terms of h: $J(n)$ is a set-valued function $J(n) = \left\{y \in \mathbb{Z} | h^{-1}(y) \leq n\right\}$, and $d(n)$ and $c(n)$ are respectively the largest and smallest values in $J(n)$. Because we will use $J(n)$ primarily as an index set, we will consider its elements in a particular order, namely increasing numerical order. These functions can be characterized by the following equations:

$$c(n) = \min(h(n), h(n-1)) = \begin{cases} \frac{2-n}{2} & n \text{ even} \\ \frac{1-n}{2} & n \text{ odd} \end{cases} \tag{B.5}$$

$$d(n) = \max(h(n), h(n-1)) = \begin{cases} \frac{n}{2} & n \text{ even} \\ \frac{n-1}{2} & n \text{ odd} \end{cases} \tag{B.6}$$

$$J(n) = \{c(n), \ldots, d(n)\}. \tag{B.7}$$

The reader may wish to verify a few facts which will be needed presently. If n is even, we have $h(n+1) < 0$, $c(n+1) = c(n) - 1$ and $d(n+1) = d(n)$. If n is odd, we have $h(n+1) > 0$, $c(n+1) = c(n)$ and $d(n+1) = d(n) + 1$.

Now we move on to sets of subsequences. We will use $\mathcal{X}^{[a,b]}$ to denote the set of all subsequences $x_a \dots x_b$ with start time a and end time b. Similarly, we will use $\mathcal{X}^{J(n)}$ to denote the set of all subsequences $x_{c(n)} \dots x_{d(n)}$ with start time $c(n)$ and end time $d(n)$. Because $|J(n)|$ is always equal to n, $\mathcal{X}^{J(n)}$ is a product of n copies of $\mathcal{X}$. In fact, $\mathcal{X}^{J(n)}$ is identical to $\mathcal{X}^n$ except that the coordinates of $\mathcal{X}^{J(n)}$ are labeled with $c(n), c(n+1), \dots, d(n)$ instead of $1, 2, \dots, n$. Thus, if n is odd, $\mathcal{X}^{J(n+1)} = \mathcal{X}^{J(n)} \times \mathcal{X}$, whereas if n is even, $\mathcal{X}^{J(n+1)} = \mathcal{X} \times \mathcal{X}^{J(n)}$.

In addition, we need to define a function that takes subsequences in $\mathcal{X}^{J(n)}$ to subsequences in $\mathbb{R}^n$ and a closely related function that takes bi-infinite sequences in $\mathcal{X}^{\mathbb{Z}}$ to semi-infinite sequences in $\mathbb{R}^{\mathbb{N}}$. We will denote both of these functions by H. They will reorder their argument's coordinates and map them into $\mathbb{R}$. If $x \in \mathcal{X}^{J(n)}$, then there is a subsequence $v \in \mathbb{R}^n$ that satisfies $v_i = g\big(x_{h(i)}\big)$ for all $i \in 1, \dots, n$. We define $H : \mathcal{X}^{J(n)} \to \mathbb{R}^n$ by $H(x) = v$. For example, if $x = x_{-1}x_0x_1x_2 \in \mathcal{X}^{J(4)}$, then

$$H(x_{-1}x_0x_1x_2) = g(x_0)g(x_1)g(x_{-1})g(x_2). \tag{B.8}$$

For an arbitrary $x = x_{c(n)} \dots x_{d(n)} \in \mathcal{X}^{J(n)}$, we have

$$H\big(x_{c(n)}x_{c(n)+1} \dots x_{d(n)}\big) = g(x_0)g(x_1) \dots g\big(x_{c(n)+d(n)+1}\big). \tag{B.9}$$

Similarly, if $\mathbf{x} \in \mathcal{X}^{\mathbb{Z}}$, then there exists $\mathbf{v} \in \mathbb{R}^{\mathbb{N}}$ such that for all $i \in \mathbb{Z}$, the coordinate v_i satisfies $v_i = g\big(x_{h(i)}\big)$. We define $H : \mathcal{X}^{\mathbb{Z}} \to \mathbb{R}^{\mathbb{N}}$ by $H(\mathbf{x}) = \mathbf{v}$. Neither version of H is invertible, because g is not invertible. However, they do have set inverses. For instance, if $A \subset \mathbb{R}^n$ then $H^{-1}(A) = \Big\{x \in \mathcal{X}^{J(n)} | H(x) \in A\Big\}$.

