MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Technical Report No. 1370 January, 1993

Equivalence and Reduction of Hidden Markov Models

Vijay Balasubramanian
(email: vijayb@ai.mit.edu)

Copyright © Massachusetts Institute of Technology, 1993

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-91-J-4038.

Equivalence and Reduction of Hidden Markov Models

by

Vijay Balasubramanian

Submitted to the Department of Electrical Engineering and Computer Science

on May 18, 1992, in partial fulfillment of the

requirements for the degree of

Master of Science in Computer Science and Engineering

Abstract

Hidden Markov Models are one of the most popular and successful techniques used in statistical pattern recognition. However, they are not well understood on a fundamental level. For example, we do not know how to characterize the class of processes that can be well approximated by HMMs. This thesis tries to uncover the source of the intrinsic expressiveness of HMMs by studying when and why two models may represent the same stochastic process. Define two statistical models to be equivalent if they are models of exactly the same process. We use the theorems proved in this thesis to develop polynomial time algorithms to detect equivalence of prior distributions on an HMM, equivalence of HMMs and equivalence of HMMs with fixed priors. We characterize Hidden Markov Models in terms of equivalence classes whose elements represent exactly the same processes and proceed to describe an algorithm to reduce HMMs to essentially unique and minimal, canonical representations. These canonical forms are essentially "smallest representatives" of their equivalence classes, and the number of parameters describing them can be considered a representation for the complexity of the stochastic process they model. On the way to developing our reduction algorithm, we define Generalized Markov Models which relax the positivity constraint on HMM parameters. This generalization is derived by taking the view that an interpretation of model parameters as probabilities is less important than a parsimonious representation of stochastic processes.

Thesis Supervisor: Tomaso Poggio

Title: Professor, Artificial Intelligence Laboratory

Thesis Supervisor: Leslie Niles

Title: Member of the Research Staff, Xerox Palo Alto Research Center

Acknowledgments

I would like to thank Les Niles for being a wonderful advisor to work for - his encouragement and the freedom he gave me were invaluable. Don Kimber, my office-mate at PARC, rekindled my interest in theoretical work during the many inspirational hours we spent excavating puzzles and problems irrelevant to his work and to mine! Don also helped me to clarify many of the issues in this thesis. Xerox PARC, where most of this work was done, was an almost perfect working environment and I would like to congratulate the Xerox Corporation for creating and maintaining such a wonderful lab. I had a very pleasant and intellectually stimulating term at LCS, working with Jon Doyle and the Medical Diagnosis Group while I was writing my thesis. I am particularly grateful to Jon for letting me focus exclusively on navigating LaTeXwhen the dreaded Thesis Deadline disrupted my formerly peaceable existence. Tze-Yun Leong was kind to let me use her cassette player and monopolize our office SPARC during my marathon thesis session. Special thanks go to my sister, Asha, who gave me a copy of "The Little Engine That Could" when I said I would have trouble finishing my thesis. My parents, as always, lent me their strength when I needed it. Finally, I wish to thank the engineer, who has designed a very interesting world.

This research was supported in part by Bountiful Nature, her coffee beans, and her tea leaves. I gratefully acknowledge the labours of the coffee and tea pickers whose efforts kept me awake long enough to produce this document.

Contents

1 Introduction and Basic Definitions **7**

1.1 Overview 7

1.2 Historical Overview 9

1.3 The Major Questions 10

1.3.1 Intuitions and Directions 11

1.3.2 Contributions of This Thesis 12

1.4 Basic Definitions 13

1.4.1 Hidden Markov Models 14

1.4.2 Variations on The Theme 15

1.4.3 Induced Probability Distributions 16

1.5 How To Calculate With HMMS 17

1.6 Roadmap 20

2 Equivalence of HMMs **23**

2.1 Definitions 23

2.2 Equivalence of Priors 25

2.3 Equivalence of Initialized HMMs 31

2.4 Equivalence of Hidden Markov Models 32

3 Reduction to Canonical Forms **43**

3.1 Generalized Markov Models 44

3.1.1 Why Should We Invent GMMs? 44
3.1.2 Definition of GMMs 47
3.1.3 Properties of GMMs 49
3.2 Canonical Dimensions and Forms 53
3.2.1 Results for HMMs 69

4 Further Directions and Conclusions **73**
4.1 Reduction of Initialized HMMs 74
4.2 Reduction While Preserving Paths 74
4.3 Training GMMs 75
4.4 Approximate Equivalence 75
4.5 Practical Applications 76
4.6 Conclusion 78

A HMMs and Probabilistic Automata **79**

Chapter 1

Introduction and Basic Definitions

1.1 Overview

Hidden Markov Models (HMMs) are one of the more popular and successful techniques for pattern recognition in use today. For example, experiments in speech recognition have shown that HMMs can be useful tools in modelling the variability of human speech.([juang91],[lee88],[rabiner86],[bahl88]) Hidden Markov Models have also been used in computational linguistics [kupiec90], in document recognition [kopec91] and in such situations where intrinsic statistical variability in data must be accounted for in order to perform pattern recognition. HMMs are constructed by considering stochastic processes that are probabilistic functions of Markov Chains. The underlying Markov Chain is never directly measured and hence the name *Hidden* Markov Model.[1] An example of an HMM could be the artificial economy of Figure 1.1. The economy in this figure transitions probabilistically between the states *Depressed*, *Normal*, and *Elevated*. The average stock price in each of these states is a probabilistic function of the state. Typically, pattern recognition using Hidden Markov Models is carried out by building HMM source models for stochastic sequences of observations.

[1]Hidden Markov Models are also closely related to Probabilistic Automata. Appendix A discusses the connections in detail and shows that with appropriate definitions of equivalence, HMMs can be considered a subclass of probabilistic automata.

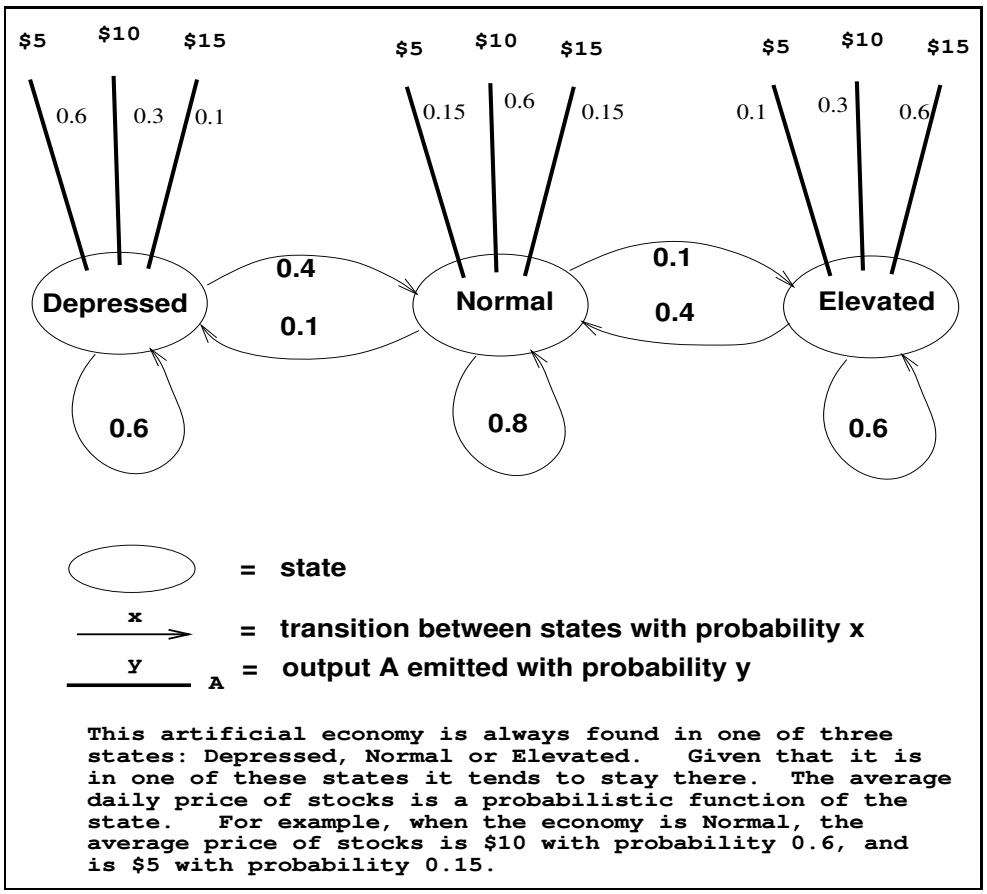


Figure 1-1: A Hidden Markov Model Economy

A given sequence is classified as arising from the source whose HMM model has the highest *a posteriori* likelihood of producing it. Despite their popularity and relative success, HMMs are not well understood on a fundamental level. This thesis attempts to lay part of a foundation for a more principled use of Hidden Markov Models in pattern recognition. In the next section I will briefly describe the history of functions of Markov Chains as relevant to this thesis. I will then proceed to discuss the motivations underlying this research and the major questions that I address here. Principally, these questions will involve the development of fast algorithms for deciding the equivalence of HMMs and reducing them to minimal canonical forms. The chapter will conclude by introducing the basic definitions and notation necessary for

understanding the rest of the thesis.

1.2 Historical Overview

As mentioned in the previous section, Hidden Markov Models are probabilistic functions of Markov Chains, of which the artificial economy in Figure 1.1 is an example. The concept of a function of a Markov Chain is quite old and the questions answered in this thesis seem to have been first posed by Blackwell and Koopmans in 1957 in the context of related *deterministic* functions of Markov Chains.[blackwell57] This work sought to find necessary and sufficient conditions that would "identify" equivalent deterministic functions of Markov Chains, and studied the question in some special cases. Gilbert, in 1959, provided a more general, but still partial, answer to this question of "identifiability" of deterministic functions of Markov Chains.[gilbert59] The topic was studied further by several authors who elucidated various aspects of the problem. ([burke58], [dharma63a], [dharma63b], [dharma68], [bodreau68], [rosenblatt71]) Functions of Markov Chains were also studied under the rubric "Grouped Markov Chains", and necessary and sufficient conditions were established for equivalence of a Grouped Chain to a traditional Markov Chain.([kemeney65], [iosifescu80]) Interest in functions of Markov Chains, and particularly, probabilistic functions of Markov Chains, has been revived recently because of their successful applications in speech recognition. The most effective recognizers in use today employ a network of HMMs as their basic technology for identifying the words in a stream of spoken language.([lee88],[levinson83]) Typically, the HMMs are used as probabilistic source models which are used to compute the posterior probabilities of a word, given a model. This thesis arises from an attempt to build part of a foundation for the principled use of HMMs in pattern recognition applications. We provide a complete characterization of equivalent HMMs and give an algorithm for reducing HMMs to minimal canonical representations. Some work on the subject of equivalent functions of Markov

Chains has been done concurrently with this thesis in Japan.[ito92] However, Ito et al. work with less general deterministic functions of Markov Chains, and find an algorithm for checking equivalent models that takes time exponential in the size of the chain. (In this thesis, we achieve polynomial time algorithms in the context of more general *probabilistic* functions of Markov Chains.) Some work has been done by Y.Kamp on the subject of reduction of states in HMMs.[kamp85] However, Kamp's work only considers the very limited case of reducing pairs of states with identical output distributions, in left-to-right models. There has also been some recent work in the theory of Probabilistic Automata (PA) which uses methods similar to ours to study equivalence of PAs.[tzeng] Tzeng cites the work of Azaria Paz [paz71] and others as achieving the previous best results for testing equivalence of Probabilistic Automata.[2] Appendix A will define Probabilistic Automata and discuss their connections with HMMs. In Chapter 3 we will define *Generalized Markov Models*, a new class of models for stochastic processes that are derived by relaxing the positivity constraint on some of the parameters of HMMs. The idea of defining GMMs arises from work by L.Niles, who studied the relationship between stochastic pattern classifiers and "neural" network schemes.[niles90] Niles demonstrated that relaxing the positivity constraint on HMM parameters had a beneficial effect on the performance of speech classifiers. He proceeded to interpret the negative weights as inhibitory connections in a network formulation of HMMs.

1.3 The Major Questions

Despite their popularity and relative success HMMs, are not well understood on a theoretical level. If we wish to apply these models in a principled manner to

[2] Paz's results placed the problem of deciding equivalence of Probabilistic Automata in the complexity class co-NP. It is well known that equivalence of deterministic automata is in P and equivalence of nondeterministic automata is PSPACE-complete. Tzeng decides equivalence of PAs in polynomial time using methods similar to ours.

Bayesian classification, we should know that HMMs are able to accurately represent the class-conditional stochastic processes appropriate to the classification domain. Unfortunately, we do not understand in detail the class of processes that can be modelled exactly by Hidden Markov Models. Even worse, we do not know how many states an HMM would need in order to approximate a given stochastic process to a given degree of accuracy. We do not even have a good grasp of precisely what characteristics of a stochastic process are difficult to model using HMMs.[3] There is a wide body of empirical knowledge that practitioners of Hidden Markov Modelling have built up, but I feel that the collection of useful heuristics and rules of thumb they represent are not a good foundation for the principled use of HMMs in pattern recognition. This thesis arises from some investigations into the properties of HMMs that are important for their use as pattern recognizers.

1.3.1 Intuitions and Directions

The basic intuition underlying a comparison of the relative expressiveness of Hidden Markov Models and the well-understood Markov Chains suggests that HMMs should be more "powerful" since we can store information concerning the past in probability distributions that are induced over the hidden states. This stored information permits the output of a finite-state HMM to be conditioned on the entire past history of outputs. This is in contrast with a finite-state Markov Chain which can be conditioned only on a finite history. On the other hand, the amount of information about the output of an HMM at time t, given by the output at time $(t-n)$, should drop off with n. It can also be seen that there are many HMMs that are models of exactly the same process, implying that there can be many redundant degrees of freedom in a Hidden Markov Model. This leads to the auxiliary problem of trying to characterize Hidden Markov Models in terms of equivalence classes that are models of precisely

[3] We can, however, reach some conclusions quickly by considering analogous questions for Finite Automata. For example, it should not be possible to build a finite-state HMM that accurately models the long-term statistics of a source that emits pallindromes with high probability.

the same process. Such an endeavour would give some insight into the features of an HMM that contribute to its expressiveness as a model for stochastic processes. Given a characterization in terms of equivalence classes, every HMM could be reduced to a canonical form which would essentially be the smallest member of its class. This is a prerequisite for the problem of characterizing the processes modelled by HMMs, since we should, at the very least, be able to say what makes one model different from another. Furthermore, the canonical representation of an HMM would presumably remove many of the superfluous features of the model that do not contribute to its intrinsic expressiveness. Therefore, we could more easily understand the structure and properties of Hidden Markov Models by studying their canonical representations. In addition, a minimal representation for a stochastic process within the HMM framework is an abstract measure of the complexity of the process. This idea has some interesting connections with Minimum Description Length principles and ideas about Kolmogorov Complexity. However, these connections are not explored in this thesis.

1.3.2 Contributions of This Thesis

Keeping the goals described above in mind, I have developed quick methods to decide equivalence of Hidden Markov Models and reduce them to minimal canonical forms. On the way, I introduce a convenient generalization of Hidden Markov Models that relaxes some of the constraints imposed on HMMs by their probabilistic interpretation. These *Generalized Markov Models* (GMMs), defined in Chapter 3, preserve the essential properties of HMMs that make them convenient pattern classifiers. They arise from the point of view that having a probabilistic interpretation of HMM parameters is peripheral to the goal of designing convenient and parsimonious representations for stochastic processes. The reduction algorithm for Hidden Markov Models will, in fact, reduce HMMs to their minimal equivalent GMMs. Towards the end of the thesis, I will also briefly consider the problem of approximate equivalence of models. This is important because, in any practical situation, HMM parameters are estimated

from data and are subject to statistical variability. I have listed the major results of this thesis below. I have developed:

1. A polynomial time algorithm to check equivalence of prior probability distributions on a given model.
2. A polynomial time algorithm to check equivalence of HMMs with fixed priors.
3. A polynomial time algorithm to check the equivalence of HMMs for arbitrary priors.
4. A definition for a new type of classifier, a Generalized Markov Model (GMM), that is derived by relaxing the positivity constraint on HMM parameters. We will give a detailed description of the relationship between HMMs and GMMs.
5. A polynomial time algorithm to canonicalize a GMM by reducing it to a minimal equivalent model that is essentially unique. The minimal representation, when appropriately restricted, will be a minimal representation of HMMs in the GMM framework. The result will also involve a characterization of the essential degree of expressiveness of a GMM.

We will see that all these results are easy to achieve when cast the language of linear vector spaces. The problems discussed here have remained open for quite a long time because they were not cast in the right language for easy solution.

1.4 Basic Definitions

In this section I will define Hidden Markov Models formally, and I will introduce the basic notation and concepts that will be useful in later chapters.

1.4.1 Hidden Markov Models

Definition 1.1 (Hidden Markov Model)

A Hidden Markov Model can be defined as a quadruple $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$ *where* $s_i \in \mathcal{S}$ *are the* **states** *of the model and* $o_j \in \mathcal{O}$ *are the* **outputs**. *Taking* $s(t)$ *to be the state and* $o(t)$ *the output of* $\mathcal{M}$ *at time* t*, we also define the* **transition matrix** $\mathbf{A}$ *and the* **output matrix** $\mathbf{B}$ *so that* $\mathbf{A}_{ij} = \Pr(s(t) = s_i | s(t-1) = s_j)$ *and* $\mathbf{B}_{ij} = \Pr(o(t) = o_i | s(t) = s_j)$. *In this thesis we only consider HMMs with discrete and finite state and output sets. So, for future use we also let* $n = |\mathcal{S}|$ *and* $k = |\mathcal{O}|$.

In order for an HMM to model a stochastic process, it must be *initialized* by specifying an initial probability distribution over states. The model then transitions probabilistically between its states based on the parameters of its transition matrix and emits symbols based on the probabilities in its output matrix. Therefore, we define an *Initialized Hidden Markov Model* as follows:

Definition 1.2 (Initialized Hidden Markov Model)

An Initialized Hidden Markov Model is a quintuple $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B}, \vec{p})$. *The symbols* $\mathcal{S}$, $\mathcal{O}$, $\mathbf{A}$, *and* $\mathbf{B}$ *represent the same quantities as they do in Definition 1.1.* $\vec{p}$ *is probability vector such that* p_i *is the probability that the model starts in state* s_i *at time* $t = 0$. *We take* $\vec{p}$ *to be a column vector. Having fixed the priors, the model may be evolved according to the probabilities encoded in the transition matrix* $\mathbf{A}$ *and the output matrix* $\mathbf{B}$. *If* $\mathcal{N}$ *is a given Hidden Markov Model, we will use the notation* $\mathcal{N}(\vec{p})$ *to denote the HMM* $\mathcal{N}$ *initialized by the prior* $\vec{p}$.

Figure 1.2 shows an example of a Hidden Markov Model as defined above. Our definition is slightly different from the standard definition of HMMs which actually corresponds to our Initialized Hidden Markov Models. In our formulation, an HMM defines a class of stochastic processes corresponding to different settings of the prior probabilites on the states. An Initialized Hidden Markov Model is a specific process derived by fixing a prior on an HMM.

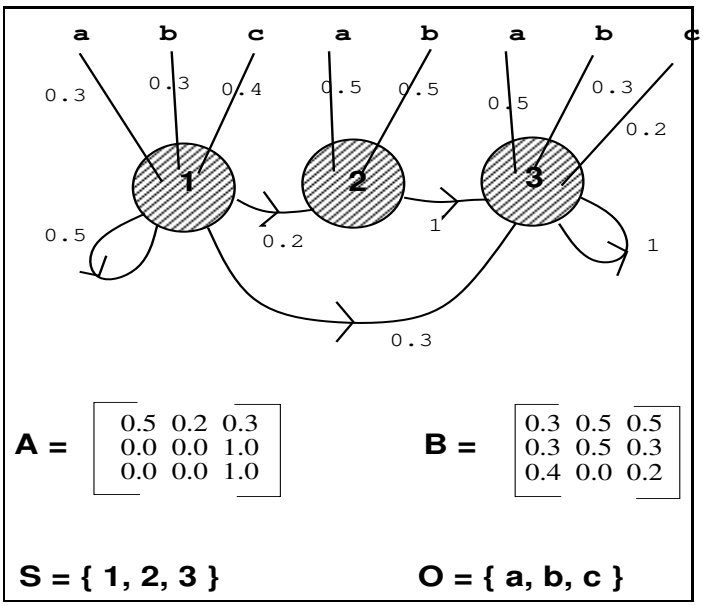


Figure 1-2: A Hidden Markov Model

1.4.2 Variations on The Theme

It should be pointed out that many variants of Hidden Markov Models appear in the literature. Authors have frequently used models in which the outputs are associated with the transitions rather than the states. It can be shown quite easily that it is always possible to convert such a model into an equivalent HMM according to our definition.[4] However, for somewhat technical reasons, converting from a hidden-transition HMM to a hidden-state HMM requires, in general, an increase in the number of states. The literature also frequently uses models with continuously varying observables. These are easily defined by replacing the "ouput matrix" **B** by continuous output densities. HMMs with Gaussian output densities are related to the Radial Basis Functions of [poggio89].[5] Some authors also designate "absorbing states" which, when entered, cause the model to terminate production of a string.

[4] This is analogous to the equivalence of Moore and Mealy Finite State Machines

[5] Suppose $\mathcal{M}$ is a Hidden Markov Model with states $\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$ and Gaussian output distributions $\{G_{s_1}, G_{s_2}, \cdots, G_{s_n}\}$ associated with the states. Also let $x = (o(1), o(2), \cdots, o(t))$ is an

The analysis of such absorbing models is somewhat different from that of the HMMs in Definition 1.1 for uninteresting technical reasons. For the substantive problems of pattern recognition an absorbing model can always be "simulated" in our formulation by creating a state which emits a single special output symbol and loops with probability 1 onto itself.

1.4.3 Induced Probability Distributions

As described in the definition of Initialized HMMs, a stochastic process can be modelled using Hidden Markov techniques by constructing an appropriate HMM, initializing it by specifying a prior probability distribution on the states, and then evolving the model according to its parameters. This evolution then produces output strings whose statistics define a stochastic process over the output set of the model. In recognition applications we are usually interested in the probability that a given observation string was produced by a source whose model is a given HMM. We quantify this by defining the probability distribution over strings induced by an Initialized Hidden Markov Model:

Definition 1.3 (Induced Probability Distributions)
Suppose we are given an HMM $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$ *and a prior distribution* $\vec{p}$*. Borrow the standard notation of the theory of regular languages, and let* $\mathcal{O}^*$ *denote the set of all finite length strings that can be formed by concatenating symbols in* $\mathcal{O}$ *together. We then define the probability that a given string* $x \in \mathcal{O}^*$ *is produced by* $\mathcal{M}(\vec{p})$ *as*

---

output string of length t, Then we can use Equation 1.1 to write:

$$\begin{aligned}\Pr(x|\mathcal{M},\vec{p}) &= \sum_{s(1),\cdots,s(t)} \Pr(s(1),\cdots,s(t)|\mathcal{M},\vec{p})\ \Pr(x|s(1),\cdots,s(t)) \\ &= \sum_{s(1),\cdots,s(t)} \Pr(s(1),\cdots,s(t)|\mathcal{M},\vec{p})\ G_{s(1)}[o(1)]\cdots G_{s(t)}[o(t)]\end{aligned}$$

Each of the products of Gaussians in the second equation defines a "center" for a Radial Basis Function. The sum over states then evaluates a weighted sum over the activations of the various "centers" which are produced as appropriate permutations of the G_{s_i}

follows. Let $m = |x|$ *and let* $s_1, s_2 \cdots s_m \in \mathcal{S}$. *Then:*

$$\Pr(x|\mathcal{M}, \vec{p}) \equiv \Pr(x|\mathcal{M}(\vec{p}), |x|) = \sum_{s_1, s_2, \cdots s_m} \Pr(s_1, \cdots s_m|\mathcal{M}(\vec{p})) \Pr(x|s_1, \cdots s_m) \quad (1.1)$$

Essentially, given a model $\mathcal{M}$, *the probability of a string* x *of length* m *is the likelihood that the model will emit the string* x *while traversing any length* m *sequence of states. Because the definition conditions the probability on the length of the string,* $\Pr(x|\mathcal{M}, \vec{p})$ *defines a probability distribution over strings* x *of length* m *for each postive integer* m. *We let* ϵ *represent the null string and set* $\Pr(\epsilon|\mathcal{M}, \vec{p}) = 1$.

The probability distributions defined above specify the statistical properties of the stochastic process for which the HMM initialized by $\vec{p}$ is a source model. Typical pattern recognition applications evaluate this "posterior probability" of an observation sequence given each of a collection of models and classify according to the model with the highest likelihood.

So an HMM defines a class of stochastic processes - each process corresponding to a different choice of initial distribution on the states. This immediately raises the question of testing whether two prior distributions on a given model induce identical processes. In Chapter 2 we will see that there is an efficient algorithm for deciding this question. But first, in the next section, we will introduce some notation and techniques that show how to use the basic definitions to calculate the quantities of interest to us.

1.5 How To Calculate With HMMS

The basic quantity we are interested in calculating is the probability a given string will be produced by a given model. We will see later that for purposes of determining the equivalence of models and reducing them to canonical forms it also useful to compute various probability distributions over the states and the outputs. In this section we

will introduce some notation that will enable us to mechanize the computation of these quantities so that later analysis becomes easy. The notation and details may become tedious and confusing and so the reader may wish to skim the section, referring back to it as necessary.

Definition 1.4 (State and Output Distributions)
Let $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$ be an HMM with a prior $\vec{p}$, n states and k outputs. Let $s(t)$ and $o(t)$ be, respectively, the state and output at time t. Let $\vec{k}(t)$ be an n-dimensional column vector such that $k_i(t) = \Pr(s(t) = s_i, o(1), o(2), \cdots, o(t-1)|\mathcal{M}, \vec{p})$. In other words, $k_i(t)$ is the joint probability of being in s_i at time t and seeing all the **previous** *outputs. We define $m_i(t)$ to be the probability of being in state s_i after* **also** *seeing the output at time t: $m_i(t) = \Pr(s(t) = s_i, o(1), \cdots, o(t-1), o(t)|\mathcal{M}, \vec{p})$. Finally, let $\vec{l}(t)$ be a column vector describing the probabilities of the various outputs at time t: $l_i(t) = \Pr(o(t) = o_i, o(1), o(2), \cdots o(t-1)|\mathcal{M}, \vec{p})$. From the defintion of the $\mathbf{B}$ matrix, we can write this as: $\vec{l}(t) = \mathbf{B}\vec{k}(t)$.*

In order to determine equivalence of HMMs and reduce them to canonical forms we will need to be able to reason conveniently about the temporal evolution of the model. Using Definition 1.4 we can write that $\vec{k}(t+1) = \mathbf{A}\vec{m}(t)$. Furthermore, if $o(t) = o_j$ we can factor the definition of $\vec{m}(t)$ to write:

$$\begin{aligned} m_i(t) &= \Pr(o(t) = o_j | s(t) = s_i, \mathcal{M}, \vec{p}, o(1), \cdots, o(t-1)) \Pr(s(t) = s_i, o(1), \cdots, o(t-1)|\mathcal{M}, \vec{p}) \\ &= \Pr(o(t) = o_j | s(t) = s_i) k_i(t) \\ &= \mathbf{B}_{ji} k_i(t) \end{aligned} \tag{1.2}$$

In order to write Equation 1.2 more compactly, we introduce the following notion of a *projection operator*:

Definition 1.5 (Projection Operators)
Suppose an HMM $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$ has k outputs. We define a set of **projection**

operators $\{\mathbf{B}_1, \mathbf{B}_2, \cdots \mathbf{B}_k\}$ *so that* $\mathbf{B}_i = Diag[ith\ row\ of\ \mathbf{B}]$. *In other words* $\mathbf{B}_i$ *is a diagonal matrix whose diagonal elements are the row in* $\mathbf{B}$ *corresponding to the output symbol* o_i. *Sometimes we will use the notation* $\mathbf{B}_o$ *to mean the projection operator corresponding to the the output* o. *(i.e.* $\mathbf{B}_o = \sum_{o_i \in \mathcal{O}} \delta(o, o_i)\mathbf{B}_i$ *where* $\delta(a, b)$ *is 1 if* $a = b$ *and 0 otherwise.) Suppose* $\vec{v}$ *is a vector whose dimension equals the number of states of the model. Then multiplying* $\vec{v}$ *by* $\mathbf{B}_i$ *weights each component of* $\vec{v}$ *by the probability that the corresponding state would emit the output* o_i.

We can use the projection operator notation to compactly write Equation 1.2 as $\vec{m}(t) = \mathbf{B}_{o(t)}\vec{k}(t)$. Now we can write $\vec{k}(t+1) = \mathbf{A}\mathbf{B}_{o(t)}\vec{k}(t)$ and $\vec{m}(t+1) = \mathbf{B}_{o(t+1)}\mathbf{A}\vec{m}(t)$. In order to summarize this we introduce a set of definitions for the *transition operators* of a Hidden Markov Model.

Definition 1.6 (Transition Operators)

Given an HMM $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$ *with* n *states we define the model* **transition operators** *as follows. Let* ϵ *be the null string. Define* $\mathbf{T}(\epsilon) = \mathbf{I}$ *where* $\mathbf{I}$ *is the* $n \times n$ *identity matrix. Also, for every* $o_i \in \mathcal{O}$ *define* $\mathbf{T}(o_k) = \mathbf{A}\mathbf{B}_{o_k}$. *We can see that* $\mathbf{T}(o_k)_{ij}$ *is the probability of emitting* o_k *in state* s_j *and then entering* s_i. *We extend these to be transition operators on* $\mathcal{O}^*$ *as follows. For any output string* $x = (o_1, o_2 \cdots o_t) \in \mathcal{O}^*$ *let:*

$$\mathbf{T}(x) = \mathbf{T}(o_1, \cdots o_t) = \mathbf{T}(o_t)\mathbf{T}(o_{t-1}) \cdots \mathbf{T}(o_1) \tag{1.3}$$

We can interpret these extended transition operators by noticing that $\mathbf{T}(x)_{ij}$ *is the probability of starting in state* s_j, *emitting the string* x, *and then entering state* s_i.

Using the transition operators of Definition 1.6 we can coveniently write all the quantities we wish to compute. Suppose $\mathcal{M}$ is an HMM with n states, k outputs and prior $\vec{p}$. Take x_t to be the output string $(o_1, o_2 \cdots o_t)$ and $\vec{1}$ to be an n-dimensional vector all of whose entries are 1. Also let x_{t-1} be the $t-1$ long prefix of x_t. Then we can

see that:

$$\vec{m}(t) = \mathbf{B}_{o_t}\mathbf{T}(x_{t-1})\vec{p} \tag{1.4}$$

$$\vec{k}(t+1) = \mathbf{A}\vec{m}(t) = \mathbf{T}(x_t)\vec{p} \tag{1.5}$$

$$\vec{l}(t+1) = \mathbf{B}\vec{k}(t+1) = \mathbf{B}\mathbf{T}(x_t)\vec{p} \tag{1.6}$$

$$\Pr(x_t|\mathcal{M},\vec{p}) = \vec{1}\cdot(\mathbf{T}(x_t)\vec{p}) \tag{1.7}$$

The reader may wish to verify some of these equations from the definitions to ensure his or her facility with the notation.

1.6 Roadmap

This chapter has developed the background necessary for understanding the results in this thesis. The basic definitions and notation given here are summarized in Table 1.1. Chapter 2 discusses the algorithms related to equivalence of Hidden Markov Models. Chapter 3 defines Generalized Markov Models and describes the algorithm for reducing HMMs to minimal canonical forms. Chapter 3 also contains a fundamental characterization of the essential expressiveness of a Hidden Markov Model. Chapter 4 presents some preliminary ideas concerning several topics including approximate equivalence and potential practical applications of the results of this thesis. Finally, Appendix A shows how HMMs, in the formulation of this paper, are related to Probabilistic Automata.

Given: an HMM $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$ with n states, k outputs and prior $\vec{p}$.

Definitions:

1. $\Pr(x|\mathcal{M}, \vec{p}) \equiv \Pr(x|\mathcal{M}(\vec{p}), |x|) = \sum_{s_1, s_2, \cdots s_m} \Pr(s_1, \cdots s_m|\mathcal{M}(\vec{p})) \Pr(x|s_1, \cdots s_m)$
2. $\Pr(\epsilon|\mathcal{M}, \vec{p}) = 1$ where ϵ is the null string
3. $\vec{k}(t)$ is an n-dimensional vector such that $k_i(t) = \Pr(s(t) = s_i, o(1), o(2), \cdots, o(t-1)|\mathcal{M}, \vec{p})$
4. $\vec{m}(t)$ is an n-dimensional vector such that $m_i(t) = \Pr(s(t) = s_i o(1), \cdots o(t-1), o(t)|\mathcal{M}, \vec{p})$
5. $\vec{l}(t)$ is a k-dimensional vector such that $l_i(t) = \Pr(o(t) = o_i o(1), o(2) \cdots o(t-1)|\mathcal{M}, \vec{p})$
6. The **projection operators** $\{\mathbf{B}_1, \mathbf{B}_2, \cdots \mathbf{B}_k\}$ are defined as $\mathbf{B}_i = Diag[ith\ row\ of\ \mathbf{B}]$. Also if $o \in \mathcal{O}$ then we write $\mathbf{B}_o$ to denote the projection operator corresponding to output o.
7. We define **transition operators** so that:
$$\begin{aligned} \mathbf{T}(\epsilon) &= \mathbf{I} \\ \mathbf{T}(o_k) &= \mathbf{A}\mathbf{B}_k, \\ \mathbf{T}(o(1), o(2), \cdots o(t)) &= \mathbf{T}(o(t)) \cdots \mathbf{T}(o(2))\mathbf{T}(o(1)) \end{aligned}$$

Model Evolution:

1. Suppose the HMM emits the output $x_t = [o(1), o(2), \cdots o(t)]$. Also use the notation x_{t-1} to mean the $t-1$ long prefix of x_t, and the symbol $\vec{1}$ to mean the n-dimensional vector all of whose entries at 1. Then we can write:
   - $\vec{m}(t) = \mathbf{B}_{o(t)}\mathbf{T}(x_{t-1})\vec{p}$
   - $\vec{k}(t+1) = \mathbf{A}\vec{m}(t) = \mathbf{T}(x_t)\vec{p}$
   - $\vec{l}(t+1) = \mathbf{B}\vec{k}(t+1) = \mathbf{B}\mathbf{T}(x_t)\vec{p}$
   - $\Pr(x_t|\mathcal{M}, \vec{p}) = \vec{1} \cdot (\mathbf{T}(x_t)\vec{p})$

Table 1.1: Summary of Important Notations

Chapter 2

Equivalence of HMMs

As discussed in the previous chapter, many different Hidden Markov Models can represent the same stochastic process. Prior to addressing questions about the expressive power of HMMs, it is important to understand exactly when two models $\mathcal{M}$ and $\mathcal{N}$ are equivalent in the sense that they represent the same statistics. In Section 2.2 we will see how to determine when two prior distributions on a given HMM induce identical stochastic processes. Section 2.3 discusses equivalence of Initialized Hidden Markov Models. Section 2.4 shows how to determine whether two HMMs are representations for the same class of stochastic processes. This will lead, in the next chapter, to a fundamental characterization of the degree of freedom available in a given model. This characterization will be used to reduce HMMs to minimal canonical forms.

2.1 Definitions

We begin by defining what we mean by equivalence of Hidden Markov Models. First of all, we should say what it means for two stochastic processes to be equivalent.

Definition 2.1 (Equivalence of Stochastic Processes)
Suppose $\mathcal{X}$ and $\mathcal{Y}$ are two stochastic processes on the same discrete alphabet $\mathcal{O}$. For each $x \in \mathcal{O}^*$ let $\Pr_{\mathcal{X}}(x)$ be the probability that after $|x|$ steps the process $\mathcal{X}$ has emitted

the string x. Define $\Pr_{\mathcal{Y}}(x)$ similarly. Then we say that $\mathcal{X}$ and $\mathcal{Y}$ are **equivalent processes** *($\mathcal{X} \Leftrightarrow \mathcal{Y}$) if and only if $\Pr_{\mathcal{X}}(x) = \Pr_{\mathcal{Y}}(x)$ for every $x \in \mathcal{O}^*$*

In Chapter 1 we discussed the interpretation of an Initialized Hidden Markov Model (IHMM) as a finite-state representation for a stochastic process, and we defined the probability distribution over strings induced by the process. We can use these definitions to say what we mean by equivalence of Initialized HMMs.

Definition 2.2 (Equivalence of Initialized HMMs)
Let $\mathcal{M}$ and $\mathcal{N}$ be two Hidden Markov Models with the same output set, and initialized by priors $\vec{p}$ and $\vec{q}$ respectively. We wish to say that these initialized models are equivalent if they represent the same stochastic process. So we say that $\mathcal{M}(\vec{p})$ is equivalent to $\mathcal{N}(\vec{q})$ ($\mathcal{M}(\vec{p}) \Leftrightarrow \mathcal{N}(\vec{q})$) if and only if $\Pr(x|\mathcal{M}, \vec{p}) = \Pr(x|\mathcal{N}, \vec{q})$ for every $x \in \mathcal{O}^*$. This is the same as saying that $\mathcal{M}(\vec{p}) \Leftrightarrow \mathcal{N}(\vec{q})$ exactly when, for every time t, the joint probability of the output with the entire previous output sequence, is the same for both models. In the notation of Chapter 1 we can write this as: $\mathbf{B}_{\mathcal{M}}\mathbf{T}_{\mathcal{M}}(x)\vec{p} = \mathbf{B}_{\mathcal{N}}\mathbf{T}_{\mathcal{N}}(x)\vec{q}$ for every $x \in \mathcal{O}^* \cup \{\epsilon\}$.

In Chapter 1 we also mentioned that different prior distributions on the same HMM could induce the same stochastic process. In order to identify the conditions under which this can occur we make the following definition.

Definition 2.3 (Equivalence of Prior Distributions)
Let $\vec{p}$ and $\vec{q}$ be two different prior distributions on an HMM $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$. We say that $\vec{p}$ and $\vec{q}$ are equivalent priors for $\mathcal{M}$ ($\vec{p} \overset{\mathcal{M}}{=} \vec{q}$) if and only if $\mathcal{M}(\vec{p}) \Leftrightarrow \mathcal{M}(\vec{q})$ i.e., if and only if the Initialized HMMs derived by fixing the priors on $\mathcal{M}$ are equivalent.

We are now ready to define equivalence of Hidden Markov Models. As discussed in Chaper 1, HMMs can be treated as finite state representations for classes of stochastic processes. We would like to say that two HMMs are equivalent if they represent the same class of processes.

Definition 2.4 (Equivalence and Subset Relations for HMMs)
Let $\mathcal{M}$ and $\mathcal{N}$ be two HMMs with the same output set. Let $\vec{p}$ and $\vec{q}$ denote prior distributions on $\mathcal{M}$ and $\mathcal{N}$ repsectively. We say that $\mathcal{N}$ is a **subset** *of $\mathcal{M}$ ($\mathcal{N} \subseteq \mathcal{M}$) if and only if for each $\vec{q}$ on $\mathcal{N}$ we can find a corresponding $\vec{p}$ on $\mathcal{M}$ such that $\mathcal{M}(\vec{p}) \Leftrightarrow \mathcal{N}(\vec{q})$. In other words, $\mathcal{N}$ is a subset of $\mathcal{M}$ if and only if the class of processes represented by $\mathcal{N}$ is a subset of the class of processes represented by $\mathcal{M}$. We can then write $\mathcal{M}$ is equivalent to $\mathcal{N}$ ($\mathcal{M} \Leftrightarrow \mathcal{N}$) exactly when $\mathcal{N} \subseteq \mathcal{M}$ and $\mathcal{M} \subseteq \mathcal{N}$.*

The basic intuition underlying all the results concerning the equivalence of HMMs is the following: The output distributions of an HMM are linear transformations that map an underlying dynamics on the states onto a dynamics on the space of observations. Heuristically, it must be the case that the components of the dynamics on the states that fall in the null-space of the output matrix must represent degrees of freedom that are irrelevant to the statistics on the outputs. So, for example, we will see that two prior distributions on a model are equivalent if and only if their difference falls in a particular subspace of null-space of the output matrix. All the algorithms discussed in this chapter will achieve their goals by rapidly checking properties of various vector spaces associated with HMMs.

2.2 Equivalence of Priors

When do two prior distributions on a given model induce the same stochastic process? This is the most basic question that we would like to answer. Using the notation developed in Chapter 1, and the definition of equivalent Initialized HMMs, we can write the condition for equivalent priors as follows: $\vec{p} \stackrel{\mathcal{M}}{=} \vec{q}$ if and only if $\mathbf{BT}(x)\vec{p} = \mathbf{BT}(x)\vec{q}$ for every $x \in \mathcal{O}^* \cup \{\epsilon\}$. Let $\vec{\delta} = \vec{p} - \vec{q}$. Then we can rephrase this as: $\mathbf{BT}(x)\,[\vec{p} - \vec{q}] = \mathbf{BT}(x)\vec{\delta} = 0$ for every $x \in \mathcal{O}^* \cup \{\epsilon\}$. In other words $\vec{p} \stackrel{\mathcal{M}}{=} \vec{q}$ if and only if for every string $x \in \mathcal{O}^* \cup \{\epsilon\}$ we can say that $\mathbf{T}(x)\vec{\delta}$ is a vector that falls in the null-space of the output matrix $\mathbf{B}$. This can be expressed in more geometrical

terms as follows.