Next, we define an *indexed product set* S to be a subset of $\mathcal{X}^{[a,b]}$ of the form $S = S_a \times \ldots \times S_b$, where $S_i \subset \mathcal{X}$. If $S = S_a \times \ldots \times S_b$ is an index product set, and $S' \subset \mathcal{X}$, then $S \times S'$ is an indexed product set contained in $\mathcal{X}^{[a,b+1]}$. We define the cylinder map Cyl which takes indexed product sets to sets of sequences as follows:

$$Cyl(S) = \Big\{\mathbf{x} \in \mathcal{X}^{\mathbb{Z}} | x_i \in S_i,\ a \le i \le b\Big\}. \tag{B.10}$$

That is, $Cyl(S)$ is the set of all sequences in $\mathcal{X}^{\mathbb{Z}}$ that match a subsequence in S. We will also define the shift map T on indexed product sets by

$$T(S) = \Big\{x \in \mathcal{X}^{[a-1,b-1]} | x_i \in S_{i+1},\ a \le i+1 \le b\Big\}. \tag{B.11}$$

(The reader may have noticed that we are using the same notation for the shift map on indexed product sets as we use on sequences.)

Lemma B.1.3. The following facts about Cyl and T may be easily verified, and we will give no proof.

1. $Cyl(S) = Cyl(S \times \mathcal{X}) = Cyl(\mathcal{X} \times S)$
2. $T(S \times \mathcal{X}) = T(S) \times \mathcal{X} = \mathcal{X} \times T(S)$
3. $T(Cyl(S)) = Cyl(T(S))$

One special interaction is worth noting. Let $S \subset \mathcal{X}^{J(n)}$ be an indexed product set $S = S_{c(n)} \times \ldots \times S_{d(n)}$ and let

$$A = Cyl(S) = \Big\{\mathbf{x} \in \mathcal{X}^{\mathbb{Z}} | x_i \in S_i \text{ for all } i,\ c(n) \leq i \leq d(n)\Big\}. \tag{B.12}$$

Then we have

$$T(S) = \Big\{\mathbf{x} \in \mathcal{X}^{[c(n)-1,d(n)-1]} | x_i \in S_{i+1},\ c(n) \leq i+1 \leq d(n)\Big\}, \tag{B.13}$$

$$T(A) = \Big\{\mathbf{x} \in \mathcal{X}^{\mathbb{Z}} | x_i \in S_{i+1},\ c(n) \leq i+1 \leq d(n)\Big\}, \tag{B.14}$$

and $T(A) = Cyl(T(S))$ and thus by lemma B.1.3, $T(A) = Cyl(T(S) \times \mathcal{X})$.

Finally, notice that for any indexed product set $S \in \mathcal{X}^{[a,b]}$ there is a set $\overline{S} \subset \mathcal{X}^{J(n)}$, where $n = \min(-2a, 2b+1)$, such that $Cyl(S) = Cyl(\overline{S})$. Since every cylinder set can be written as $Cyl(S)$ for some $S \in \mathcal{X}^{[a,b]}$, this means that every cylinder set can be written as $Cyl(\overline{S})$ for some $\overline{S} \in \mathcal{X}^{J(n)}$.

At last, we are ready to state and prove a result. This is Kolmogorov's extension theorem in a form which applies to processes.