Theorem 2.1 *Equivalence of Priors (Geometrical Interpretation)* [1]

Suppose $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$ *is a Hidden Markov Model with* n *states and* k *outputs. Let* $\vec{p}$ *and* $\vec{q}$ *be two prior distributions on* $\mathcal{M}$ *with* $\vec{\delta} = \vec{p} - \vec{q}$*. Let* $\mathcal{N}$ *denote the null-space of the linear transformation* $\mathbf{B}$ *and let* $\mathcal{I}$ *be the largest subspace of* $\mathcal{N}$ *that is invariant under each of the transformation operators* $\mathbf{T}(o_i)$*. Then* $\vec{p} \stackrel{\mathcal{M}}{=} \vec{q}$ *if and only if* $\vec{\delta} \in \mathcal{I}$*.*

Proof: First of all suppose $\vec{\delta} \in \mathcal{I} \subseteq \mathcal{N}$. Then because $\mathcal{I}$ is invariant under all the $\mathbf{T}(o_i)$ we know that $\mathbf{T}(o_i)\vec{\delta} \in \mathcal{I}$ and, by induction, we can say that for every $x = [o(1), o(2), \cdots, o(t)] \in \mathcal{O}^*$ it is true that $\mathbf{T}(x)\vec{\delta} = \mathbf{T}(o_t) \cdots \mathbf{T}(o_1)\vec{\delta} \in \mathcal{I}$. We conclude that $\mathbf{T}(x)\vec{\delta} \in \mathcal{N}$ for every $x \in \mathcal{O}^* \cup \{\epsilon\}$. Therefore, by our earlier discussion, $\vec{p}$ is equivalent to $\vec{q}$. This proves the sufficiency of our condition for equivalence. Next we prove necessity. Suppose that $\vec{p} \stackrel{\mathcal{M}}{=} \vec{q}$. Then let $\mathcal{D} = \left\{\vec{\delta}(x) : \vec{\delta}(x) = \mathbf{T}(x)\vec{\delta},\ x \in \mathcal{O}^* \cup \{\epsilon\}\right\}$ be the set of all differences between $\mathbf{T}(x)\vec{p}$ and $\mathbf{T}(x)\vec{q}$ for every string x. If $\vec{\delta}(x)$ is any vector in $\mathcal{D}$ and $\mathbf{T}(o_i)$ is any transition operator, then $\mathbf{T}(o_i)\vec{\delta}(x)$ is also in $\mathcal{D}$. So $\mathcal{D}$ is invariant under the action of the every transition operator and, therefore, so is $Span(\mathcal{D})$. By assumption of equivalence of priors, every vector in $\mathcal{D}$ lies in the null-space of $\mathbf{B}$. So $Span(\mathcal{D}) \subseteq \mathcal{N}$. We conclude that $Span(\mathcal{D})$ is a subspace of the largest subspace of $\mathcal{N}$ that is invariant under all the transition operators. This proves the necessity of our condition for equivalence. □

In effect, the difference between equivalent priors is a vector that lies in a subspace that contributes nothing to the probability distribution over outputs, and remains in this subspace as the model evolves. It is not enough that $\vec{\delta}$ simply be in the null-space

[1] We remind the reader of the following linear algebraic notions. The *null-space* of a linear transformation $\mathbf{B}$ from $\mathbf{R}^n$ to $\mathbf{R}^k$ is the subspace of $\mathbf{R}^n$ that is mapped by $\mathbf{B}$ into the k-dimensional zero vector. An *invariant subspace* of a linear transformation $\mathbf{T}$ from $\mathbf{R}^n$ to $\mathbf{R}^n$ is a subspace $\mathbf{V}$ such that $\mathbf{T}$ maps every vector in $\mathbf{V}$ into $\mathbf{V}$.

of $\mathbf{B}$ because some of the vectors in the null-space may contribute to the dynamics of the system and affect later distributions over outputs. The fact that $\vec{\delta}$ lies in an invariant subspace of the null-space guarantees that $\vec{\delta}$ will *never* contribute to the distribution over outputs, even after the model evolves. Figure 2.1 shows a simple example in which *all the states have the same output distribution*, so that the null-space of $\mathbf{B}$ consists of all vectors that sum to zero. Furthermore, for every i, $\mathbf{B}_i$ is proportional to the identity so that $\mathbf{T}(o_i)$ is proportional to $\mathbf{A}$. Since $\mathbf{A}$ is stochastic it preserves sums and so we see that the space of vectors which sum to zero is an invariant subspace of every $\mathbf{T}(o_i)$. For *any* priors $\vec{p}$ and $\vec{q}$ we know that $\vec{\delta} = \vec{p} - \vec{q}$ sums to zero since $\vec{p}$ and $\vec{q}$ are both stochastic. So, as we would expect for this degenerate case, Theorem 2.1 tells us that *all* prior distributions on the model induce equivalent stochastic processes.

Although Theorem 2.1 gives a good understanding of *why* two priors may be equivalent for a model, it is not in a form that is immediately useful for developing a quick algorithm. So we prove another form of the theorem that will be used directly in the algorithm of Figure 2.2

Theorem 2.2 *Equivalence of Priors*

Let $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$ be a Hidden Markov Model. Suppose $\vec{p}$ and $\vec{q}$ are two prior distributions on $\mathcal{M}$ with $\vec{\delta} = \vec{p} - \vec{q}$. Define $\mathcal{D} = \left\{\vec{\delta}(x) : \vec{\delta}(x) = \mathbf{T}(x)\,\vec{\delta},\ x \in \mathcal{O}^* \cup \{\epsilon\}\right\}$, and let $\mathcal{V}$ be any collection of vectors in $\mathcal{D}$ that forms a basis for the vector space spanned by the elements of $\mathcal{D}$. Then $\vec{p} \overset{\mathcal{M}}{=} \vec{q}$ if and only every vector in $\mathcal{V}$ lies in the null-space of $\mathbf{B}$.

Proof: First suppose that $\vec{p} \overset{\mathcal{M}}{=} \vec{q}$. Then $\mathcal{V} \subset \mathcal{D}$ and so, from the previous discussion, every vector in $\mathcal{V}$ must fall in the null-space of $\mathbf{B}$, proving the necessity of the theorem. Now suppose that $\mathbf{B}\vec{v}_j = 0$ for every vector $\vec{v}_j \in \mathcal{V}$. Then, since $\mathcal{V}$ is a basis for the span of $\mathcal{D}$, for every $\vec{\delta}_i \in \mathcal{D}$ there exists a collection of coefficients $\{c_{ij}\}$ such that $\vec{\delta}_i = \sum_{j=1}^{|\mathcal{V}|} c_{ij}\vec{v}_j$. So, for every $\vec{\delta}_i$ we can write $\mathbf{B}\vec{\delta}_i = \mathbf{B}\sum_{j=1}^{|\mathcal{V}|} c_{ij}\vec{v}_j = \sum_{j=1}^{|\mathcal{V}|} c_{ij}\,(\mathbf{B}\vec{v}_j) = 0$.

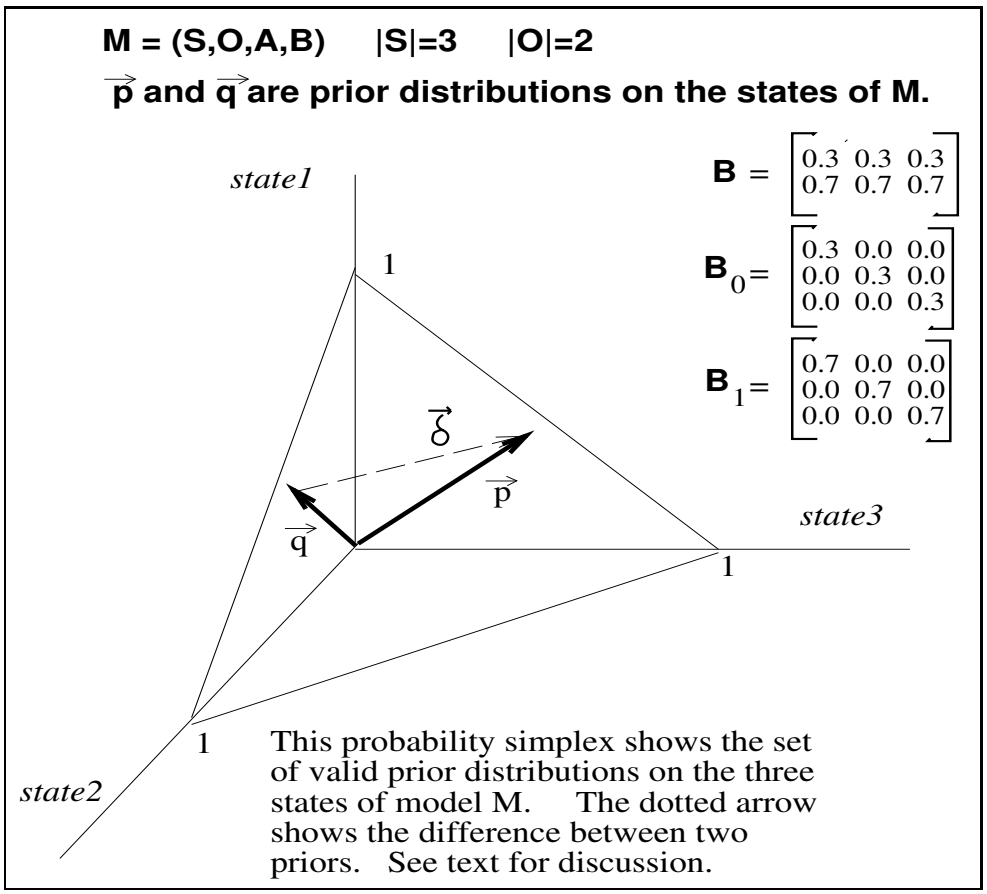


Figure 2-1: Geometrical Interpretation of Equivalence of Priors

This is the same as saying that $\mathbf{BT}(x)\vec{\delta} = 0$ for every $x \in \mathcal{O}^* \cup \{\epsilon\}$. Consequently, we have the desired result that $\vec{p} \stackrel{\mathcal{M}}{=} \vec{q}$. □

Theorem 2.2 provides a necessary and sufficient condition for equivalence of priors on a Hidden Markov Model. We can use it to construct an algorithm by quickly generating the basis $\mathcal{V}$ of the theorem and checking that the elements of the basis fall in the null-space of $\mathbf{B}$. The algorithm in Figure 2.2 does exactly this.[2] We will now

[2]Our procedure for checking equivalence of priors can be optimized in various ways. One such optimization will be presented in the analysis of the running time of the algorithm. We present the algorithm of Figure 2.2 because it is easier to explain.

argue that the algorithm is correct and proceed to calculate its running time.

Given: An HMM $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$
where $|\mathcal{S}| = n$, $|\mathcal{O}| = k$
And priors $\vec{p}$ and $\vec{q}$ on $\mathcal{M}$

1. $\mathbf{V} = \{\ \}$
2. **Queue**$= \{\ \vec{\delta}\}$

\# Step 1: *Find a Basis*
3. Until ($|$**Queue**$| = 0$) or ($|\mathbf{V}| = n$) do
4. Let $\vec{f}$ = first element in **Queue**
5. Remove $\vec{f}$ from **Queue**
6. If $\vec{f} \notin Span(\mathbf{V})$ Then
7. Add $\vec{f}$ to $\mathbf{V}$
8. For each $o_i \in \mathcal{O}$ do
9. Add $\mathbf{T}(o_i)\vec{f}$ to **Queue**

\# Step 2: *Test the basis*
10. For each $\vec{v} \in \mathbf{V}$ do
11. If $\mathbf{B}\vec{v} \neq 0$ Then Return(NOT-EQUIVALENT)

12. Return(EQUIVALENT)

Figure 2-2: Algorithm for Detecting Equivalence of Priors

Correctness: The algorithm of Figure 2.2 proceeds in two steps. In Step 1 it finds a basis $\mathbf{V}$ and, in Step 2, it checks the necessary and sufficient condition for equivalence given in Theorem 2.2. So, it checks equivalence of priors correctly if $\mathbf{V}$ is indeed a basis for the span of $\mathcal{D} = \left\{\vec{\delta}(x) : \vec{\delta}(x) = \mathbf{T}(x)\,\vec{\delta},\ x \in \mathcal{O}^* \cup \{\epsilon\}\right\}$. In order to analyze the algorithm we will use the terminology that the vector $\mathbf{T}(o_i)\vec{v}$ is a *child* of the vector $\vec{v}$. When the basis finding step of the algorithm terminates, $\mathbf{V}$ contains a linearly independent collection of vectors. If the step terminated because $|\mathbf{V}| = n$, we must have a basis for $Span(\mathcal{D})$ since the vectors in $\mathcal{D}$ are n-dimensional. Suppose

now that the basis finding step terminated because **Queue** was empty. Each child of each of the vectors in $\mathbf{V}$ was added to **Queue** by line 9. So each of these children is either in $\mathbf{V}$ or was found to be a linear combination of a set of vectors in $\mathbf{V}$. Let $\mathcal{C}$ denote the set of children of elements of $\mathbf{V}$ that are not themselves in $\mathbf{V}$. Then we can write $\vec{c}_i = \sum_{\vec{v}_j \in \mathbf{V}} d_{ij}\vec{v}_j$ for every $\vec{c}_i \in \mathcal{C}$. Suppose now $\vec{v} \in \mathcal{D}$ is not in $\mathbf{V}$ and is not a child of a vector in $\mathbf{V}$. By construction of the algorithm we can find some string x and some $\vec{c}_i$ which is a child of a vector in $\mathbf{V}$ such that $\mathbf{T}(x)\vec{c}_i = \vec{v}$. We wish to show that every such $\vec{v}$ is in the span of $\mathbf{V}$. We will do this by induction on the length of the string x. If $|x| = 1$ so that $x = o_k \in \mathcal{O}$, then for some $\vec{c}_i$ we know that $\vec{v} = \mathbf{T}(o_k)\vec{c}_i = \mathbf{T}(o_k)\sum_{\vec{v}_j \in \mathbf{V}} d_{ij}\vec{v}_j = \sum_{\vec{v}_j \in \mathbf{V}} d_{ij}\mathbf{T}(o_k)\vec{v}_j$. So we see that $\vec{v}$ is a linear combination of children of elements of $\mathbf{V}$, which all necessarily fall in the span of $\mathbf{V}$. Hence $\vec{v}$ falls in the span of $\mathbf{V}$ if $\vec{v} = \mathbf{T}(x)\vec{c}_i$ for any x of length one and any $\vec{c}_i \in \mathcal{C}$. Now assume that for every x such that $|x| \leq t$ we know that $\vec{v} = \mathbf{T}(x)\vec{c}_i$ is in the span of $\mathbf{V}$. So we write that $\vec{v} = \sum_{\vec{v}_j \in \mathbf{V}} d_{vj}\vec{v}j$. Then for every string $y = xo_k$ of length $t+1$ we know that there is a $\vec{c}_j$ such that $\vec{u} = \mathbf{T}(y)\vec{c}_j = \mathbf{T}(o_k)\mathbf{T}(x)\vec{c}_i = \mathbf{T}(o_k)\vec{v} = \mathbf{T}(o_k)\sum_{\vec{v}_j \in \mathbf{V}} d_{vj}\vec{v}j$. Taking the multiplication by $\mathbf{T}(o_k)$ into the sum we see that $\vec{u}$ is a linear combination of vectors in $\mathbf{V}$ and their children, all of which fall in $Span(\mathbf{V})$. So $\vec{u} \in Span(\mathbf{V})$ also. By induction on $t = |x|$, all $\vec{v} \in \mathcal{D}$ are in the span of $\mathbf{V}$. Therefore, as claimed, $\mathbf{V}$ is a basis for the span of the vectors in $\mathcal{D}$. The second step of the algorithm then evaluates the necessary and sufficient condition of Theorem 2.2 on the basis generated in the first step. Therefore, our algorithm is correct. □

Running Time: We will now compute the worst case running time of the equivalent priors algorithm asuming unit cost arithmetic operations. Once the basis $\mathbf{V}$ is generated in Step 1, the check performed in Step 2 takes $O(n^2k)$ time since $|\mathbf{V}| \leq n$ and each multiplication by $\mathbf{B}$ takes time $O(nk)$. In addition, it takes $O(n^2k)$ time to generate all the $\mathbf{T}(o_i)$ matrices used in the algorithm from the given $\mathbf{A}$ and $\mathbf{B}$ matrices. To analyze Step 1, we observe that each multiplication of $\vec{f}$ by $\mathbf{T}(o_i)$ in line 9

takes time $O(n^2)$. In the worst case the basis $\mathbf{V}$ generated in Step 1 will contain n elements. For every $\vec{v} \in \mathbf{V}$ and every $o_i \in \mathcal{O}$, line 9 adds all vectors $\mathbf{T}(o_i)\vec{v}$ to **Queue**. So, in all, time $O(n^2 \cdot nk) = O(n^3 k)$ could be spent extending **Queue**. The final contribution to the running time is from the check in line 6 of the algorithm to see if $\vec{f}$ should be added to the partially generated basis. We observe that $\vec{f} \notin Span(\mathbf{V})$ can be tested in time $O(n|\mathbf{V}|^2 + |\mathbf{V}|^3)$ by standard Gaussian elimination.[press90] In the worst case, the first $n-1$ vectors that are tested in line 6 will be added to the basis, and all the remaining $nk - (n-1)$ vectors in **Queue** will have to be tested to find the last basis vector. So, for large k and n, these tests will take time $O(n^3) \cdot O(nk) = O(n^4 k)$. This gives an $O(n^4 k)$ running time for the algorithm. We can do better by being a little more clever about the test in line 6. An optimized algorithm would maintain, in addition to the basis set $\mathbf{V}$, a set $\mathbf{U}$ of orthonormal basis vectors produced by applying the Gram-Schmidt procedure to $\mathbf{V}$. Every time a vector $\vec{f}$ is extracted from **Queue**, it is orthogonalized against the current set $\mathbf{U}$. If the residue of this procedure is the zero vector, $\vec{f}$ is in $Span(\mathbf{U}) = Span(\mathbf{V})$, and so $\vec{f}$ is thrown away.[3] If the residue is non-zero, $\vec{f}$ is added to $\mathbf{V}$ and the residue is added to $\mathbf{U}$. The Gram-Schmidt procedure would take time $O(n|\mathbf{V}|)$ since it just involves projection of $\vec{f}$ onto each of the vectors in $\mathbf{U}$ and $|\mathbf{U}| = |\mathbf{V}|$. Repeating the earlier analysis gives a worst case running time of $O(n^3 k)$ for this optimized algorithm.

The next section uses this result concerning equivalence of priors to develop an algorithm to test equivalence of Initialized Hidden Markov Models.