Theorem B.1.4 (Bidirectional Discrete version of Kolmogorov's Extension Theorem). Suppose we have a sequence of measures ν_n on $\mathcal{X}^{J(n)}$ which satisfy the following conditions for all n and for all indexed product sets $S \subset \mathcal{X}^{J(n)}$. If n is odd,

$$\nu_n(S) = \nu_{n+1}(S \times \mathcal{X}) \tag{B.15}$$

and if n is even,

$$\nu_n(S) = \nu_{n+1}(\mathcal{X} \times S) \text{ and} \tag{B.16}$$

$$\nu_n(S) = \nu_{n+1}(T(S) \times \mathcal{X}). \tag{B.17}$$

Then there exists a unique stationary process $\mathcal{P} = \left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$ such that, for all n and for all $S \subset \mathcal{X}^{J(n)}$,

$$\mathbf{P}(Cyl(S)) = \nu_n(S). \tag{B.18}$$

The proof of this statement is in two parts. In the first, we show that $\mathbf{P}$ exists using the awkward mapping tricks defined above, theorem B.1.2 and equations B.15 and B.16. In the second, we use equations B.15, B.16, and B.17 to show that $\mathbf{P}$ is stationary.

Proof. We define a sequence of measures μ_n on $\mathbb{R}^n$ as follows: if $A \subset \mathbb{R}^n$, we define $\mu_n(A) = \nu_n\big(H^{-1}(A)\big)$, that is

$$\mu_n(A) = \nu_n\left\{x \in \mathcal{X}^{J(n)} | H(x) \in A\right\} \tag{B.19}$$

The following calculation establishes that the μ_ns are consistent. By definition,

$$\mu_{n+1}(A \times \mathbb{R}) = \nu_{n+1}\big(H^{-1}(A \times \mathbb{R})\big). \tag{B.20}$$

Now, $H^{-1}(A \times \mathbb{R})$ can be rewritten as

$$H^{-1}(A \times \mathbb{R}) = \begin{cases} H^{-1}(A) \times \mathcal{X} & n \text{ odd} \\ \mathcal{X} \times H^{-1}(A) & n \text{ even,} \end{cases} \tag{B.21}$$

so we have

$$\mu_{n+1}(A \times \mathbb{R}) = \begin{cases} \nu_{n+1}\big(H^{-1}(A) \times \mathcal{X}\big) & n \text{ odd} \\ \nu_{n+1}\big(\mathcal{X} \times H^{-1}(A)\big) & n \text{ even.} \end{cases} \tag{B.22}$$

And by applying equations B.15 and B.16 to the right-hand sides, we get

$$\begin{aligned} \mu_{n+1}(A \times \mathbb{R}) &= \nu_n\big(H^{-1}(A)\big) \\ &= \mu_n(A). \end{aligned} \tag{B.23}$$

Thus, we can apply Kolmogorov's Extension Theorem to the measures μ_n, which gives us a unique measure $\overline{\mathbf{P}}$ on $\left(\mathbb{R}^{\mathbb{N}}, \mathcal{R}^{\mathbb{N}}\right)$ which agrees with the μ_n: if A_i is a Borel set for all $i \in 1, \ldots, n$ and $A = A_1 \times \ldots \times A_n$, then

$$\overline{\mathbf{P}}(x | x_i \in A_i,\ i \in 1, \ldots, n). \tag{B.24}$$

Now we define $\mathbf{P}$. For all $S \in \mathbb{X}$,

$$\mathbf{P}(S) = \overline{\mathbf{P}}(H(x) | x \in S). \tag{B.25}$$

This $\mathbf{P}$ is in fact an extension of the ν_ns. For all $S \in \mathcal{X}^{J(n)}$, we have

$$\begin{aligned}\nu_n(S) &= \mu_n(H(S)) \\ &= \overline{\mathbf{P}}\Big(a \in \mathbb{R}^{\mathbb{N}} | a_1 \dots a_n \in H(S)\Big) \\ &= \mathbf{P}\Big(\mathbf{x} \in \mathcal{X}^{\mathbb{Z}} | x_{c(n)} \dots x_{d(n)} \in S\Big) = \mathbf{P}(Cyl(S))\end{aligned} \tag{B.26}$$

Thus, we have shown that $\mathbf{P}$ exists.