2.3 Equivalence of Initialized HMMs

In order to develop an algorithm to check equivalence of Initialized Hidden Markov Models we will utilize a popular trick from the theory of Finite Automata. Given two models we will build a new HMM whose properties will enable us to check equivalence

[3] We are using the term "residue" to mean the piece of a vector that is left after removing all components along vectors in a given set.

of the two given models easily. (See Figure 2.3) Suppose $\mathcal{M} = (\mathcal{S}_\mathcal{M}, \mathcal{O}, \mathbf{A}_\mathcal{M}, \mathbf{B}_\mathcal{M})$ and $\mathcal{N} = (\mathcal{S}_\mathcal{N}, \mathcal{O}, \mathbf{A}_\mathcal{N}, \mathbf{B}_\mathcal{N})$ are two HMMs initialized by priors $\vec{p}$ and $\vec{q}$ respectively. Then we construct a new HMM $\mathcal{Q} = (\mathcal{S}_\mathcal{Q}, \mathcal{O}_\mathcal{Q}, \mathbf{A}_\mathcal{Q}, \mathbf{B}_\mathcal{Q})$ where $\mathcal{S}_\mathcal{Q} = \mathcal{S}_\mathcal{M} \cup \mathcal{S}_\mathcal{N}$ and $\mathcal{O}_\mathcal{Q} = \mathcal{O}$. If $\mathcal{M}$ has m states, $\mathcal{N}$ has n states and $|\mathcal{O}| = k$ we define:

$$\mathbf{A}_\mathcal{Q} = \begin{bmatrix} \mathbf{A}_\mathcal{M} & 0_{m\times n} \\ 0_{n\times m} & \mathbf{A}_\mathcal{N} \end{bmatrix} \tag{2.1}$$

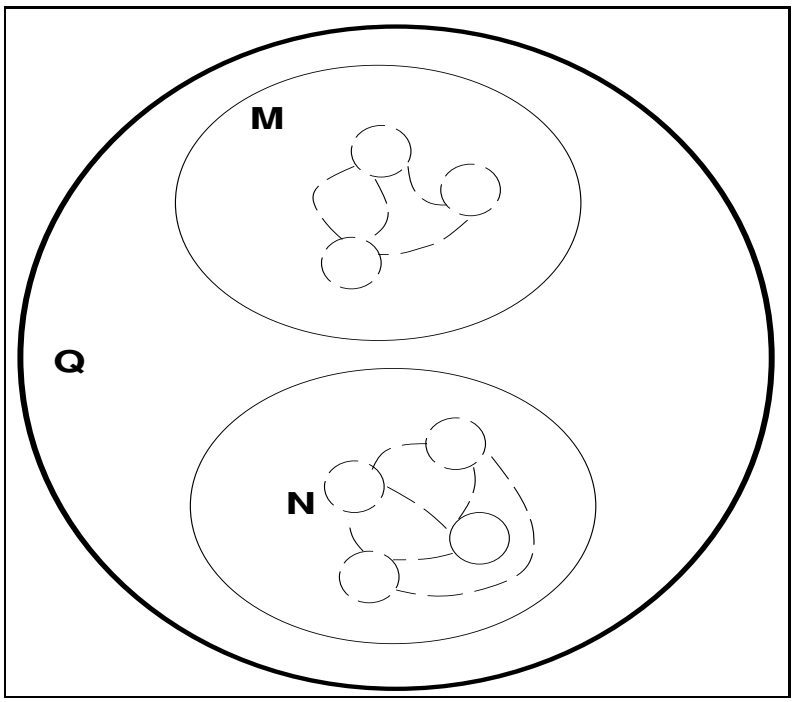
$$\mathbf{B}_\mathcal{Q} = \begin{bmatrix} \mathbf{B}_\mathcal{M} & \mathbf{B}_\mathcal{N} \end{bmatrix} \tag{2.2}$$

(We are using the notation $0_{i\times i}$ for the i by i matrix whose entries are all zero.) Essentially, $\mathcal{Q}$ consists of two disjoint HMMs, $\mathcal{M}$ and $\mathcal{N}$, which have been concatenated together as in Figure 2.3. Let $\vec{p}_\mathcal{Q} = \left[\vec{p}, \vec{0}_\mathcal{N}\right]$ be a prior on $\mathcal{Q}$ such that it equals the prior $\vec{p}$ on the states corresponding to $\mathcal{M}$ and is zero on the states corresponding to $\mathcal{N}$. Also define $\vec{q}_\mathcal{Q} = \left[\vec{0}_\mathcal{M}, \vec{q}\right]$ similarly. Then, by construction, it must be true for any $x \in \mathcal{O}^* \cup \{\epsilon\}$ that $\Pr(x|\mathcal{M}, \vec{p}) = \Pr(x|\mathcal{Q}, \vec{p}_\mathcal{Q})$ and also $\Pr(x|\mathcal{N}, \vec{q}) = \Pr(x|\mathcal{Q}, \vec{q}_\mathcal{Q})$. So $\mathcal{M}(\vec{p}) \Leftrightarrow \mathcal{N}(\vec{q})$ if and only if $\vec{p}_\mathcal{Q}$ and $\vec{q}_\mathcal{Q}$ are equivalent priors for our new HMM $\mathcal{Q}$. Therefore, as a corollary of the results from the previous section, we can check equivalence of two initialized Hidden Markov Models in $O((n+m)^3 k)$ time if the models have n and m states respectively and share an output set of size k.

In the next section we will investigate algorithms for deciding subset relations and and equivalence of Hidden Markov Models.

2.4 Equivalence of Hidden Markov Models

In Chapter 1 we discussed the interpretation of HMMs as representations for classes of stochastic processes, whose elements are derived by initializing prior distributions on the models. Definition 2.4 defined an HMM $\mathcal{N}$ to be a subset of an HMM $\mathcal{M}$ ($\mathcal{N} \subseteq \mathcal{M}$) when every process that can be represented by $\mathcal{N}$ can also be represented by $\mathcal{M}$. Equivalence of Hidden Markov Models was defined by saying $\mathcal{M} \Leftrightarrow \mathcal{N}$ exactly when



To test equivalence of two Initialized HMMs, $\mathcal{M}$ and $\mathcal{N}$, we first construct a larger HMM $\mathcal{Q}$, which contains $\mathcal{M}$ and $\mathcal{N}$ as disjoint internal chains. If $\vec{p}$ and $\vec{q}$ are the fixed priors on $\mathcal{M}$ and $\mathcal{N}$ respectively, checking equivalence of the priors $(\vec{p}, \vec{0})$ and $(\vec{0}, \vec{q})$ for the model Q should check that $\mathcal{M}$ and $\mathcal{N}$ are equivalent Initialized HMMs.

Figure 2-3: Checking Equivalence of Initialized HMMs

$\mathcal{M} \subseteq \mathcal{N}$ and $\mathcal{N} \subseteq \mathcal{M}$. This definition partitions HMMs into disjoint equivalence classes that are representations for the same sets of stochastic processes. (This does not, of course, partition the stochastic processes representable by HMMs into disjoint classes since a given process may be representable by non-equivalent HMMs.) Our goal in the next chapter will be to find a way of generating a minimal, canonical representative of each equivalence class in order to isolate the essential expressive degrees of freedom in an HMM. Producing such canonical representations will also reduce the computational overhead involved in the use of large models. As a prelude, in this section, we will develop an algorithm that will check whether two models $\mathcal{M}$ and $\mathcal{N}$ are in a subset relation to each other. A corollary will let us check

equivalence of Hidden Markov Models. We will build up to the algorithm and the associated characterization of equivalent HMMs by proving a series of lemmas.

Let $\mathcal{M}_1 = (\mathcal{S}_1, \mathcal{O}, \mathbf{A}_1, \mathbf{B}_1)$ and $\mathcal{M}_2 = (\mathcal{S}_2, \mathcal{O}, \mathbf{A}_2, \mathbf{B}_2)$ be two Hidden Markov Models. From the definitions we see that $\mathcal{M}_2 \subseteq \mathcal{M}_1$ exactly when for every prior $\vec{p}_2$ on $\mathcal{M}_2$ we can find a prior $\vec{p}_1$ on $\mathcal{M}_1$ that makes $\mathcal{M}_1(\vec{p}_1) \Leftrightarrow \mathcal{M}_2(\vec{p}_2)$. Using the definition of equivalent Initialized HMMs (Definition 2.2) we can write this as: *for every prior $\vec{p}_2$ on $\mathcal{M}_2$ there exists a prior $\vec{p}_1$ on $\mathcal{M}_1$ such that $\forall x \in \mathcal{O}^* \cup \{\epsilon\}$ we can write $\mathbf{B}_1\mathbf{T}_1(x)\vec{p}_1 = \mathbf{B}_2\mathbf{T}_2(x)\vec{p}_2$.* This implies the following lemma which essentially says that there is a stochastic matrix that transforms the priors on one machine into equivalent priors on the other.

Lemma 2.1 *Transformation of Priors*

If $\mathcal{M}_1 = (\mathcal{S}_1, \mathcal{O}, \mathbf{A}_1, \mathbf{B}_1)$ and $\mathcal{M}_2 = (\mathcal{S}_2, \mathcal{O}, \mathbf{A}_2, \mathbf{B}_2)$ then $\mathcal{M}_2 \subseteq \mathcal{M}_1$ if and only if there exists a stochastic matrix[4] $\mathbf{C}$ such that $\forall x \in \mathcal{O}^* \cup \{\epsilon\}$, $\mathbf{B}_1\mathbf{T}_1(x)\mathbf{C} = \mathbf{B}_2\mathbf{T}_2(x)$.

Proof: First, suppose $\mathcal{M}_2 \subseteq \mathcal{M}_1$. Let $\vec{e}_2(i)$ be a prior on $\mathcal{M}_2$ with all its mass on state s_i. Let $\vec{p}_1(i)$ be the corresponding prior on $\mathcal{M}_1$ such that $\forall x \in \mathcal{O}^* \cup \{\epsilon\}$, $\mathbf{B}_1\mathbf{T}_1(x)\vec{p}_1(i) = \mathbf{B}_2\mathbf{T}_2(x)\vec{e}_2(i)$. Such an $\vec{p}_1(i)$ exists by assumption of $\mathcal{M}_2 \subseteq \mathcal{M}_1$. Let $\mathbf{C}$ be a matrix whose i^{th} column is $\vec{p}_1(i)$. In other words, $\mathbf{C} = [\vec{p}_1(1)|\vec{p}_1(2)|\cdots|\vec{p}_1(n_2)]$ where n_2 is the number of states in $\mathcal{M}_2$. It is clear that any prior on $\mathcal{M}_2$ can be written as $\vec{p}_2 = \sum_{i=1}^{n_2} p_i\vec{e}_2(i)$ and that we will have:

$$\begin{aligned}
\forall x \in \mathcal{O}^* \cup \{\epsilon\}, \quad \mathbf{B}_2\mathbf{T}_2(x)\vec{p}_2 &= \mathbf{B}_2\mathbf{T}_2(x)\sum_{i=1}^{n_2} p_i\vec{e}_2(i) \\
&= \mathbf{B}_1\mathbf{T}_1(x)\sum_{i=1}^{n_2} p_i\vec{p}_1(i) \\
&= \mathbf{B}_1\mathbf{T}_1(x)\mathbf{C}\vec{p}_2
\end{aligned}$$

[4]By "stochastic matrix" we mean a matrix whose entries are all non-negative and whose columns sum to one

Since this is true for *any* $\vec{p}_2$ we can conclude that if $\mathcal{M}_2 \subseteq \mathcal{M}_1$ then $\forall x \in \mathcal{O}^* \cup \{\epsilon\}$ we can write $\mathbf{B}_1\mathbf{T}_1(x)\mathbf{C} = \mathbf{B}_2\mathbf{T}_2(x)$. Furthermore, by construction, $\mathbf{C}$ is stochastic. To prove the lemma in the other direction, suppose that the matrix $\mathbf{C}$ exists and, for any prior $\vec{p}_2$ on $\mathcal{M}_2$, let $\vec{p}_1 = \mathbf{C}\vec{p}_2$ be the corresponding prior on $\mathcal{M}_1$. Then, by the definition of equivalence, $\mathcal{M}_1(\vec{p}_1) \Leftrightarrow \mathcal{M}_2(\vec{p}_2)$ since $\forall x \in \mathcal{O}^* \cup \{\epsilon\}$ we can write $\mathbf{B}_1\mathbf{T}_1(x)\left(\mathbf{C}\vec{p}_2\right) = \mathbf{B}_2\mathbf{T}_2(x)\vec{p}_2$. Since this is true for any $\vec{p}_2$ we conclude that $\mathcal{M}_2 \subseteq \mathcal{M}_1$. □

Lemma 2.1 is not a sufficiently powerful characterization of equivalence of HMMs to enable us to construct an algorithm to check equivalence. Essentially, we want to find a necessary and sufficient condition that does not require us to examine every finite prefix of outputs of a process in order to check the equivalence of models. Our previous results achieved this goal by examining the properties of various vector spaces and checking an equivalence condition on their bases. The next lemma we prove will tell us how to find such a vector space that allows us to relax the equivalence condition in Lemma 2.1. In order to do this we need to introduce a little additional notation.

Definition 2.5 *Suffix Matrix*

Let $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$ *be an HMM. Define a* **suffix matrix** $\Sigma(x) = \mathbf{B}\mathbf{T}(x)$ *for every* $x \in \mathcal{O}^* \cup \{\epsilon\}$. *So* $\Sigma(x)_{ij} = \Pr(\mathcal{M}\ \textit{emits}\ xo_i | \mathcal{M}\ \textit{started in state}\ s_j)$. *The name* **suffix matrix** *originates from the observation that if* $z = xy$ *is a string with prefix* x *and suffix* y, *then* $\Sigma(z) = \Sigma(y)\mathbf{T}(x)$. *Suppose* y *is any string in* $\mathcal{O}^*$. *Then we can always write* $y = xo_i$ *where* $o_i \in \mathcal{O}$ *and* $x \in \mathcal{O}^* \cup \{\epsilon\}$. *For any* $y = xo_i \in \mathcal{O}^*$ *we will use the notation* $\vec{\sigma}(y)$ *to mean the* i^{th} *row of* $\Sigma(x)$. *The* j^{th} *component of* $\vec{\sigma}(y)$ *satisfies the equation* $\sigma(y)_j = \Pr(\mathcal{M}\ \textit{emits}\ y | \mathcal{M}\ \textit{started in state}\ s_j)$.

Lemma 2.1 implies that if $\mathcal{M}_2 \subseteq \mathcal{M}_1$, then linear dependence amongst the rows of $\Sigma_1(x)$ implies dependence amongst the rows of $\Sigma_2(x)$. This provides a clue that the key to equivalence of HMMs lies in comparing the spaces spanned by the rows of the suffix matrix. Investigating this idea leads to the following lemma.

Lemma 2.2 *Equivalence Condition*

Let $\mathcal{M}_1 = (\mathcal{S}_1, \mathcal{O}, \mathbf{A}_1, \mathbf{B}_1)$ *and* $\mathcal{M}_2 = (\mathcal{S}_2, \mathcal{O}, \mathbf{A}_2, \mathbf{B}_2)$ *be two Hidden Markov Models. Let* $\mathcal{U}_1 = \{\vec{\sigma}_1(y) : y \in \mathcal{O}^*\}$ *be the set of all rows of the suffix matrices of* $\mathcal{M}_1$*. Let* $\mathcal{V} = \{\vec{\sigma}_1(x_1), \vec{\sigma}_1(x_2), \cdots \vec{\sigma}_1(x_l)\}$ *be a basis for* $Span(\mathcal{U}_1)$*. Then* $\mathcal{M}_2 \subseteq \mathcal{M}_1$ *if and only if there exists a stochastic matrix* $\mathbf{C}$ *that satisfies the following conditions:*

$$\forall o_j \in \mathcal{O},\ \ \vec{\sigma}_1(o_k)\mathbf{C} = \vec{\sigma}_2(o_k) \tag{2.3}$$

$$\forall x_i \text{ such that } \vec{\sigma}_1(x_i) \in \mathcal{V},\ \ \vec{\sigma}_1(x_i)\mathbf{C} = \vec{\sigma}_2(x_i) \tag{2.4}$$

$$\forall o_j \in \mathcal{O} \text{ and } \forall \vec{\sigma}_1(x) \in \mathcal{V},\ \ \vec{\sigma}_1(x)\left[\mathbf{T}_1(o_j)\mathbf{C} - \mathbf{C}\mathbf{T}_2(o_j)\right] = 0 \tag{2.5}$$

Prior to proving this lemma it will help to gain some intuition for what it means. Remember that the matrix $\mathbf{C}$ in Lemma 2.1 transforms priors on $\mathcal{M}_2$ into priors on $\mathcal{M}_1$, and that the j^{th} component of $\vec{\sigma}_1(x)$ is the probability of emitting string x, having started in state s_j. Using these two facts we can see that Equation 2.4 says that that for any choice of priors on $\mathcal{M}_2$ there is a prior on $\mathcal{M}_1$ such that the probability of emitting a string y is the same for both models if $\vec{\sigma}_1(y)$ is in the basis for $Span\,(\mathcal{U}_1)$. Equation 2.3 says the same thing for all strings of length one. We will eventually use these two facts in the base case of an induction to prove the lemma. We will see that Equation 2.5 is a way of saying that if $\vec{\sigma}_1(x)\mathbf{C} = \vec{\sigma}_2(x)$ for some x then this condition is also satisfied for any string y that is one symbol longer than x. We will use this as the induction step in the proof below.

Proof: First we will prove that if $\mathcal{M}_2 \subseteq \mathcal{M}_1$ then Equations 2.3 to 2.5 will be true. So suppose that $\mathcal{M}_2 \subseteq \mathcal{M}_1$. Then by Lemma 2.1, there is a stochastic matrix $\mathbf{C}$ such that for every $x \in \mathcal{O}^* \cup \{\epsilon\}$, every row $\vec{\sigma}_1(xo_i)$ of $\Sigma_1(x)$ satisfies $\vec{\sigma}_1(xo_i)\mathbf{C} = \vec{\sigma}_2(xo_i)$ where $\vec{\sigma}_2(xo_i)$ is the corresponding row of $\Sigma_2(x)$. This at once makes Equations 2.3 and 2.4 true. Then we turn to Equation 2.5. Let x be any string in $\mathcal{O}^* \cup \{\epsilon\}$ and let $y = o_i x$ be an $|x| + 1$ long string with x as a suffix. Then by assumption of $\mathcal{M}_2 \subseteq \mathcal{M}_1$, and using the definition of the suffix matrix we can make the following

series of statements:

$$\begin{aligned} \vec{\sigma}_1(y)\mathbf{C} &= \vec{\sigma}_2(y) \\ \vec{\sigma}_1(x)\mathbf{T}_1(o_i)\mathbf{C} &= \vec{\sigma}_2(x)\mathbf{T}_2(o_i) \\ \vec{\sigma}_1(x)\mathbf{T}_1(o_i)\mathbf{C} &= \vec{\sigma}_1(x)\mathbf{C}\mathbf{T}_2(o_i) \\ \Longrightarrow \vec{\sigma}_1(x)\left[\mathbf{T}_1(o_i)\mathbf{C} - \mathbf{C}\mathbf{T}_2(o_i)\right] &= 0 \end{aligned} \tag{2.6}$$

The second equation is derived from the first from the definition of $\Sigma(x)$ and $\vec{\sigma}(x)$. The third equation simply replaces $\vec{\sigma}_2(x)$ by $\vec{\sigma}_1(x)\mathbf{C}$ by assumption of $\mathcal{M}_2 \subseteq \mathcal{M}_1$ and Lemma 2.1. Since Equation 2.6 holds for every $o_i \in \mathcal{O}$ and for every x such that $\vec{\sigma}_1(x) \in \mathcal{V}$, we have proven the necessity of the lemma. Next we will prove the lemma in the other direction. Suppose that a stochastic matrix $\mathbf{C}$ satisfying the conditions of the lemma exists. Then, by Equation 2.3, $\vec{\sigma}_1(x)\mathbf{C} = \vec{\sigma}_2(x)$ for every string x of length 1. Then assume that for all x of length less than or equal to l we can write $\vec{\sigma}_1(x)\mathbf{C} = \vec{\sigma}_2(x)$. For any such x ($|x| \leq l$) we can write $\vec{\sigma}_1(x) = \sum_{i=1}^{|\mathcal{V}|} d_i\vec{\sigma}_1(x_i)$ for some choice of d_i, where the $\vec{\sigma}_1(x_i)$ are elements of the basis $\mathcal{V}$. So, by the induction assumption, and Equation 2.4, we can write: $\vec{\sigma}_2(x) = \vec{\sigma}_1(x)\mathbf{C} = \sum_{i=1}^{|\mathcal{V}|} d_i\vec{\sigma}_1(x_i)\mathbf{C} = \sum_{i=1}^{|\mathcal{V}|} d_i\vec{\sigma}_2(x_i)$. But this means that for every output o_i, we can use Equation 2.5 to write:

$$\vec{\sigma}_1(o_ix)\mathbf{C} = \vec{\sigma}_1(x)\mathbf{T}_1(o_i)\mathbf{C} = \sum_{i=1}^{|\mathcal{V}|} d_i\vec{\sigma}_1(x_i)\mathbf{T}_1(o_i)\mathbf{C} \tag{2.7}$$

$$= \sum_{i=1}^{|\mathcal{V}|} d_i\vec{\sigma}_1(x_i)\mathbf{C}\mathbf{T}_2(o_i) \tag{2.8}$$

$$= \sum_{i=1}^{|\mathcal{V}|} d_i\vec{\sigma}_2(x_i)\mathbf{T}_2(o_i) \tag{2.9}$$

$$= \vec{\sigma}_2(x)\mathbf{T}_2(o_i) = \vec{\sigma}_2(o_ix) \tag{2.10}$$

We go from Equation 2.7 to Equation 2.8 by applying condition 2.5 of the lemma. The next two lines simply substitute the expression for $\vec{\sigma}_2(x)$ obtained from the induction

assumption. The conclusion is that if $\vec{\sigma}_1(x)\mathbf{C} = \vec{\sigma}_2(x)$ for all strings x of length less than or equal to l, then the same is true for strings of length $l+1$. This completes the induction and proves that for every x, we can write $\vec{\sigma}_1(x)\mathbf{C} = \vec{\sigma}_2(x)$, implying that $\forall x \in \mathcal{O}^* \cup \{\epsilon\}\Sigma_1(x)\mathbf{C} = \Sigma_2(x)$. By Lemma 2.1, this shows that $\mathcal{M}_2 \subseteq \mathcal{M}_1$. □

Lemma 2.2 could be used to build a polynomial time algorithm for testing equivalence of HMMs. Such an algorithm would begin by generating the basis $\mathcal{V}$ in the lemma. We would use the efficient basis-generation technique used in Step 1 of our algorithm for checking equivalence of prior distributions. Then we would use linear programming techniques to find a matrix $\mathbf{C}$ satisfying the conditions of the lemma.[5] $\mathcal{M}_2 \subseteq \mathcal{M}_1$ only if such a matrix is found. Since linear programming problems can be solved quickly, such an algorithm would run in polynomial time.([karmarkar84]) However, it is possible to do even better. Some recent results in the theory of probabilistic automata ([tzeng]), that are achieved using methods similar to ours, suggest that the following lemma should be true.