To show that $\mathbf{P}$ is stationary, we need to show that $\mathbf{P}(T(A)) = \mathbf{P}(A)$ for a sufficiently rich set of $A \in \mathbb{X}$. Any collection which contains all the cylinder sets will suffice. The collection we choose is

$$\mathcal{A} = \Big\{Cyl(S) | \text{ indexed product sets } S \subset \mathcal{X}^{J(n)} \text{ for some } n\Big\}. \tag{B.27}$$

Thus, every $A \in \mathcal{A}$ has an associated n and an associated S. (Of course, there will be more than one suitable n, S pair, but there will be a smallest n and a unique associated S, and these are the n and S to which we refer.)

If n is even, equation B.17 gives us

$$\begin{aligned}\mathbf{P}(A) &= \nu_n(S) \\ &= \nu_{n+1}(T(S) \times \mathcal{X}) \\ &= \mathbf{P}(Cyl(T(S) \times \mathcal{X})).\end{aligned} \tag{B.28}$$

And since $Cyl(T(S) \times \mathcal{X}) = Cyl(T(S)) = T(A)$, this becomes $\mathbf{P}(A) = \mathbf{P}(T(A))$.

If n is odd, we expand S by one and then do a similar calculation.

$$\begin{aligned}\mathbf{P}(A) &= \nu_n(S) \\ &= \nu_{n+1}(S \times \mathcal{X}) \\ &= \nu_{n+2}(T(S \times \mathcal{X}) \times \mathcal{X}) \\ &= \mathbf{P}(Cyl(T(S \times \mathcal{X}) \times \mathcal{X}))\end{aligned} \tag{B.29}$$

Here we can again apply lemma B.1.3 to get $Cyl(T(S \times \mathcal{X}) \times \mathcal{X}) = T(A)$, and thus $\mathbf{P}(A) = \mathbf{P}(T(A))$.■

Now we are ready to prove theorem B.1.1, which we restate here:

Theorem B.1.1. Given a map $f : \mathcal{X}^* \to [0, 1]$ statisfying

1. $f(\lambda) = 1$, and
2. For all words $w \in \mathcal{X}^*$,

$$f(w) = \sum_{z \in \mathcal{X}} f(zw) = \sum_{z \in \mathcal{X}} f(wz), \tag{B.30}$$

there is a unique stationary process $\mathcal{P} = \left(\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P}\right)$ such that for all $w \in \mathcal{X}^*$, $\mathbf{P}(w) = f(w)$.

The proof will proceed as follows: we will construct measures ν_n which satisfy equations B.15, B.16, and B.17, and then apply Theorem B.1.4 to get the result.

Proof. For all $w \in \mathcal{X}^*$ and $|w| = n$, let $S = \{w\}$ and define

$$\nu_n(S) = f(w). \tag{B.31}$$

For a general $S \subset \mathcal{X}^{J(n)}$, S is a disjoint union of sets of the form $S_w = \{w\}$. Thus we may safely define the measure ν_n on all subsets of $\mathcal{X}^{J(n)}$ by

$$\nu_n(S) = \sum_{w \in S} \nu_n(S_w) = \sum_{w \in S} f(w). \tag{B.32}$$

We need to show that ν_n is a probability measure; that is, we need to show that $\nu_n\left(\mathcal{X}^{J(n)}\right) = 1$. We will do this by induction on n. If $n = 0$ then $\mathcal{X}^{J(n)} = \{\lambda\}$ and we are given $f(\lambda) = 1$, so ν_1 is a probability measure. The induction step depends on equation B.30, and the odd and even cases must be done separately. If n is odd and ν_n is a probability distribution, then we have

$$\begin{aligned} \nu_{n+1}\left(\mathcal{X}^{J(n+1)}\right) &= \sum_{w \in \mathcal{X}^{J(n+1)}} f(w) \\ &= \sum_{w \in \mathcal{X}^{J(n)}} \sum_{x \in \mathcal{X}} f(wx) \\ &= \sum_{w \in \mathcal{X}^{J(n)}} f(w) \\ &= \nu_n\left(\mathcal{X}^{J(n)}\right) = 1. \end{aligned} \tag{B.33}$$

If n is even and ν_n is a probability measure, then the calculation is the same except that we write $f(xw)$ in place of $f(wx)$ and we use the other half of equation B.30.

Now we must show that equations B.15, B.16, and B.17 are satisfied. We will establish each of them using equation B.30 much as before. First equation B.15, assuming n is odd:

$$\begin{aligned} \nu_n(S) = \sum_{w \in S} f(w) &= \sum_{w \in S} \sum_{x \in \mathcal{X}} f(wx) \\ &= \sum_{z \in (S \times \mathcal{X})} f(z) \\ &= \nu_{n+1}(S \times \mathcal{X}). \end{aligned} \tag{B.34}$$

Second, equation B.16, assuming n is even:

$$\begin{aligned}\nu_n(S) = \sum_{w\in S} f(w) &= \sum_{w\in S}\sum_{x\in\mathcal{X}} f(xw) \\ &= \sum_{z\in(\mathcal{X}\times\mathcal{S})} f(z) \\ &= \nu_{n+1}(\mathcal{X}\times S)\end{aligned} \tag{B.35}$$

Lastly, equation B.17, again assuming n is even:

$$\begin{aligned}\nu_n(S) = \sum_{w\in S} f(w) &= \sum_{w\in S}\sum_{x\in\mathcal{X}} f(wx) \\ &= \sum_{z\in(S\times\mathcal{X})} f(z).\end{aligned} \tag{B.36}$$

But this time, $S\times\mathcal{X} \not\subset \mathcal{X}^{J(n)}$, so the expression $\nu_{n+1}(S\times\mathcal{X})$ does not make sense. However, we do have $T(S\times\mathcal{X}) = T(S)\times\mathcal{X} \subset \mathcal{X}^{J(n)}$, and f does not depend on time indices, so we have

$$\begin{aligned}\nu_n(S) = \sum_{z\in(S\times\mathcal{X})} f(z) &= \sum_{z\in(T(S)\times\mathcal{X})} f(z) \\ &= \nu_{n+1}(T(S)\times\mathcal{X}).\end{aligned} \tag{B.37}$$

We have now shown that the ν_ns satisfy all of the conditions of theorem B.1.4. Thus, applying this theorem, we have shown that there exists a unique stationary process $\mathcal{P} = (\mathcal{X}^{\mathbb{Z}}, \mathbb{X}, \mathbf{P})$ such that for all n and for all $S\subset\mathcal{X}^{J(n)}$, we have $\mathbf{P}(Cyl(S)) = \nu_n(S)$. Therefore, if we let $S = \{w\}$ for any length n word w, we have

$$f(w) = \nu_n(S) = \mathbf{P}(Cyl(S)) = \mathbf{P}(w). \tag{B.38}$$

So we are done.■

B.2 Martingales

This section presents the martingale convergence results needed in chapters 2 and 3. Let X be a random variable on a probability space $(\Omega, \mathcal{F}, \mathbf{P})$, and let $\mathcal{G}$ be a sub-σ-field of $\mathcal{F}$. That is, $\mathcal{G}$ is a σ-field and $\mathcal{G}\subset\mathcal{F}$ as sets of sets.

Definition B.2.1. The *conditional expectation* of X with respect to $\mathcal{G}$ is any random variable Y that satisfies

1. Y is measureable with respect to $\mathcal{G}$, and
2. for all $A\in\mathcal{G}$,

$$\int_A X\,d\mathbf{P} = \int_A Y\,d\mathbf{P}. \tag{B.39}$$

Several facts about conditional expectation are worth noting. First, conditional expectations exist on all the probability spaces considered in this dissertation. Second, conditional expectations are unique up to sets of measure zero — if both Y and Y' satisfy definition B.2.1, then $Y = Y'$ almost everywhere. Third, a conditional expectation is a variable, not a constant. That is, Y is a function on Ω. However, Y is constant almost everywhere on atoms of $\mathcal{G}$. (A set $A \in \mathcal{G}$ is an atom if the only sets in $\mathcal{G}$ which are subsets of A are the empty set and A itself.)