Lemma 2.3 *All* $\mathbf{C}$ *matrices are equivalent*

Let $\mathbf{C}_1$ *and* $\mathbf{C}_2$ *be any two stochastic matrices satisfying* $\vec{\sigma}_1(x_i)\mathbf{C}_1 = \vec{\sigma}_1(x_i)\mathbf{C}_2$ *for every* $\vec{\sigma}_1(x_i) \in \mathcal{V}$*, where* $\mathcal{V}$ *is the basis in Lemma 2.2. Then, for any string* x *we can write* $\vec{\sigma}_1(x)\mathbf{C}_1 = \vec{\sigma}_1(x)\mathbf{C}_2$*.*

Proof: Suppose x is any string. Then for some choice of d_i we know that $\vec{\sigma}_1(x) = \sum_{i=1}^{|\mathcal{V}|} d_i\vec{\sigma}_1(x_i)$ where the $\vec{\sigma}_1(x_i)$ are the elements of the basis in Lemma 2.2. Then it is clear that $\vec{\sigma}_1(x)\mathbf{C}_1 = \sum_{i=1}^{|\mathcal{V}|} d_i\vec{\sigma}_1(x_i)\mathbf{C}_1 = \sum_{i=1}^{|\mathcal{V}|} d_i\vec{\sigma}_1(x_i)\mathbf{C}_2 = \vec{\sigma}_1(x)\mathbf{C}_2$. □

Collecting all our lemmas together, we can finally state our theorem characterizing equivalent Hidden Markov Models.

[5] We need to use linear programming rather than straightforward linear algebra because the stochasticity constraints on $\mathbf{C}$ involve inequalities.

Theorem 2.3 *Equivalence of HMMs*

Let $\mathcal{M}_1 = (\mathcal{S}_1, \mathcal{O}, \mathbf{A}_1, \mathbf{B}_1)$ *and* $\mathcal{M}_2 = (\mathcal{S}_2, \mathcal{O}, \mathbf{A}_2, \mathbf{B}_2)$ *be two Hidden Markov Models. Let* $\mathcal{U}_1 = \{\vec{\sigma}_1(y) : y \in \mathcal{O}^*\}$ *be the set of all rows of the suffix matrices of* $\mathcal{M}_1$*. Let* $\mathcal{V} = \{\vec{\sigma}_1(x_1), \vec{\sigma}_1(x_2), \cdots \vec{\sigma}_1(x_l)\}$ *be a basis for* $Span(\mathcal{U}_1)$*. Then* $\mathcal{M}_2 \subseteq \mathcal{M}_1$ *if and only if the following two conditions hold. (a) There exists a stochastic matrix* $\mathbf{C}$ *such that for every* x_i *satisfying* $\vec{\sigma}_1(x_i) \in \mathcal{V}$ *we can write* $\vec{\sigma}_1(x_i)\mathbf{C} = \vec{\sigma}_2(x_i)$*. (b) For* **any** *stochastic* $\mathbf{C}$ *satisfying condition (a), the following must be true:*

$$\forall o_j \in \mathcal{O},\ \ \vec{\sigma}_1(o_k)\mathbf{C} = \vec{\sigma}_2(o_k) \qquad (2.11)$$

$$\forall o_j \in \mathcal{O} \text{ and } \forall \vec{\sigma}_1(x) \in \mathcal{V},\ \ \vec{\sigma}_1(x)\left[\mathbf{T}_1(o_j)\mathbf{C} - \mathbf{C}\mathbf{T}_2(o_j)\right] = 0 \qquad (2.12)$$

$\mathcal{M}_1 \Leftrightarrow \mathcal{M}_2$ *if and only if* $\mathcal{M}_2 \subseteq \mathcal{M}_1$ *and* $\mathcal{M}_1 \subseteq \mathcal{M}_2$*.*

Proof: The proof follows easily from Lemmas 2.2 and 2.3. Suppose the conditions (a) and (b) of our theorem hold, and pick any $\mathbf{C}$ satisfying them. This $\mathbf{C}$ also satisfies the conditions of Lemma 2.2 so that $\mathcal{M}_2 \subseteq \mathcal{M}_1$. So conditions (a) and (b) are sufficient to guarantee that $\mathcal{M}_2 \subseteq \mathcal{M}_1$. Next we show that they are also necessary conditions. So suppose that $\mathcal{M}_2 \subseteq \mathcal{M}_1$. First notice that Equation 2.11 says that $\vec{\sigma}_1(x)\mathbf{C} = \vec{\sigma}_2(x)$ for every string x of length 1. Also remember from the proof of Lemma 2.2 that Equation 2.12 essentially says that if $\vec{\sigma}_1(x) \in \mathcal{V}$, then any string $y = o_i x$ satisfies the condition $\vec{\sigma}_1(y)\mathbf{C} = \vec{\sigma}_2(y)$. Lemma 2.3 tells us that if $\mathbf{C}_1$ and $\mathbf{C}_2$ both satisfy condition (a), then $\vec{\sigma}_1(x)\mathbf{C}_1 = \vec{\sigma}_1(x)\mathbf{C}_2$ for any string x. So, if *any* $\mathbf{C}$ satisifies condition (a) and the equations of condition (b), then *every* $\mathbf{C}$ satisfying (a) also satisfies condition (b). By Lemma 2.2 there is a stochastic matrix $\mathbf{C}$ satisfying condition (a) and Equations 2.11 and 2.12. Therefore, as discussed above, *every* $\mathbf{C}$ fulfilling condition (a) also satisfies the equations of (b). This proves that the (a) and (b) are necessary conditions for $\mathcal{M}_2 \subseteq \mathcal{M}_1$ to be true. We have already shown that they are sufficient conditions and so our proof of the theorem is complete. □

Algorithm: We can use Theorem 2.3 to develop a polynomial time algorithm to test equivalence of HMMs. We do this by first checking if $\mathcal{M}_2 \subseteq \mathcal{M}_1$ and then checking $\mathcal{M}_1 \subseteq \mathcal{M}_2$. So suppose we are trying to check that $\mathcal{M}_2 \subseteq \mathcal{M}_1$. The subset-checking algorithm starts by generating the basis of Theorem 2.3 using the method of Step 1 in the algorithm for determining equivalence of priors. It then tries to find a matrix $\mathbf{C}$ satisfying the equivalence condition (a) for this basis. If no such matrix can be found, then $\mathcal{M}_2 \not\subseteq \mathcal{M}_1$. If a $\mathbf{C}$ satisfying condition (a) is found, we check that it satisfies the equations of condition (b). If it passes this test, Lemma 2.2 tells us that $\mathcal{M}_2 \subseteq \mathcal{M}_1$. We check $\mathcal{M}_1 \subseteq \mathcal{M}_2$ similarly and answer the question of equivalence appropriately. Correctness of this algorithm is immediate from the correctness of our earlier algorithm to determine equivalence of priors, and from Theorem 2.3.

We will now compute the running time of the HMM equivalence algorithm, assuming unit cost arithmetic. First of all, it takes $O(k(n_1^2 + n_2^2))$ time to generate all the $\mathbf{T}_1(o_i)$ and $\mathbf{T}_2(o_i)$ matrices from the parameters of the HMMs. From our earlier analysis, the basis-finding algorithm takes worst-case time $O(n_1^3 k)$ when appropriately optimized. We also need to compute $\vec{\sigma}_2(x_i)$ corresponding to the $\vec{\sigma}_1(x_i) \in \mathbf{V}$. This can be done at the same time that the basis is generated, simply adding a factor of 2 to the cost. Once the basis is generated, finding a matrix $\mathbf{C}$ satisfying condition (a) involves solving a system of $n_2|\mathbf{V}|$ equations in $n_1 n_2$ variables, subject to $n_2 + n_1 n_2$ stochasticity constraints. Since the constraints involve only linear inequalities (the columns of $\mathbf{C}$ sum to one and $\forall i, j \;\; \mathbf{C}_{ij} \geq 0$) we can solve for $\mathbf{C}$ using linear programming.([chvatal80]) Karmarkar ([karmarkar84]) gives a worst-case $O(Ln^{3.5})$ time algorithm for linear programming where n is the number of variables and L is size of the linear program in bits. (This is also competitive in practice with the simplex algorithm.) It is a somewhat sticky business to translate the bit complexity in terms of L into a complexity in terms of the number of variables and equations in the linear program. In rough terms, if we are dealing with a fixed number of bits per number, we can say that L is of the order of $O(mn)$, where mn is roughly the size of

the linear programming tableau. Using this, we conclude that we can find $\mathbf{C}$, if a solution exists, in worst-case time $O\left[(n_2|V| + n_2 + n_1 n_2)(n_1 n_2)^{4.5}\right] = O\left[(n_1 n_2)^{5.5}\right]$ where we have used the fact that $|\mathbf{V}| \leq n_1$. Once we have generated a matrix $\mathbf{C}$, checking that it satisfies Equation 2.11 takes time $O(k n_1 n_2)$ and checking Equation 2.12 takes time $O\left[n_1 k(n_1^2 + n_2^2 + 2n_1 n_2)\right]$. (Once again, we have used the fact that $|\mathbf{V}| \leq n_1$.) Gathering all these terms together, and picking the dominant terms as n_1, n_2 and k grow large, we find that our algorithm for checking $\mathcal{M}_2 \subseteq \mathcal{M}_1$ runs in worst-case time $O\left(n_1 k(n_1^2 + n_2^2 + 2n_1 n_2) + (n_1 n_2)^{5.5}\right)$. The complexity of checking $\mathcal{M}_1 \subseteq \mathcal{M}_2$ is obtained by exchanging n_1 and n_2 everywhere in this expression. The algorithm presented here can be optimized in various ways to do somewhat better, but these optimizations are less interesting and more complicated to explain.

Chapter 3

Reduction to Canonical Forms

In the previous chapter we defined equivalence of stochastic processes and proved how and why prior distributions on a model may be equivalent. We used these results to characterize equivalent Initialized Hidden Markov Models. Finally, we made various appeals to linear algebraic arguments to develop necessary and sufficient conditions for the equivalence of HMMs. However, our results concerning equivalent HMMs did not give a clear intuitive characterization of the intrinsic expressiveness of Hidden Markov Models. In an effort to achieve such a characterization, this chapter will define the *canonical dimension* of a model. The definition is related to our formulation of the theorems describing equivalent HMMs, and will lead quickly to an algorithm for finding canonical representations of models. All the theorems in this section will be proven in the context of Generalized Markov Models (GMMs) which relax the postitivity constraints on the parameters of HMMs. We will see that all processes that can be modelled exactly by Hidden Markov Models can also be modelled by Generalized Markov Models. Some kinds of GMMs, with appropriate restrictions placed on the allowable prior distributions, are equivalent to HMMs. In Section 3.2.1 we will see how the results achieved in this chapter should be modified to apply to HMMs. We begin by defining Generalized Markov Models and discussing their properties.

3.1 Generalized Markov Models

In this section we will define a new class of models of stochastic processes. Since this new class contains the processes modelled by traditional Hidden Markov Models, we will christen it the class of *Generalized Markov Models*. Essentially, the generalization involves relaxing the positivity constraint imposed by the probabilistic interpretation of the parameters describing the underlying Markov Chain of an HMM. First we will discuss why such a generalization may be a good idea, and then we will proceed to define GMMs and describe their properties.

3.1.1 Why Should We Invent GMMs?

Empirical Reasons: Our first motivation for defininining GMMs is empirical. L.Niles, in discussing the connections between stochastic classifiers and neural network schemes, describes experiments with an HMM-net, a network implementation of an HMM.[niles90] He reports that corrective training methods lead to HMM-net parameters that violate probability constraints, but are more more successful in classification tasks. Niles points out that relaxing the stochasticity constraint on HMM parameters while preserving the formal structure[1] results in a perfectly valid classifier and decision-boundary model. Of course, the Bayesian formulation of classification is lost. However, Bayesian methods are only optimal if the true distributions are known, and this is very far from the case in most applications of HMMs. In light of these facts, Niles suggests that HMMs with "negative parameters" may be interesting because, in the HMM-net formulation of Hidden Markov Models, they have a natural interpretation as inhibitory connections. If we wish to follow this lead and investigate the properties of various HMM-like models we should be able to *analytically* compare the properties of the different schemes in order to be able to choose between them in a principled

[1] By formal structure we mean, for example,the formal manipulations by which posterior probabilities are extracted from the model. Of course, once the model parameters cannot be interepreted as probabilities, we will be computing some non-probabilistic score.

manner. This thesis initially arose from an attempt to understand the properties of HMMs sufficiently well to facilitate comparison with other classification schemes. The Generalized Markov Models we will define in this chapter are a natural generalization of HMMs which follow the empirical lead in [niles90] suggesting that "negative parameters" may be a good idea. We are able to describe, in detail, the conections between GMMs and HMMs.

Theoretical Reasons: We are also motivated to define Generalized Markov Models from a theoretical perspective. First of all, we will take the view that an HMM is simply an iterative, finite-state scheme used to represent the statistics of stochastic processes. The interpretation of the model parameters as probabilities is peripheral to the actual goal of realizing parsimonious and easily manipulated representations of wide classes of stochastic processes. Therefore, there is no intrinsic reason why the paramaters of the model should be probabilities, unless we derive a clear benefit from the constraints imposed by such an interpretation. If we discover that allowing negative parameters in our model permits us to build better models, we should not allow the probabilistic viewpoint to stop us. Secondly, in vague terms, all the results from the previous chapter dealt with general linear combinations of elements of vector spaces as opposed to convex combinations of vectors on simplices. (Probabilistic parameter spaces normally lead to the latter situation.) It seems natural, therefore, to ask whether it is really necessary for the parameters of an HMM-like model to be positive in order to successfully model stochastic processes. For example, we may be able to define a prior with "negative" parameters, without changing the probability distributions over outputs that we care about. Suppose $\vec{p}$ is a prior on a model $\mathcal{M}$, and $\mathcal{I}$ is an invariant subspace of the null-space of the output matrix. Then we can remove the components of a $\vec{p}$ that lie in $\mathcal{I}$ and the resulting vector $\vec{p}\,'$ will induce the same stochastic process on $\mathcal{M}$. (See the theorems in Section 2.2) Notice that $\vec{p}\,'$ *may have negative components*, although it must still sum to one since the vectors in $\mathcal{I}$ necessarily sum to zero. Given this fact, define a *valid prior* to be *any* (not

necessarily stochastic) vector that induces a valid stochastic process when it initializes a model. Clearly, from the above discussion, the set of valid priors extends beyond the probability simplex. Extending the argument, we could permit the columns of the transition matrix **A** of a model to also be pseudo-stochastic.[2] A generalized model, defined by relaxing constraints in this fashion, has the potential to model a wider class of processes with the same number of states. This is particularly important in pattern recognition applications because it is usually far from clear that the true model of the system is a probabilistic function of a Markov Chain. Typically, the best we can hope for is to *approximate* the statistics of a process as closely as possible with our model. Therefore, a more expressive formalism could intrinsically provide a better model.

Reasons of Parsimony: The final reason to consider Generalized Markov Models is basically an argument that a smaller model is usually better. As discused in the previous paragraph, we would like to have more expressive formalisms for modelling stochastic processes since we are typically dealing with problems of approximating a system. However, if the formalism involves too many degrees of freedom, it will suffer from the curse of dimensionality - it will become very difficult to estimate the values of the model parameters from the sparse data that is typically available. So we basically want to "say more with fewer parameters". We can also make the computational argument that, in general, the more parameters we have to manipulate, the slower all our algorithms will be. At the same time, the formal methods of manipulating HMMs are so easy, intuitive and efficient that we would love to be able to keep them. The Generalized Markov Models defined in this thesis achieve both these goals by preserving the formal structure of HMMs, but liberating them from constraints that limit the class of processes a given number of parameters could model. Essentially, we attempt to get more mileage from each parameter of a model

[2] We do not relax the stochastic constraints on the ouput matrix because this makes analysis considerably harder.

by allowing it to range over a greater domain in a natural way. We will see, for example, that the smallest HMM equivalent to a given model may have more states that its smallest representation in the GMM formalism. This is our principal reason for defining GMMs.

We can see from these arguments that it may be worthwhile to consider generalizations of HMMs as techniques for modelling stochastic processes, specially for pattern recognition applications. In particular, we have seen that it may be a good idea to relax the positivity constraint on the parameters of Hidden Markov Models. We will now define Generalized Markov Models and discuss their properties.

3.1.2 Definition of GMMs

Our first task is to define what we mean by "relaxing the positivity constraint" on probabilities. To this end we make the following definition of a *pseudo-stochastic* vector:

Definition 3.1 *Pseudo-probability and Pseudo-stochasticity*
Define an n-dimensional vector $\vec{v}$ to be **pseudo-stochastic** *if each of its components is real and $\sum_{i=1}^{n} v_i = 1$. Each entry of such a vector is called a* **pseudo-probability**. *Pseudo-probabilities of alternative independent events add just like true probabilities. Also define a pseudo-stochastic matrix to be one whose columns are pseudo-stochastic vectors. A pseudo-Markov Chain is a Markov Chain whose transition matrix and prior distribution are both pseudo-stochastic. In the rest of this chapter we will use frequently use the term "probability" even when we mean pseudo-probability. The usage will be obvious from the context.*

We will define GMMs by essentially replacing the probabilities describing the underlying Markov Chain of an HMM with pseudo-probabilities. We will need to impose some additional constraints on allowable priors on to ensure that the model describes valid stochastic processes.

Definition 3.2 *Generalized Markov Models (GMMs)*

A Generalized Markov Model is defined as a quadruple $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$ *where* $\mathcal{S}$ *is a set of* n *states,* $\mathcal{O}$ *is a discrete set of* k *outputs and* $\mathbf{B}$ *is a stochastic output matrix as in the definition of HMMs. Define an* n*-dimensional pseudo-probability vector* $\vec{v}$ *to be* **possible** *for* $\mathcal{M}$ *if the product* $\mathbf{B}\vec{v}$ *is a stochastic vector. (In other words* $\vec{v}$ *is possible if* $\mathbf{B}$ *maps* $\vec{v}$ *to a probability distribution over the outputs.) Also define an* n*-dimensional vector* $\vec{u}$ *to be* **valid** *for* $\mathcal{M}$ *if* $\vec{u}$ *induces a valid stochastic process when* $\mathcal{M}$ *is initialized by* $\vec{u}$ *and evolved according to the formal rules specified in Chapter 1. We demand that all* n*-dimensional stochastic vectors be valid for* $\mathcal{M}$*. The transition matrix* $\mathbf{A}$ *of a GMM must then be a pseudo-stochastic matrix whose columns are valid vectors for* $\mathcal{M}$*.*

We can see that GMMs are very similar to HMMs except that the underlying chain is a pseudo-Markov Chain. By this definition, every HMM is structurally a GMM, but in the GMM formulation we would be permitted to initialize the model with valid priors that are not stochastic. Definition 3.2 is not very constructive in that it does not characterize what the valid priors on a model look like. The results we will arrive at in this chapter, including the derivation of canonical forms for GMMs, do not require such a characterization. We will return to this sticky issue briefly at the end of the section.