The connection between conditional expectation and the conditional probability of elementary probability is as follows. If Z is a random variable on $(\Omega, \mathcal{F}, \mathbf{P})$, then Z induces a σ-field $\sigma(Z)$ on Ω, namely the smallest σ-field containing all sets of the form $Z^{-1}(B)$ for Borel sets B. Let A be a set in $\mathcal{F}$ with indicator function (characteristic function) 1_A. If we fix a constant $c \in \mathbb{R}$ and evaluate the conditional expectation $\mathbf{E}(1_A|\sigma(Z))$ on $x \in Z^{-1}(c)$, we find that

$$\mathbf{E}(1_A|\sigma(Z))(x) = \mathbf{P}(x \in A|Z = c) \tag{B.40}$$

for almost every such x.

A *filtration* is an increasing sequence of σ-fields $\{\mathcal{F}\} = \mathcal{F}_1 \subset \mathcal{F}_2 \subset \ldots$.

Definition B.2.2. A sequence $\{X\} = X_1, X_2, \ldots$ is a *martingale* with respect to the filtration $\{\mathcal{F}\}$ if

1. for all i, X_i is measureable with respect to $\mathcal{F}_i$, and
2. for all $s, t \in \mathbb{N}$ such that $t > s$, $X_s = \mathbf{E}(X_t|\mathcal{F})$

Martingale convergence theorems are commonly stated like this: if $\{X_n\}$ is a martingale and $\mathbf{E}(|X_n|) < \infty$ for all n, then there exists a random variable X such that X_n converges almost surely to X with $\mathbf{E}(|X|) < \infty$. Note that this statement establishes that the limit X exists, not what X is.

The result needed in this disseratation is this. Given a filtration $\{\mathcal{F}\}$, let $\mathcal{G}$ be the smallest σ-field that contains every $\mathcal{F}_i$. Let A be a set in $\mathcal{F}$, and define X_n to be the conditional expectation $\mathbf{E}(1_A|\mathcal{F}_n)$. Then $\{X\}$ is a martingale with respect to $\{\mathcal{F}\}$. The fact we need is $X_n \to \mathbf{E}(1_A|\mathcal{G})$ almost surely.

The following result appears as theorem 4.3 in [36].

Theorem B.2.3. (Doob's Martingale Convergence Theorem) If $\{\mathcal{F}_n\}$ is an increasing sequence of σ-fields (that is, for all $n \geq 0$, $\mathcal{F}_n$ is a sub-σ-field of $\mathcal{F}_{n+1}$),

and X is a measureable function such that $\mathbf{E}(|X|) < \infty$, then $\lim_{n\to\infty} \mathbf{E}(X|\mathcal{F}_n)$ converges almost surely to $\mathbf{E}(X|\mathcal{F})$, where $\mathcal{F}$ is the smallest σ-field which contains all of the $\mathcal{F}_n$s.

Now the result we need is simply Doob's Martingale convergence theorem restricted to the case in which X is an indicator function.

Corollary B.2.4. If $\{\mathcal{F}_n\}$ is an increasing sequence of σ-fields and A is an event, then $\mathbf{P}(A|\mathcal{F}_n) \to \mathbf{P}(A|\mathcal{F})$ almost surely, where $\mathcal{F}$ is the smallest σ-field which contains all of the $\mathcal{F}_n$s.

Bibliography

[1] D. Blackwell, "The entropy of functions fo finite-state markov chains," in *Transactions of the First Prague Conference on Information Theory, Statistical Decision Functions, Random Processes*, (Prague), pp. 13–20, Publishing house of the Czechoslovak Academy of Sciences, 1957.

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

[3] L. R. Rabiner and B. H. Juang, "An introduction to hidden markov models," *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 3, pp. 4–16, January 1986.

[4] M. J. Russell and R. K. Moore, "Explicit modeling of state occupancy in hidden markov models for speech signals," in *Proceedings ICASSP-85 International Conference on Acoustics, Speech, and Signal Processing*, (New York), pp. 5–8, Institute of Electrical and Electronic Engineers, IEEE, 1985.

[5] K. S. Fu, *Statistical Pattern Classification Using Contextual Information*. Research Studies Press, 1980.

[6] K. Kobayashi, *A Coding-Theoretic Study on Finitary Information Systems*. University of Tokyo, 1987. Dr. Thesis.

[7] H. Ito, S. ichi Amari, and K. Kobayashi, "Identifiability of hidden markov processes and their minimum degrees of freedom," *IEEE Transactions on Information Theory*, vol. 38, pp. 324–333, March 1992. A version of this paper appeared in Electronics and Communications in Japan, vol. 74, pp. 77-84.

[8] V. Balasubramanian, "Equivalence and reduction of hidden markov models," Tech. Rep. A.I. Techhnical Report No. 1370, Massachusetts Institute of Technology, January 1993.

[9] J. P. Crutchfield and K. Young, "Inferring statistical complexity," *Physics Review Letters*, vol. 63, pp. 105–108, 1989.

[10] S. W. Dharmadhikari, "Functions of finite markov chains," *Annals of Mathematical Statistics*, vol. 34, pp. 1022–1032, 1963.

[11] J. P. Crutchfield, "The calculi of emergence: Computation, dynamics, and induction," *Physica D*, vol. 75, pp. 11 – 54, 1994.

[12] A. M. Fraser and A. Dimitriadis, "Forcasting probability densities using hidden markov models with mixed states," in *Time Series Prediction: Predicting the Future and Understanding the Past* (A. Weigend and N. Gershenfeld, eds.), Addison-Wesley, 1993.

[13] R. Durrett, *Probability: Theory and Examples*. Belmont, California: Duxbury, 1991.

[14] K. L. Chung, *A Course in Probability Theory*, vol. 21 of *Probability and mathematical statistics*. Academic Press, 2nd ed., 1974.

[15] J. P. Crutchfield, "Semantics and thermodynamics," in *Nonlinear Modeling and Forecasting* (M. Casdagli and S. Eubank, eds.), vol. XII of *Santa Fe Institute Studies in the Sciences of Complexity*, (Reading, Massachusetts), p. 317, Addison-Wesley, 1991.

[16] B. H. Juang and L. R. Rabiner, "Hidden markov models for speech recognition," *Technometrics*, vol. 33, pp. 251–272, August 1991.

[17] L. R. Rabiner, "A tutorial on hidden markov models selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, February 1989.

[18] D. Des Jardins, "-." personal communication, 1995.

[19] J. P. Crutchfield, "Reconstructing language hierarchies," in *Information Dynamics* (H. A. Atmanspracher and H. Scheingraber, eds.), (New York), p. 45, Plenum, 1991.

[20] E. J. Gilbert, "On the identifiability problem for functions of finite markov chains," *Annals of Mathematical Statistics*, vol. 30, pp. 688–697, 1959.

[21] A. Heller, "On stochastic processes derived from markov chains," *Annals of Mathematical Statistics*, vol. 36, pp. 1286–1291, 65.

[22] L. T. Niles and H. F. Silverman, "Combining hidden markov model and neural network classifiers," in *Proceedings ICASSP-90 International Conference on Acoustics, Speech and Signal Processing*, (New York), pp. 417–420, Institute of Electrical and Electronic Engineers, IEEE, 1990.

[23] B. G. Leroux, "Maximum-likelihood estimation for hidden markov models," *Stochastic Processes and their Applications*, vol. 40, pp. 127–143, 1992.

[24] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite markov chains," *Annals of Mathematical Statistics*, vol. 37, pp. 1554–1563, 1966.

[25] W. Qian and D. M. Titterington, "Estimation of parameters in hidden markov models," *Philosophcal Transactions of the Royal Society, Series A Physical Sciences and Engineering*, vol. 337, pp. 407–28, 16 December 1991.

[26] A. Kehagias, "The local backward-forward algorithm," in *IJCNN 91 Seattle: International Joint Conference on Neural Networks*, (New York), pp. 321–326 in Volume 2 of 2, IEEE, IEEE, 1991.