GMM Evolution: We will evolve a GMM forward in time by treating pseudo-probabilities formally as if they are true probabilities. In particular projection and transition operators are formally defined exactly as in Table 1.1. The only difference lies in the interpretation of the various quantities. The $(ij)^{th}$ component of the transition operator $\mathbf{T}(o_k)$ is now understood to be the pseudo-probability that the underlying chain will transition from state s_j to s_i, weighted by the true probability of emitting o_k in state s_j. All probabilities related to the states in an HMM are replaced by pseudo-probabilities in a GMM, but we still retain the true probability

interpretation of distributions over outputs. The suffix matrix of Definition 2.5 will be important to us in our discussion of reduction of HMMs. For any string x, the suffix matrix is defined as $\Sigma(x) = \mathbf{BT}(x)$ where $\mathbf{B}$ is the GMM output matrix and $\mathbf{T}(x)$ is the GMM transition operator for string x. In the context of GMMs $\Sigma(x)_{ij}$ is the probability that the model emits the string xo_i given a pseudo-probability of 1 that the model started in state s_j. The meaning of the vectors $\vec{\sigma}(x)$ in Definition 2.5 is also appropriately modified. Henceforth, when we speak of transition operators, suffix matrices or any other quantity originally defined for HMMs in the context of GMMs, we will be referring to these objects interpreted as described above.

3.1.3 Properties of GMMs

The most important observation to make about the properties of Generalized Markov Models is that all the equivalence results of the previous chapter carry over with only minor modifications. In this section we will describe these modifications. First of all, we define equivalence of GMMs and Initialized GMMs in exactly the same terms as for HMMs. Priors are equivalent if the induce the same stochastic process on a model, and initialized models are equivalent if they represent the same stochastic process. The essential difference is just that we will allow pseudo-stochastic priors and transition matrices. Then, Theorems 2.1 and 2.2 concerning equivalence of prior distributions on HMMs apply immediately to equivalence of pseudo-priors on GMMS. We can see this is the case because the proofs of these theorems rely only on the linear structure of the model and do not depend on any property related to stochasticity. Consequently, the characterization of equivalent Initialized HMMs applies at once to Initialized GMMs also. At first sight, it appears to be a little more difficult to translate the theorems concerning equivalence of HMMs into the GMM context, because they appear to require various quantities to be stochastic. However, a more careful examination shows they only depend on the fact that stochastic vectors sum to one. The positivity of probabilities is not used anywhere. We will use this to state

the following lemmas concerning equivalent GMMs. We will only sketch the proofs since they parallel those of Chapter 2 with minor modifications that the reader can easily see. As before, we will say that $\mathcal{M}_2 \subseteq \mathcal{M}_1$ if every stochastic process that can be generated by setting a pseudo-prior on $\mathcal{M}_2$ can also be generated by $\mathcal{M}_1$.

Lemma 3.1 *Transformation of Pseudo-priors on GMMs*
If $\mathcal{M}_1 = (\mathcal{S}_1, \mathcal{O}, \mathbf{A}_1, \mathbf{B}_1)$ *and* $\mathcal{M}_2 = (\mathcal{S}_2, \mathcal{O}, \mathbf{A}_2, \mathbf{B}_2)$ *are GMMs, then* $\mathcal{M}_2 \subseteq \mathcal{M}_1$ *if and only if there exists a* **pseudo-stochastic** *matrix* $\mathbf{C}$*such that we can write* $\mathbf{B}_1\mathbf{T}_1(x)\mathbf{C} = \mathbf{B}_2\mathbf{T}_2(x)$ *for every* $x \in \mathcal{O}^* \cup \{\epsilon\}$*. Furthermore, suppose we know that* $\mathbf{C}'$ *is a pseudo-stochastic matrix that transforms the* **stochastic** *priors* $\vec{p}$ *on* $\mathcal{M}_2$ *into equivalent valid priors* $\vec{q}$ *on* $\mathcal{M}_1$*. Then* $\mathbf{C}'$ *transforms* **every valid prior** *on* $\mathcal{M}_2$ *into an equivalent valid prior on* $\mathcal{M}_1$*, so that* $\mathcal{M}_2 \subseteq \mathcal{M}_1$*.*

Proof: The proof of the first part of Lemma 3.1 follows the proof of Lemma 2.1. Essentially, we consider pseudo-priors $\vec{e}_2(i)$ with all the mass on a state s_i of $\mathcal{M}_2$. By assumption of $\mathcal{M}_1 \Leftrightarrow \mathcal{M}_2$, there are equivalent pseudo-priors $\vec{p}_1(i)$ on $\mathcal{M}_1$. The $\vec{p}_1(i)$ are necessarily valid for $\mathcal{M}_1$ because they induce valid stochastic processes by assumption. The columns of the transformation matrix $\mathbf{C}$, as in Lemma 2.1, will be set equal to the $\vec{p}_1(i)$. The proof then exactly parallels that of Lemma 2.1. To prove the second part of the lemma, suppose that $\mathbf{C}'$ transforms stochastic priors on $\mathcal{M}_2$ on equivalent valid priors on $\mathcal{M}_1$. Then, it transforms the $\vec{e}_2(i)$ into pseudo-stochastic $\vec{p}_1 i = \mathbf{C}'\vec{e}_2(i)$ such that $\Pr(x|\mathcal{M}_2, \vec{e}_2(i)) = \Pr(x|\mathcal{M}_1, \vec{p}_1 i)$ for every string x. Next, observe that *every* valid prior $\vec{p}_2$ on $\mathcal{M}_2$ can be written as a linear combination of the *stochastic* unit priors $\vec{e}_2(i)$: $\vec{p}_2 = \sum_{i=1}^{n_2} a_i \vec{e}_2(i)$. Consequently, we can write for every $x \in \mathcal{O}^* \cup \{\epsilon\}$ that:

$$\begin{aligned} \Pr(x|\mathcal{M}_2, \vec{p}_2) &= \sum_{i=1}^{n_2} a_i \Pr(x|\mathcal{M}_2, \vec{e}_2(i)) \\ &= \sum_{i=1}^{n_2} a_i \Pr(x|\mathcal{M}_1, \mathbf{C}'\vec{e}_2(i)) \end{aligned}$$

$$\begin{aligned} &= \Pr(x|\mathcal{M}_1, \sum_{i=1}^{n_2} a_i \mathbf{C}'\vec{e}_2(i)) \\ &= \Pr(x|\mathcal{M}_1, \mathbf{C}'\vec{p}_2) \end{aligned} \tag{3.1}$$

This indicates that it is always true that $\mathcal{M}_2(\vec{p}_2) \Leftrightarrow \mathcal{M}_1(\mathbf{C}')$ so long as $\vec{p}_2$ is a valid prior for $\mathcal{M}_2$. Since we only assumed that $\mathbf{C}'$ correctly transformed stochastic priors, this proves the second part of the lemma. □

The second part of the lemma essentially says that we get equivalence of GMMs for free if we can prove that the stochastic priors on a pair of machines can be transformed into equivalent pseudo-priors on each other. A corollary of this is that equivalent HMMs are also equivalent GMMs. This is true because we know that if $\mathcal{M}_2$ and $\mathcal{M}_1$ are HMMs, and $\mathcal{M}_2 \subseteq \mathcal{M}_1$, then we can transform *stochastic* priors on $\mathcal{M}_2$ into equivalent priors on $\mathcal{M}_1$ using the transformation matrix $\mathbf{C}$ of Lemma 2.1. Therefore, Lemma 3.1 tells us that $\mathbf{C}$ also transforms *all* valid priors on $\mathcal{M}_2$ into valid priors on $\mathcal{M}_1$, implying that $\mathcal{M}_2 \subseteq \mathcal{M}_1$ even when the models are treated as GMMs.

Finally, we turn our attention to Lemma 2.2 and Theorem 2.3 which proved necessary and sufficient conditions for the equivalence of HMMs. Using Lemma 3.1, and our earlier discussion of the suffix matrix for GMMs, we can see that these results can be applied directly in the GMM context. We would simply need to require that the transformation matrix $\mathbf{C}$ they invoked be pseudo-stochastic instead of stochastic. Having convinced ourselves that all the results characterizing equivalence of HMMs carry over to GMMs also, we see that the algorithms developed in Chapter 2 can be applied to GMMs also. We only need to modify the algorithm for checking equivalence of un-initialized HMMs by relaxing the stochasticity requirement on the transformation matrix $\mathbf{C}$ that it solves for. This actually makes the algorithm more efficient since we now only need to solve a system of linear *equalities* rather than *inequalitites*. (We no longer need the constraint that the entries of $\mathbf{C}$ should be non-negative.) Standard methods for solving systems of linear equalities

run in $O(m^3 + m^2 n)$ time where n is the number of variables and m is the number of equations.[press90] Repeating the analysis of the algorithm for determining equivalence of HMMs, we find that, in the worst case, we will need to solve $n_1 n_2$ equations in $n_1 n_2$ variables, subject to n_2 pseudo-stochasticity constraints. This would take time $O\left[(n_1 n_2)^3 + (n_1 n_2)^2 n_1 n_2\right] = O\left[(n_1 n_2)^3\right]$. We conclude that our algorithm for deciding $\mathcal{M}_2 \subseteq \mathcal{M}_1$, where $\mathcal{M}_2$ and $\mathcal{M}_1$ are GMMs, has a worst-case running time of $O\left(n_1 k(n_1^2 + n_2^2 + 2n_1 n_2) + (n_1 n_2)^3\right)$. This is somewhat better than the running time achieved in the context of HMMs. As before $\mathcal{M}_1 \Leftrightarrow \mathcal{M}_2$ is decided by checking that $\mathcal{M}_2 \subseteq \mathcal{M}_1$ and $\mathcal{M}_1 \subseteq \mathcal{M}_2$.

Our discussions of Generalized Markov Models have swept an important issue under a definitional rug. Our formulation of GMMs is not satisfactory since it does not characterize what makes a given pseudo-stochastic vector valid for a given model. Consequently, the definition is not clear about exactly what forms the transition matrix $\mathbf{A}$ is allowed to take. Since this thesis only compares GMMs with each other, this does not become a difficulty for us - we will always work with models that are presumed to be well-defined. (Obviously, some such models exist since HMMs are themselves GMMs with priors restricted to be stochastic.) However, if we want to build GMMs for practical applications we must have a more constructive method of evaluating the validity of pseudo-stochastic vectors for a given model. At least partly because of the non-constructive definition of GMMs, we have not discussed the issue of parameter-estimation and training of these models from data. However, even without properly understanding the nature of valid vectors for GMMs, we can make progress towards developing training algorithms. Some relevant ideas will be presented in the next chapter.

3.2 Canonical Dimensions and Forms

We will now define the *canonical dimension* of a GMM. This will be a measure that characterizes the essential degree of freedom available in the model. As decribed above, we will freely borrow from the notation defined in Chapters 1 and 2 to manipulate HMMs. Distributions over the outputs of the model will remain stochastic. However, "distributions" over the states of the model will be pseudo-stochastic. We will now use the suffix matrix (Definition 2.5) to define the canonical dimension of a Generalized Markov Model.

Definition 3.3 *Canonical Dimension*
Let $\mathcal{M}$ be a Generalized Markov Model with suffix matrices $\Sigma(x)$ for every $x \in \mathcal{O}^* \cup \{\epsilon\}$ as in Definition 2.5. Also let $\mathcal{U} = \{\vec{\sigma}(y) : y \in \mathcal{O}^*\}$ be the set of all rows of the suffix matrices of $\mathcal{M}$ as in Lemma 2.2. We define the canonical dimension of $\mathcal{M}$ ($d_{\mathcal{M}}$) to be the dimension of the space spanned by the vectors in $\mathcal{U}$. In other words, $d_{\mathcal{M}} = \dim[Span(\mathcal{U})]$.

In order to understand the meaning of the canonical dimension of a model, remember that if $\vec{\sigma}(x) \in \mathcal{U}$, then the j^{th} component of $\vec{\sigma}(x)$ is the probability that the model starts in state s_j, and emits the string x. So, in some sense, the canonical dimension of a model captures the maximal degree of freedom we have to define different stochastic processes by setting up different valid prior distributions. Our definition is also motivated by the following easy result that equivalent GMMs must have the same canonical dimension.

Theorem 3.1 *Invariance of Canonical Dimensions*
Let $\mathcal{M}_1$ be a GMM with n_1 states and canonical dimension d_1. Let $\mathcal{M}_2$ be any GMM with n_2 states that is equivalent to $\mathcal{M}_1$. Let d_2 denote the canonical dimension of $\mathcal{M}_2$. Then it must be the case that $d_2 = d_1$ and $n_2 \geq d_1$.

Proof: If $\mathcal{M}_1 \Leftrightarrow \mathcal{M}_2$, then $\mathcal{M}_1 \subseteq \mathcal{M}_2$ and $\mathcal{M}_2 \subseteq \mathcal{M}_1$. Suppose then that $\mathcal{M}_2 \subseteq \mathcal{M}_1$. Then, using Lemma 3.1, and we can write:

$$\forall x \in \mathcal{O}^* \cup \{\epsilon\} : \quad \vec{\sigma}_1(x)\mathbf{C} = \vec{\sigma}_2(x) \tag{3.2}$$

But we can expand $\vec{\sigma}_1(x)$ in terms of a basis $\{\vec{\sigma}_1(x_i)\}$ for $\mathcal{U}$, the span of $\{\vec{\sigma}_1(x)\}$, to write:

$$\begin{aligned} \vec{\sigma}_1(x)\mathbf{C} &= \sum_i^{d_1} b_i \vec{\sigma}_1(x_i)\mathbf{C} && (3.3) \\ &= \sum_i^{d_1} b_i \vec{\sigma}_2(x_i) && (3.4) \\ \Longrightarrow \vec{\sigma}_2(x) &= \sum_i^{d_1} b_i \vec{\sigma}_2(x_i) && (3.5) \end{aligned}$$

Equation 3.5 shows that the collection of vectors $\{\vec{\sigma}_2(x_i)\}$ forms a basis for the span of $\mathcal{U}_2$ so that $d_2 \leq |\{\vec{\sigma}_2(x_i)\}| = |\{\vec{\sigma}_1(x_i)\}| = d_1$. Similarly, since $\mathcal{M}_1 \subseteq \mathcal{M}_2$ also, we can say that $d_1 \leq d_2$ giving us the result that $d_1 = d_2$. Finally, notice that the canonical dimension of a model $\mathcal{M}$ with n states must be less than or equal to n, since the $\vec{\sigma}(x)$ vectors for $\mathcal{M}$ will have only n components. So, if $\mathcal{M}_2$ is equivalent to $\mathcal{M}_1$, $n_2 \geq d_2 = d_1$. □

Theorem 3.1 tells us that we cannot build a GMM equivalent to $\mathcal{M}_1$ with less than d_1 states. Next we want to show that if $\mathcal{M}_1$has canonical dimension d_1 and n_1 states, where $n_1 > d_1$, then we can effectively construct an equivalent model $\mathcal{M}'$ with only d_1 states. We will prove this by first demonstrating how a particular special type of GMM can be reduced. We will then reduce every GMM to this special form, thereby proving the desired result.

Lemma 3.2 *Reduction of a Special Form*

Let $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$ be a GMM with n states. Let $\mathcal{I}$ be the largest subspace of the

null-space of $\mathbf{B}$ *that is invariant under each of the transition operators* $\mathbf{T}(o_k)$*. Also let* $\vec{B}_i$ *and* $\vec{T}_i(x)$ *denote the* i^{th} *columns of* $\mathbf{B}$ *and* $\mathbf{T}(x)$ *respectively. Suppose that there is a collection of coefficients* $\{f_{ij}\}$ *and an index* α*,* $1 \leq \alpha < n$ *such that:*

$$\forall\, l \leq \alpha \quad : \quad \vec{B}_l = \sum_{j=\alpha+1}^{n} f_{jl} \vec{B}_j \tag{3.6}$$

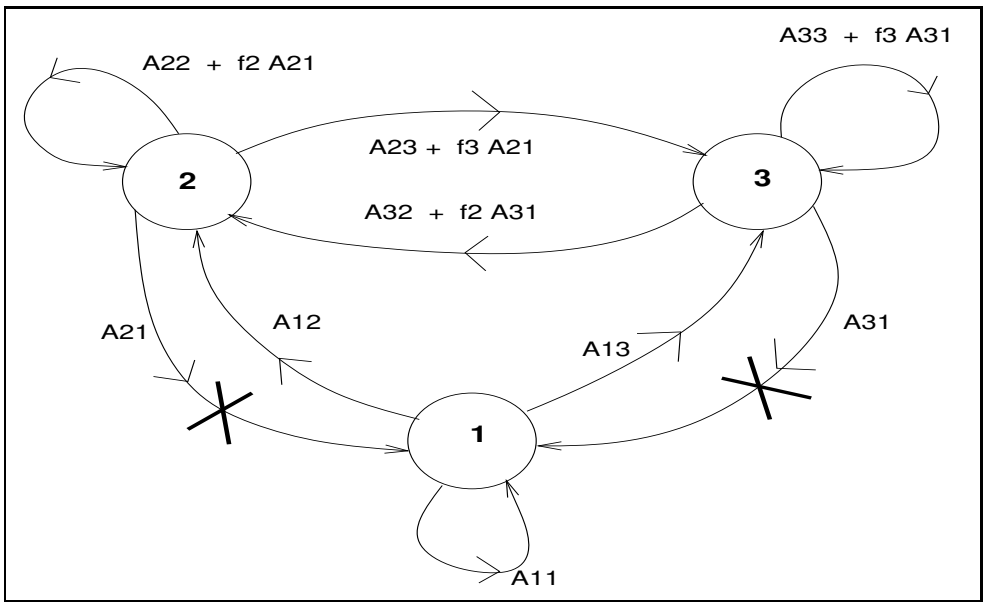
$$\forall o_k \in \mathcal{O} \text{ and } \forall\, l \leq \alpha \quad : \quad \vec{T}_l(o_k) = \sum_{j=\alpha+1}^{n} f_{jl} \vec{T}_j(o_k) + \vec{\Delta}_l(o_k) \tag{3.7}$$

where $\vec{\Delta}_l(o_k) \in \mathcal{I}$*. We will call the states* $\{s_1, s_2, \cdots s_\alpha\}$ *the* **dependent** *states of* $\mathcal{M}$*, and* $\{s_{\alpha+1}, s_{\alpha+2}, \cdots s_n\}$ *the* **independent** *states of* $\mathcal{M}$*. We can build a model* $\mathcal{M}' = (\mathcal{S}', \mathcal{O}, \mathbf{A}', \mathbf{B}')$ *with* $n' = n - \alpha$ *states, such that* $\mathcal{M}' \Leftrightarrow \mathcal{M}$*, and* $\mathcal{S}'$ *contains only the independent states of* $\mathcal{M}$*.*

Prior to proving the lemma it will help to have some intuitions for why it should be true. The lemma basically says that a model can be reduced to a smaller size if the output distributions are linearly dependent and the corresponding columns of every $\mathbf{T}(o_k)$ are dependent with the same coefficients. The basic idea of the proof is to realize that passing through one of the states s_l for $l \leq \alpha$ is indistinguishable from passing through the states s_m for $m > \alpha$ with pseudo-probabilities weighted according to the appropriate linear dependency coefficients.[3] (See Figure 3.2) We can use this observation to redistribute the priors and the outgoing probabilties from each state in such a way that the linearly dependent states are never visited and can be thrown away. The proof below is simply a formalization of this idea.

Proof: In the following discussion we will adopt the convention that variables indexing the states of $\mathcal{M}'$ will range over $\alpha + 1$ to n. Our proof will proceed in five steps. First we will define $\mathbf{B}'$ and $\mathbf{A}'$. In the second step we will prove an useful

[3]This is true up to the vector $\vec{\Delta}_l(o_i)$. However, $\vec{\Delta}_l(o_i)$ lies in an invariant subspace of the null-space of $\mathbf{B}$. Consequently, it never contributes to distributions over the outputs, and can be ignored.



The figure shows an HMM for which $\vec{B}_1 = f_2\vec{B}_2 + f_3\vec{B}_3$ and the $\mathbf{T}(o_k)$ satisfy Equation 3.7. (We have suppressed the output distributions in the figure.) In order to remove the dependent state s_1, we excise the transitions to s_1 and add them to the transitions between the independent states weighted appropriately by f_2 and f_3. The priors are redistributed in the same way. If we do this, observe that s_1 is never visited and can be thrown away.

Figure 3-1: Reduction of A Special Form

invariance property of $\mathbf{A}'$. Next we will define a pseudo-stochastic transformation of the priors on $\mathcal{M}$ into priors on $\mathcal{M}'$. Then we will use the invariance property of $\mathbf{A}'$ to show that $\mathcal{M} \subseteq \mathcal{M}'$. Finally, we will demonstrate that $\mathcal{M}' \subseteq \mathcal{M}$. We will find it convenient to define the following matrix:

$$\mathbf{F} = \begin{bmatrix} f_{(\alpha+1)1} & f_{(\alpha+1)2} & \cdots & f_{(\alpha+1)\alpha} \\ f_{(\alpha+2)1} & f_{(\alpha+2)2} & \cdots & f_{(\alpha+2)\alpha} \\ \vdots & \vdots & \vdots & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{n\alpha} \end{bmatrix} \tag{3.8}$$

So $\mathbf{F}$ is an $n' \times \alpha$ matrix whose components are the expansion coefficients assumed in the lemma. Note that $\mathbf{F}$ must be pseudo-stochastic, since all the vectors on both sides of Equation 3.6 are stochastic and therefore sum to one. We are now ready to construct the reduced model $\mathcal{M}'$.

First of all, we will take the new output matrix $\mathbf{B}'$ to simply be the last $n' = n - \alpha$ columns of $\mathbf{B}$. Our earlier intutitions concerning $\mathbf{A}'$ said that the transitions to dependent states should be redistributed according to the weights of the expansion coefficients. Putting this idea into symbols gives:

$$\mathbf{A}'_{ij} = \mathbf{A}_{ij} + \sum_{l=1}^{\alpha} \mathbf{A}_{lj} f_{il} \tag{3.9}$$

We can use the $\mathbf{F}$ matrix defined earlier to compactly write down the relationship between $\mathbf{A}$, $\mathbf{A}'$, $\mathbf{B}$ and $\mathbf{B}'$:

$$\mathbf{A}' = [\mathbf{F}|I_{n' \times n'}]\, \mathbf{A} \begin{bmatrix} 0_{\alpha \times n'} \\ I_{n' \times n'} \end{bmatrix} \tag{3.10}$$

$$\mathbf{B} = \mathbf{B}' [\mathbf{F}|I_{n' \times n'}] \tag{3.11}$$

$I_{n' \times n'}$ is the n' by n' identity matrix and $[\mathbf{F}|I_{n' \times n'}]$ is the matrix consisting of $\mathbf{F}$ and $I_{n' \times n'}$ concatenated together. $0_{\alpha \times n'}$ is the $\alpha \times n'$ zero matrix. Now suppose that $\vec{P}(t, x_{t-1})$ is a vector such that $P_i(t, x_{t-1})$ is the pseudo-probability that the model $\mathcal{M}$ emits the string x_{t-1} and then enters the state s_i at time t. (This is the pseudo-distribution over states *before* seeing the output at time t.) Then suppose it is also true that:

$$\vec{P}\,'(t, x_{t-1}) = [\mathbf{F}|I_{n' \times n'}] \left(\vec{P}(t, x_{t-1}) + \vec{\delta} \right) \tag{3.12}$$

where $\vec{\delta}$ is a vector that lies in $\mathcal{I}$. We claim that if Equation 3.12 holds, then the joint probability of the output at time t and x_{t-1} is the same for $\mathcal{M}$ and $\mathcal{M}'$. Furthermore, regardless of the output at time t it will be true that $\vec{P}\,'(t+1, x_t) = [\mathbf{F}|I_{n' \times n'}] \left(\vec{P}(t+1, x_t) + \vec{\delta}' \right)$, where $\vec{\delta}'$ is a vector lying in $\mathcal{I}$, the invariant subsapce of

the null-space of $\mathbf{B}$. We can prove the first part of the claim by observing that:

$$\begin{aligned}
\mathbf{B}'\vec{P}\,'(t, x_{t-1}) &= \mathbf{B}'[\mathbf{F}|I_{n'\times n'}]\left(\vec{P}(t, x_{t-1}) + \vec{\delta}\right) \\
&= \mathbf{B}\left(\vec{P}(t, x_{t-1}) + \vec{\delta}\right) \\
&= \mathbf{B}\vec{P}(t, x_{t-1}) \qquad (3.13)
\end{aligned}$$

The last equation follows because $\vec{\delta}$ is in the null-space of $\mathbf{B}$. In order to prove the second part of the claim we assume without loss of generality that $o(t) = o_k$ and evolve the model forward in time. In order to do this, note that the transition operator $\mathbf{T}'(o_k)$ can be written as:

$$\begin{aligned}
\mathbf{T}'(o_k) &= \mathbf{A}'\mathbf{B}'_k \qquad (3.14) \\
&= [\mathbf{F}|I_{n'\times n'}]\,\mathbf{A}\begin{bmatrix} 0_{\alpha\times n'} \\ I_{n'\times n'} \end{bmatrix}[0_{n'\times\alpha}|I_{n'\times n'}]\,\mathbf{B}_k\begin{bmatrix} 0_{\alpha\times n'} \\ I_{n'\times n'} \end{bmatrix} \qquad (3.15) \\
&= [\mathbf{F}|I_{n'\times n'}]\,\mathbf{A}\left[\begin{array}{c|c} 0_{\alpha\times n} & \\ \hline 0_{n'\times\alpha} & I_{n'\times n'} \end{array}\right]\mathbf{B}_k\begin{bmatrix} 0_{\alpha\times n'} \\ I_{n'\times n'} \end{bmatrix} \qquad (3.16)
\end{aligned}$$

where we have used the fact that $\mathbf{B}'_k$ consists of the last n' rows and columns of $\mathbf{B}_k$. We can simplify this a little further by using the notation $\vec{A}_i = i^{th}$ column of $\mathbf{A}$ to write:

$$\begin{aligned}
\mathbf{A}\left[\begin{array}{c|c} 0_{\alpha\times n} & \\ \hline 0_{n'\times\alpha} & I_{n'\times n'} \end{array}\right]\mathbf{B}_k &= \left[0_{n\times\alpha}|\vec{A}_{\alpha+1}|\vec{A}_{\alpha+2}|\cdots|\vec{A}_n\right]\mathbf{B}_k \qquad (3.17) \\
&= \left[0_{n\times\alpha}|\vec{T}_{\alpha+1}(o_k)|\vec{T}_{\alpha+2}(o_k)|\cdots|\vec{T}_n(o_k)\right] \qquad (3.18)
\end{aligned}$$

Using this we can conveniently compute $\vec{P}\,'(t+1, x_t)$ as shown below. We will let $\vec{\delta}$ and $\vec{\delta}\,'$ denote vectors in $\mathcal{I}$, the invariant subspace of the null-space of $\mathbf{B}$. For compactness

of the equations we will also write $T_\alpha(o_k)$ for $\left[0_{n\times\alpha}|\vec{T}_{\alpha+1}(o_k)|\vec{T}_{\alpha+2}(o_k)|\cdots|\vec{T}_n(o_k)\right]$.

$$\begin{aligned}
\vec{P}\,'(t+1,x_t) &= \mathbf{T}'(o_k)\vec{P}\,'(t,x_{t-1}) && (3.19)\\
&= [\mathbf{F}|I_{n'\times n'}]\,T_\alpha(o_k)\begin{bmatrix} 0_{\alpha\times n'} \\ I_{n'\times n'}\end{bmatrix}[\mathbf{F}|I_{n'\times n'}]\left(\vec{P}(t,x_{t-1})+\vec{\delta}\right) && (3.20)\\
&= [\mathbf{F}|I_{n'\times n'}]\,T_\alpha(o_k)\left[\begin{array}{c|c} 0_{\alpha\times n} & \\ \hline \mathbf{F} & I_{n'\times n'}\end{array}\right]\left(\vec{P}(t,x_{t-1})+\vec{\delta}\right) && (3.21)
\end{aligned}$$

We can now use the fact that the columns of $\mathbf{T}(o_k)$ are linearly dependent according to Equation 3.7 to write:

$$\begin{aligned}
T_\alpha(o_k)\left[\begin{array}{c|c} 0_{\alpha\times n} & \\ \hline \mathbf{F} & I_{n'\times n'}\end{array}\right] &= \left[0_{n\times\alpha}|\vec{T}_{\alpha+1}(o_k)|\vec{T}_{\alpha+2}(o_k)|\cdots|\vec{T}_n(o_k)\right]\left[\begin{array}{c|c} 0_{\alpha\times n} & \\ \hline \mathbf{F} & I_{n'\times n'}\end{array}\right]\\
&= \mathbf{T}(o_k)+\left[\vec{\Delta}_1(o_k)|\vec{\Delta}_2(o_k)|\cdots|\vec{\Delta}_\alpha(o_k)|0_{n\times n'}\right] && (3.22)\\
&= \mathbf{T}(o_k)+\Delta && (3.23)
\end{aligned}$$

where we have set $\Delta = \left[\vec{\Delta}_1(o_k)|\vec{\Delta}_2(o_k)|\cdots|\vec{\Delta}_\alpha(o_k)|0_{n\times n'}\right]$. Observe that for any vector $\vec{x}$ of appropriate dimension, $\Delta\vec{x}\in\mathcal{I}$ since every column of Δ is an element of $\mathcal{I}$, the invariant null-space of $\mathbf{B}$. Therefore, plugging Equation 3.23 into Equation 3.21, we find that:

$$\begin{aligned}
\vec{P}\,'(t+1,x_t) &= [\mathbf{F}|I_{n'\times n'}]\left(\mathbf{T}(o_k)\vec{P}(t,x_{t-1})+\vec{\delta}'\right) && (3.24)\\
&= [\mathbf{F}|I_{n'\times n'}]\left(\vec{P}(t+1,x_t)+\vec{\delta}'\right) && (3.25)
\end{aligned}$$

where $\vec{\delta}\,'$ is some vector in $\mathcal{I}$.[4] Equation 3.25 shows us that if the pseudo-probabilities on $\mathcal{M}'$ satisfy Equation 3.12 at time t, they do so also at time $t+1$ and, by induction on t, for all future times. This invariance property of $\mathbf{A}$ will be useful shortly in

[4] We get Equation 3.24 by using the facts that $\mathbf{T}(o_k)\vec{\delta}\in\mathcal{I}$ since $\vec{\delta}\in\mathcal{I}$, and $\Delta\vec{x}\in\mathcal{I}$ for any $\vec{x}$ as discussed before.

proving that $\mathcal{M}'$ and $\mathcal{M}$ are equivalent.

We are finally in a position to show that $\mathcal{M}' \subseteq \mathcal{M}$. If $\vec{p}$ is a prior on $\mathcal{M}$, let $\vec{p}\,' = [\mathbf{F}|I_{n'\times n'}]\,\vec{p}$ be the corresponding prior on $\mathcal{M}'$. These prior distributions cause Equation 3.12 to be satisfied for $t = 0$ and $x = \epsilon$. Therefore, by our earlier discussion, Equation 3.12 is satisfied for all times t and strings x_{t-1}. We also showed that if Equation 3.12 is satisfied, then the two models have the same probabilities of producing the various outputs. Hence, we can conclude that $\mathcal{M}'(\vec{p}\,') \Leftrightarrow \mathcal{M}(\vec{p})$. Since $[\mathbf{F}|I_{n'\times n'}]$ is a pseudo-stochastic transformation of priors on $\mathcal{M}$ into equivalent priors on $\mathcal{M}'$, we know that $\mathcal{M} \subseteq \mathcal{M}'$. To show that $\mathcal{M}' \subseteq \mathcal{M}$, we will first show that every stochastic prior on $\mathcal{M}'$ can be transformed into an equivalent valid prior on $\mathcal{M}$. Lemma 3.1 will then show that $\mathcal{M}' \subseteq \mathcal{M}$. So suppose that $\vec{q}\,'$ is a stochastic prior on $\mathcal{M}'$. Then, construct a prior $\vec{q}$ on $\mathcal{M}$ such that:

$$\vec{q} = \begin{bmatrix} 0_{\alpha\times n'} \\ I_{n'\times n'} \end{bmatrix} \vec{q}\,' = (\vec{0}_\alpha, \vec{q}) \tag{3.26}$$

where $\vec{0}_\alpha$ is the $\alpha-$dimensional zero vector. We can see at once that $\vec{q}\,' = [\mathbf{F}|I_{n'\times n'}]\,\vec{q}$. Therefore, our earlier discussion shows that $\mathcal{M}'(\vec{q}\,') \Leftrightarrow \mathcal{M}(\vec{q})$. So Equation 3.26 defines a pseudo-stochastic transformation of stochastic priors on $\mathcal{M}'$ into equivalent valid priors on $\mathcal{M}$. By Lemma 3.1 we can conclude that $\mathcal{M} \subseteq \mathcal{M}'$. Putting everything together we finally reach the desired conclusion that $\mathcal{M}' \Leftrightarrow \mathcal{M}$. □

All that remains in our quest to find minimal representations for GMMs is a way of transforming all reducible GMMs into the special form that was reduced in Lemma 3.2. We will now prove a theorem that shows that all reducible GMMs are *already* is the special form of Lemma 3.2. This is then used to reduce GMMs to their minimal equivalent representations.

Theorem 3.2 *Reduction of GMMs to Minimal Representations*

Let $\mathcal{M}_1 = (\mathcal{S}_1, \mathcal{O}, \mathbf{A}_1, \mathbf{B}_1)$ *be a GMM with* n_1 *states and canonical dimension* $d_1 < n_1$.

Then $\mathcal{M}_1$ can be reduced to a minimal equivalent model $\mathcal{M}^*$ with only d_1 states. If a model has only as many states as its canonical dimension, we will call it a **minimal representation** *for its equivalence class.*

Proof: We defined the canonical dimension of $\mathcal{M}_1$ to be the dimension of the span of $\mathcal{U}_1 = \{\vec{\sigma}_1(y) : y \in \mathcal{O}^*\}$, where the $\vec{\sigma}_1(y)$ are rows of the suffix matrices of $\mathcal{M}$. Let $\mathcal{V} = \{\vec{\sigma}_1(x_1), \vec{\sigma}_1(x_2), \cdots, \vec{\sigma}_1(x_{d_\mathcal{M}})\}$ be a collection of vectors in $\mathcal{U}_1$ that forms a basis for $Span(\mathcal{U}_1)$. Then consider a matrix $\mathbf{G}$ whose rows are the elements of $\mathcal{V}$. We can write:

$$\mathbf{G} = \begin{bmatrix} \vec{\sigma}_1(x_1) \\ \vec{\sigma}_1(x_2) \\ \vdots \\ \vec{\sigma}_1(x_{d_1}) \end{bmatrix} = \left[\begin{array}{c|c|c|c} \vec{g}_1 & \vec{g}_2 & \cdots & \vec{g}_{n_1} \end{array} \right] \tag{3.27}$$

(In this equation the vectors $\vec{g}_i$ represent the columns of $\mathbf{G}$.) $\mathbf{G}$ is a $d_1 \times n_1$ matrix whose rows are linearly independent. So, it has a row-rank d_1 and this means that its column rank is also d_1. So, there are only d_1 independent columns in $\mathbf{G}$. Assume, without loss of generality, that the last d_1 columns of $\mathbf{G}$ are the independent columns and let $\alpha = n_1 - d_1$. There must be a set of coefficients $\{f_{jl}\}$ such that we can write:

$$\forall\, l \leq \alpha : \quad \vec{g}_l = \sum_{j=\alpha+1}^{n_1} f_{jl} \vec{g}_j \tag{3.28}$$

We are going to use this fact to show $\mathcal{M}_1$ already satisifies the conditions of Lemma 3.2 and can therefore be reduced to a smaller size. In order to do this we will find it convenient to introduce the following matrix:

$$\mathbf{F} = \begin{bmatrix} f_{(\alpha+1)1} & f_{(\alpha+1)2} & \cdots & f_{(\alpha+1)\alpha} \\ f_{(\alpha+2)1} & f_{(\alpha+2)2} & \cdots & f_{(\alpha+2)\alpha} \\ \vdots & \vdots & \vdots & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{n\alpha} \end{bmatrix} \tag{3.29}$$

This matrix is formally the same as the **F** matrix used in the proof of the special case reduction lemma. We will see that the similarity is not coincidental. We can use the **F** matrix to rewrite Equation 3.28 more compactly as follows:

$$\mathbf{G}\begin{bmatrix} -I_{\alpha\times\alpha} \\ \mathbf{F} \end{bmatrix} = 0 \tag{3.30}$$

Remember now that every row of every suffix matrix $\Sigma_1(x)$ can be written as a linear combination of the rows of **G**. This implies that corresponding to every matrix $\Sigma_1(x)$, there is another matrix $\mathbf{S}(x)$ such that $\Sigma_1(x) = \mathbf{S}(x)\mathbf{G}$. (The i^{th} row of $\mathbf{S}(x)$ contains the coefficients expressing the i^{th} row of $\Sigma_1(x)$ as a linear combination of the rows of **G**.) Using this we find that:

$$\forall x \in \mathcal{O}^* \cup \{\epsilon\} : \quad \Sigma_1(x)\begin{bmatrix} -I_{\alpha\times\alpha} \\ \mathbf{F} \end{bmatrix} = \mathbf{S}(x)\mathbf{G}\begin{bmatrix} -I_{\alpha\times\alpha} \\ \mathbf{F} \end{bmatrix} = 0 \tag{3.31}$$

By picking $x = \epsilon$ so that $\Sigma_1(x) = \mathbf{B}_1$, and expanding the matrix notation into a summation, we find that:

$$\forall\, l \leq \alpha : \quad \vec{B}_l = \sum_{j=\alpha+1}^{n_1} f_{jl}\vec{B}_j \tag{3.32}$$

where $\vec{B}_i$ is the i^{th} column of **B**.one Notice that this is exactly the first condition we need in order to apply our earlier lemma on reduction of certain special types of GMMs. Next, for notational convenience, we define $\Delta(o_k)$ such that:

$$\Delta(o_k) = \mathbf{T}_1(o_k)\begin{bmatrix} -I_{\alpha\times\alpha} \\ \mathbf{F} \end{bmatrix} \tag{3.33}$$

We will refer to the i^{th} columns of Δo_k and $\mathbf{T}_1(o_k)$ as $\vec{\Delta}_i(o_k)$ and $\vec{T}_i(o_k)$ respectively. Then, for any string $y = o_k x$ which starts with the output o_k we can write

Equation 3.31 as:

$$\Sigma_1(y)\begin{bmatrix} -I_{\alpha\times\alpha} \\ \mathbf{F} \end{bmatrix} = \mathbf{B}_1\mathbf{T}_1(y)\mathbf{T}_1(o_k)\begin{bmatrix} -I_{\alpha\times\alpha} \\ \mathbf{F} \end{bmatrix} = \mathbf{B}_1\mathbf{T}_1(x)\Delta(o_k) = 0 \tag{3.34}$$

Since this equality holds for every string $x \in \mathcal{O}^* \cup \{\epsilon\}$, we can conclude that the columns of $\Delta(o_k)$ are elements of $\mathcal{I}$, the invariant null-space of $\mathbf{B}_1$. By expanding the definition of $\Delta(o_k)$ we then find that:

$$\forall o_k \in \mathcal{O} \; and \; \forall \, l \leq \alpha : \;\; \vec{T}_l(o_k) = \sum_{j=\alpha+1}^{n} f_{jl}\vec{T}_j(o_k) + \vec{\Delta}_l(o_k) \tag{3.35}$$

where the $\vec{\Delta}_l(o_k) \in \mathcal{I}$ are the columns of $\Delta(o_k)$. Now Equations 3.32 and 3.35 are exactly the conditions that make Lemma 3.2 true. Consequently, any GMM with canonical dimension d_1, has only d_1 independent states. The method outlined in the proof of Lemma 3.2 can then be used to reduce $\mathcal{M}_1$ to an model $\mathcal{M}^*$ with only d_1 states. Since Theorem 3.1 tells us that no smaller model can be equivalent to $\mathcal{M}$, $\mathcal{M}^*$ is a *minimal* representation of $\mathcal{M}$. □

Theorem 3.2 shows how a GMM can be reduced to a minimal representation. We will discuss how this result applies to Hidden Markov Models in Section 3.2.1. In addition to finding minimal models, we also want our representations to be "canonical" in the sense that they are essentially unique. Next, we will prove two theorems that provide a deeper understanding of the essential reasons for reducibility of GMMs, and characterize the relationship between equivalent minimal representations of a given model $\mathcal{M}$.

Theorem 3.3 *Geometric Characterization of Minimal Represenations*

As before, we will call a model a **minimal representation** *if it is the smallest model in its equivalence class. A model is minimal if and only if its invariant null-*

space $\mathcal{I}$ consists of only the zero vector. Hence, priors are equivalent for a minimal representation only if they are equal to each other.