[27] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[28] S. W. Dharmadhikari, "Sufficient conditions for a stationary process to be a function of finite markov chain," *Annals of Mathematical Statistics*, vol. 34, pp. 1033–1041, 1963.

[29] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipies in C*. Cambridge University Press, first ed., 1988.

[30] Y. Ephraim, A. Dembo, and L. R. Rabiner, "A minimum discrimination information approach for hidden markov modeling," *IEEE Transactions on Information Theory*, vol. 35, pp. 1001–1013, September 1989.

[31] B. H. Juang and L. R. Rabiner, "The segmental k-means algorithm for estimating parameters of hidden markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, pp. 1639–1641, September 1990.

[32] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. S. w, "Geometry from a time series," *Phys. Rev. Let.*, vol. 45, p. 712, 1980.

[33] F. Takens, "Detecting strange attractors in fluid turbulence," in *Symposium on Dynamical Systems and Turbulence* (D. A. Rand and L. S. Young, eds.), vol. 898, (Berlin), p. 366, Springer-Verlag, 1981.

[34] J. J. Birch, "Approximations for the entropy for functions of markov chains," *Annals of Mathematical Statistics*, vol. 33, pp. 930–938, 1962.

[35] J. J. Birch, "On information rates for finite-state channels," *Information and Control*, vol. 6, pp. 372–380, 63.

[36] J. L. Doob, *Stochastic Processes*. Wiley publications in statistics, Wiley, 1953.

Index

alphabet, 7

bad histories, 17
band-merging process, 19
base word, 8
Baum-Welsh, 83

Cantor, 46
conditional expectation, 123
conditional future distribution, 14
conjugate, 59
cutpoint, 95
cylinder set, 8

degeneralization problem, 108
deterministic, 35

elusive, 20, 22
equivalence class, 14
equivalent, 59

fair coin, 11
filtration, 124
forward-backward, 83
future, 12
  events, 12
  σ-field, 12
  matrix, 65
  space, 12
  wordlist, 65

Generalized Hidden Markov Model, 2, 55, 57
GHMM, 55, 57
Golden Mean Process, 43
good histories, 17

Hidden Markov Model, 1, 28
history, 12
  events, 12
  σ-field, 12
  matrix, 65
  space, 12
  wordlist, 65
HMM, 28

identifiability, 72
indexed product set, 118
induced, 39
inducing set of histories, 18
infinitely preceded, 21
irreducible, 27

joint matrices, 28

labeled directed graph, 33

Markov Chain, 25
martingale, 124
match, 8
minimal, 65
  for a process, 86
minimization, 72
mixed state, 36
mixed state representation, 41
mixed state version, 37
modified nested parentheses process, 34

next symbol, 8
normalize, 36
null history, 13
null word, 13

order, 100

presentation, 25
presentation states, 28
process, 7, 10
  defined by an HMM presentation, 32
process state, 15, 17
process state graph, 20
Proto-Generalized Hidden Markov Model, 56
Proto-GHMM, 56

quasi-initial vector, 76
quasi-presentation, 76
quasi-transition matrix, 76

rank, 87
reachable, 21
realization, 13
reconstruction from a sample, 94
reconstruction problem, 83
recurrent, 21
recurrent component, 27
reducible, 27
reverse state, 85

SDFA, 35
shift map, 9
Simple Nondeterministic Source (SNS), 44
standard initial vector, 73
standard presentation, 72, 73
standard transition matrices, 73
stationary, 9
stationary distribution, 28
stochastic, 53
Stochastic Deterministic Finite Automaton, 35
stochastic finite automaton, 35
strictly transient, 22
subcolumn, 87
subsequence, 8
sufficient, 65
  for a process, 86
symbol, 2, 7
synchronizing word, 24

transient, 22
transition matrix, 29
Two Biased Coins (2BC), 47

underlying Markov Chain, 28
unit-sum, 52

valid, 56

word, 8
wordlist, 65