Proof: Let $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$ be a Generalized Markov Model with n states. We remind the reader that the invariant null-space $\mathcal{I}$ is the largest subspace of the null-space of the output matrix $\mathbf{B}$, that is invariant under the action of every transition operator $\mathbf{T}(o_k)$. Suppose, first of all, that $\mathcal{M}$ is a minimal representation. Suppose also that there is a vector $\vec{\delta} \in \mathcal{I}$ which has some non-zero components. By definition of being an element of $\mathcal{I}$ we can write:

$$\forall x \in \mathcal{O}^* \cup \{\epsilon\} : \quad \mathbf{B}\mathbf{T}(x)\vec{\delta} = \Sigma(x)\vec{\delta} = 0 \tag{3.36}$$

By picking $x = \epsilon$ and $x = o_k y$ where y is any string we can write:

$$\mathbf{B}\vec{\delta} = 0 \tag{3.37}$$

$$\forall y \in \mathcal{O}^* \cup \{\epsilon\} : \quad \mathbf{B}\mathbf{T}(y)\left[\mathbf{T}(o_k)\vec{\delta}\right] = \mathbf{B}\mathbf{T}(y)\vec{\Delta}(o_k) = 0 \tag{3.38}$$

where we have written $\vec{\Delta}(o_k)$ for $\mathbf{T}(o_k)\vec{\delta}$. The second equation says that $\vec{\Delta}(o_k) \in \mathcal{I}$, the invariant null-space of $\mathbf{B}$. Writing this out as an equation for the columns of $\mathbf{B}$ and $\mathbf{T}(o_k)$, and assuming, without loss of generality, that $\delta_1 \neq 0$, we find that:

$$\vec{B}_1 = \sum_{j=2}^{n} -\frac{\delta_j}{\delta_1}\vec{B}_j \tag{3.39}$$

$$\forall o_k \in \mathcal{O} \;\; \vec{T}_1(o_k) = \sum_{j=2}^{n} -\frac{\delta_j}{\delta_1}\vec{T}_j(o_k) + \vec{\Delta}(o_k) \tag{3.40}$$

(As before, we are writing $\vec{B}_i$ and $\vec{T}_i(o_k)$ for the the i^{th} columns of $\mathbf{B}$ and $\mathbf{T}(o_k)$ respectively.) But this means that s_1 is a dependent state, in the sense of Lemma 3.2, and can be reduced away. This contradicts the assumed minimality of the model. So we see that if $\mathcal{M}$ is a minimal representation, then $\mathcal{I}$ can consist only of the zero vector. Next we will prove that if $\mathcal{I} = \{\vec{0}\}$, then the model is necessarily minimal. So

assume that $\mathcal{I} = \{\vec{0}\}$. Then suppose that $\mathcal{M}$ is not minimal and therefore $n > d_{\mathcal{M}}$, where $d_{\mathcal{M}}$ is the canonical dimension of $\mathcal{M}$. Theorem 3.2 then tells us that there is a collection of coefficients $\{f_{ij}\}$, not all of which are zero, such that:

$$\forall x \in \mathcal{O}^* \cup \{\epsilon\} : \quad \Sigma_1(x) \begin{bmatrix} -I_{\alpha\times\alpha} \\ \mathbf{F} \end{bmatrix} = 0 \tag{3.41}$$

where $\mathbf{F}$ is defined by Equation 3.29. Let $\vec{\delta}$ be any column of the matrix to the right of $\Sigma(x)$ in Equation 3.41. Then $\vec{\delta}$ is a vector with some non-zero components that lies in $\mathcal{I}$.[5] This contradicts our assumption about $\mathcal{I}$, telling us that if $\mathcal{I}$ consists only of the zero vector, the model cannot have more states than the canonical dimension, and is, therefore, minimal. So we have proved that for a model $\mathcal{M}$ to be a minimal representation, it is necessary and sufficient that its invariant null-space consists only of the zero vector. Observe that, according to the GMM version of Theorem 2.1, this implies that equivalent priors on a minimal representation are equal to each other. □

Theorems 3.1 and 3.2 told us that this minimal model has exactly as many states as its canonical dimension. The result proven just above showed that a minimal model can be characterized geometrically as having an invariant null-space consisting only of the zero vector. Furthermore, the invariant null-space of a model with n states has a dimension $n - d_{\mathcal{M}}$ where $d_{\mathcal{M}}$ is the canonical dimension of the model. One consequence of this is that no two unequal priors on a minimal model are equivalent. In other words, equivalence of priors on a minimal model implies equality of priors. This tells us that the minimal representation indeed removes every last shred of redundancy available in a model. Every stochastic process that can be modelled by setting the priors on the machine is represented precisely once, by a distinct prior. We could use this to build an algorithm to reduce a model to its minimal representation. First of all,

[5] This is so because for every string x we know that $\Sigma(x) = \mathbf{BT}(x)\vec{\delta} = 0$.

we would find the invariant null-space $\mathcal{I}$ via standard methods for decomposing vector spaces based on their invariance properties under different operators. Then we would find a basis for $\mathcal{I}$, and use the basis vectors as shown in the proof of Theorem 3.3 as the linear dependency coefficients required by the reduction lemma. However, we can build a cleaner algorithm directly from Theorem 3.2. We will do this after proving one more theorem which characterizes the relationship between equivalent minimal representations in the GMM formalism for a class of stochastic processes.

Theorem 3.4 *Relationship Between Minimal Representations*
Suppose $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathbf{A}, \mathbf{B})$ *and* $\mathcal{M}' = (\mathcal{S}', \mathcal{O}, \mathbf{A}', \mathbf{B}')$ *are two n-state GMMs, both of which are minimal representations of a class of processes with canonical dimension* $d_{\mathcal{M}}$. *Then* $\mathcal{M}$ *and* $\mathcal{M}'$ *are related by a change of basis for the n-dimensional space of vectors over the states.*

Proof: Since $\mathcal{M}$ and $\mathcal{M}'$ are equivalent models, Lemma 3.1 tells us that there are two pseudo-stochastic matrices $\mathbf{C}$ and $\mathbf{C}'$ such that:

$$\forall x \in \mathcal{O}^* \cup \{\epsilon\} : \quad \mathbf{B}\mathbf{T}(x)\mathbf{C} = \mathbf{B}'\mathbf{T}'(x) \tag{3.42}$$

$$\forall x \in \mathcal{O}^* \cup \{\epsilon\} : \quad \mathbf{B}'\mathbf{T}'(x)\mathbf{C}' = \mathbf{B}\mathbf{T}(x) \tag{3.43}$$

Picking $x = \epsilon$, this tells us that $\mathbf{B}\mathbf{C} = \mathbf{B}'$ and $\mathbf{B}'\mathbf{C}' = \mathbf{B}$. Then, substituting Equation 3.43 back into Equation 3.42, and bringing all terms to the right hand side, we find that:

$$\forall x \in \mathcal{O}^* \cup \{\epsilon\} : \quad \mathbf{B}'\mathbf{T}'(x)\,[I - \mathbf{C}'\mathbf{C}] = \Sigma'(x)\,[I_{n\times n} - \mathbf{C}'\mathbf{C}] = 0 \tag{3.44}$$

This means that the corresponding columns of $I_{n\times n}$ and $\mathbf{C}'\mathbf{C}$ are equivalent priors for $\mathcal{M}'$. But we know from Theorem 3.3 that priors on minimal models are equivalent if and only if they are equal. So we conclude that $\mathbf{C}'\mathbf{C} = I_{n\times n}$. Similarly, we find that $\mathbf{C}\mathbf{C}' = I_{n\times n}$, and so we can say that $\mathbf{C}'$ and $\mathbf{C}$ are non-singular matrices and are

inverses of each other.

Now define the terms *state vector space* and *output vector space* to mean the vector spaces associated with distributions over states and outputs respectively. We will show that $\mathcal{M}$ is the same as model $\mathcal{M}'$ specified in a different basis for the state vector space of the model. First all, suppose U is a vector space, and $\mathbf{S}$ is a non-singular transformation matrix describing a change of basis for U. Then the change of basis is described by the following tranformations:

1. Every $\vec{x} \in U$ is transformed to $\mathbf{S}\vec{x}$
2. Every linear operator O which maps U into U is transformed to $\mathbf{S}O\mathbf{S}^{-1}$.
3. Every linear operator P mapping U into any other vector space is transformed to $P\mathbf{S}^{-1}$.

Now let $\mathbf{S} = \mathbf{C}'$ and let $\mathbf{S}^{-1} = \mathbf{C}$. Equation 3.43 tells us that the priors on $\mathcal{M}$ are mapped onto the priors on $\mathcal{M}'$ by $\mathbf{S}$(i.e., $\vec{p}\,' = \mathbf{S}\vec{p}$). We have already observed that $\mathbf{B}' = \mathbf{B}\mathbf{S}^{-1}$. Next, consider the equation $\mathbf{B}\mathbf{T}(y)\mathbf{T}(x)\mathbf{S}^{-1} = \mathbf{B}'\mathbf{T}'(y)\mathbf{T}'(x)$. Substituting for $\mathbf{B}\mathbf{T}(y)$ we find that for every $y \in \mathcal{O}^* \cup \{\epsilon\}$ we can write

$$\mathbf{B}'\mathbf{T}'(y)\mathbf{S}\mathbf{T}(x)\mathbf{S}^{-1} = \mathbf{B}'\mathbf{T}'(y)\mathbf{T}'(x) \tag{3.45}$$

$$\Longrightarrow \quad \mathbf{B}'\mathbf{T}'(y)\left[\mathbf{S}\mathbf{T}(x)\mathbf{S}^{-1} - \mathbf{T}'(x)\right] = 0 \tag{3.46}$$

This implies that the corresponding columns of $\mathbf{S}\mathbf{T}(x)\mathbf{S}^{-1}$ and $\mathbf{T}'(x)$, when appropriately normalized to sum to 1, would be equivalent priors for $\mathcal{M}'$. So, by Theorem 3.3 they are equal to each other and we can write:

$$\mathbf{T}'(x) = \mathbf{S}\mathbf{T}(x)\mathbf{S}^{-1} \tag{3.47}$$

From this we also know that $\Sigma'(x) = \mathbf{B}'\mathbf{T}'(x) = \mathbf{B}\mathbf{S}^{-1}\mathbf{S}\mathbf{T}(x)\mathbf{S}^{-1} = \mathbf{B}\mathbf{T}(x)\mathbf{S}^{-1} =$

$\Sigma(x)\mathbf{S}^{-1}$. Summarizing our conclusions we find that:

$$\vec{p}' = \mathbf{S}\vec{p} \tag{3.48}$$

$$\mathbf{B}' = \mathbf{B}\mathbf{S}^{-1} \tag{3.49}$$

$$\mathbf{T}'(x) = \mathbf{S}\mathbf{T}(x)\mathbf{S}^{-1} \tag{3.50}$$

$$\Sigma'(x) = \Sigma(x)\mathbf{S}^{-1} \tag{3.51}$$

These equations decribe transformations that are formally identical to a basis transformation represented by the matrix $\mathbf{S}$. Furthermore, every quantity used to prove the theorems of this thesis consisted of sums and products of the quantities in Equations 3.48 to 3.51. So we conclude that equivalent minimal representations are related by a basis transformation for the state vector space of the models. □

Theorem 3.4 tells us that the minimal representation obtained in Theorem 3.2 is essentially unique, up to a change of basis for the state vector space. So we have indeed achieved a satisfactory characterization of the degree of expressiveness in a GMM and obtained a minimal, canonical representation for the equivalence classes of GMMs. We will now describe an algorithm that will canonicalize a model by reducing it to its minimal, canonical representation.

Reduction Algorithm: In order to construct an algorithm to canonicalize GMMs we will follow the proof of Theorem 3.2. In order to reduce a model $\mathcal{M}$ to its minimal equivalent form, we need to generate a basis for the span of $\mathcal{U} = \{\vec{\sigma}(x) : x \in \mathcal{O}^*\}$. Using the methods developed in our very first algorithm to check equivalence of prior distributions, we can generate such a basis in $O(n^3k)$ time, where n is the number of states and k is the number of outputs. Then we use standard Gausian elimination to find the linear dependencies amongst the $\vec{g}_i$ vectors defined in Equation 3.27. This will take time $O(n^3 + n^2k)$.[press90] The proof of Theorem 3.2 shows that the coefficients of these linear dependencies are the $\{f_{ij}\}$ required by Lemma 3.2 to reduce the model.

The reduction procedure takes time $O(n^2 + nk)$ since we simply have to set the (at most) $O(n^2 + nk)$ parameters of the reduced model according to the rules specified in Lemma 3.2. Therefore, for large k and n we can reduce a GMM to its minimal, canonical representation in worst-case time $O(n^3 k)$.

3.2.1 Results for HMMs

Hidden Markov Models are derived from the subclass of GMMs with stochastic transition matrices by restricting the priors to also be stochastic. This restriction on the priors makes it a little difficult to compare HMMs directly to GMMs. However, we can make good progress by saying that a GMM $\mathcal{M}$ **contains** an HMM $\mathcal{N}$ if for every stochastic prior $\vec{p}$ on $\mathcal{N}$, we can find an equivalent pseudo-stochastic prior $\vec{q}$ on $\mathcal{M}$. In other other words, $\mathcal{M}$ contains $\mathcal{N}$ if every process that can be modelled by HMM $\mathcal{N}$ can also be modelled by GMM $\mathcal{M}$. Now let $\mathcal{N}_G$ denote the GMM derived by removing the stochasticity restriction on the priors on $\mathcal{N}$. Clearly, if $\mathcal{N}_G \subseteq \mathcal{M}$ then $\mathcal{M}$ contains $\mathcal{N}$, since N_G can model every process modelled by $\mathcal{N}$. By definition of containment, it is also clear that if $\mathcal{M}$ contains $\mathcal{N}$, then all the stochastic priors on $\mathcal{N}_G$ can be mapped to equivalent priors on $\mathcal{M}$. But, by Lemma 3.1 this means that $\mathcal{N}_G \subseteq \mathcal{M}$. So we see that GMM $\mathcal{M}$ contains an HMM $\mathcal{N}$ if and only if $\mathcal{N}_G \subseteq \mathcal{M}$ where $\mathcal{N}_G$ is the GMM derived by removing the stochasticity restriction on the priors on $\mathcal{N}$. We can use this to state the following theorem.

Theorem 3.5 *Minimal Representations of HMMs*

Suppose $\mathcal{N}$ is an HMM and $\mathcal{N}^*$ is the smallest HMM equivalent to $\mathcal{N}$. Let $\mathcal{N}_G$ and $\mathcal{N}_G^*$ denote the GMMs derived by removing the stochasticity constraints on $\mathcal{N}$ and $\mathcal{N}^*$ respectively. Then every GMM $\mathcal{M}$ that contains $\mathcal{N}$ must satisfy $\mathcal{N}_G \subseteq \mathcal{M}$. Furthermore, the minimal HMM $\mathcal{N}^*$ has at least as many states as the smallest GMM equivalent to $\mathcal{N}_G$.

Proof: First of all, suppose that $\mathcal{M}$ is a GMM that contains $\mathcal{N}$. Then we know that the stochastic priors on $\mathcal{N}_G$ can be transformed into equivalent priors on $\mathcal{M}$. By Lemma 3.1 we can then conclude that $\mathcal{N}_G \subseteq \mathcal{M}$. Next, by assumption of $\mathcal{N} \Leftrightarrow \mathcal{N}^*$, the stochastic priors on $\mathcal{N}_G$ and $\mathcal{N}_G^*$ can be transformed onto equivalent priors on each other. Therefore, Lemma 3.1 tells us that $\mathcal{N}_G \Leftrightarrow \mathcal{N}_G^*$ also. Now let $\mathcal{M}^*$ be the smallest GMM equivalent to $\mathcal{N}_G$. Then, since $\mathcal{M}^* \Leftrightarrow \mathcal{N}_G^*$ and $\mathcal{M}^*$ is a minimal model, we can conclude that $\mathcal{N}_G^*$ has at least as many states as $\mathcal{M}^*$. □

As a corollary of this theorem we can show that the smallest GMM containing a given HMM $\mathcal{N}$ is the minimal representation for $\mathcal{N}_G$. This is because we have shown that every GMM $\mathcal{M}$ containing $\mathcal{N}$ must satisfy $\mathcal{N}_G \subseteq \mathcal{M}$. It is easy to show that this implies that the canonical dimension of $\mathcal{M}$ must be at least as large as that of $\mathcal{N}_G$. It can also be shown that if A and B are GMMs with the same canonical dimension and $A \subseteq B$, then $A \Leftrightarrow B$. Putting these facts together we can see that the minimal representation of $\mathcal{N}_G$ is the smallest GMM we could possibly pick to contain $\mathcal{N}$.

Theorem 3.5 showed that the minimal HMM representation of a class of processes will be at least as big as the minimal GMM containing that class. We can also show that if we insist on having a stochastic interpretation of the parameters of a model, we may sometimes need *many* more states than the minimal GMM can achieve. We can see this as follows. Notice that the space of distributions on outputs spanned by a k-output HMM defines a convex polyhedron on the $k-1$ dimensional probability simplex. The vertices of the polyhedron are defined by the convex hull of the output distributions on the states. By choosing the priors on the model appropriately we can explore every corner of the polyhedron. In the worst case, the output distributions of every state may fall on the convex hull, and so it would be impossible to build a smaller stochastic model of them. However, if we permit ourselves to use general linear combinations, we may find that many of the output distributions are linear combinations of each other, which leads to potential reducibility. This shows that if our goal is to find parsimonious and easily manipulable representations for stochastic

processes, using GMMs would appear to be a very reasonable course of action.

If we insist on using models with stochastic parameters, it is possible to define a *stochastic canonical dimension* of an HMM. This quantity would represent the number of "basis vectors" we would need if we only used convex combinations in all the places where we currently use general linear combinations. Analysis of this definition is more diffcult since the "basis vectors" for convex combinations correspond to vertices of convex polyhedra and the wealth of results concerning bases for linear vector spaces is not available. However, a brief consideration of the problem suggests that it is very likely that an HMM can be reduced to a minimal *stochastic* representation with only as many states as its *stochastic caonical dimension*.

We have now concluded the major portion of this thesis. The next chapter will discuss further directions of research and point out several questions that were not sufficiently investigated in this work.

Chapter 4

Further Directions and Conclusions

In Chapter 3 we defined Generalized Markov Models, a new class of finite-state representations for stochastic processes, and saw how the results on equivalence of HMMs could be extended to GMMs. We used this to define the canonical dimension of a GMM and developed a complete characterization of the minimal, canonical representations for these models. We also saw how HMMs are related to GMMs, and observed that a minimal representation for a stochastic process in the HMM formalism necessarily has at least as many states as the minimal representation in the GMM model. One issue that was not thoroughly investigated in this thesis involves characterizing the class of valid priors on a GMM. Since the definition of GMMs was not constructive, it is not obvious what the space of valid priors on a model looks like. Hence, we do not have a characterization of the class of valid transition matrices for GMMs. One way of trying to understand this issue is to apply the well-worn vector space techniques of this thesis once again, this time to the task of determining whether a given pseudo-prior is valid for a model. Similarly, we could determine whether a given transition matrix is allowable. In addition to the problem of characterizing the valid priors on a model, there are several other important issues that were not considered in this thesis. We will discuss these in the sections below.

4.1 Reduction of Initialized HMMs

In Chapter 3 we addressed the problem of reducing GMMs to minimal canonical forms. We saw that a class of processes with canonical dimension d needed at least d states in its GMM representation. In many applications, after the stage of training parameters for a model is completed, we will not actually need the freedom of being able to set different prior distributions on the model. In other words, we will actually be dealing with an Initialized GMM. Since the model now represents a *single* process, it may be possible to reduce the number of states still further.[1] So we should consider how to reduce an Initialized GMM $(\mathcal{M}, \vec{p})$ to a minimal representation $(\mathcal{N}, \vec{q})$ such that $\mathcal{N}(\vec{q}) \Leftrightarrow \mathcal{M}(\vec{p})$ and $\mathcal{N}$ has as few states as possible.

4.2 Reduction While Preserving Paths

In some pattern recognition applications of Hidden Markov Models the maximum likelihood path producing an output sequence x is as important as the probability that x is produced. In such cases, we will be faced with two new issues that were not addressed in this thesis. First of all, we will have to give meaning to a "maximum likelihood path" in a Generalized Markov Model. Secondly, we will have to find a method of model reduction that preserves enough information about them to recover the identity of paths in the original model from paths in the reduced model. There are some applications in which we are only interested in passage through some small number of states rather than the entire path. In such situations, the simplest way of achieving reduction while preserving paths would be to declare the appropriate states to be *irreducible*. Such states would never be merged with others in the reduction algorithm and so their identity would be preserved in the reduced model.

[1] For example, suppose a GMM has one state that loops on itself with probability 1, and we initialize the model with all the mass on the looping state. Then, once we have fixed this prior, all the other states are clearly unnecessary.

4.3 Training GMMs

When we defined Generalized Markov Models in Chapter 3 we made no mention of training algorithms for these models. This was partly because the class of valid transition matrices and priors was not characterized, and this makes it difficult to evaluate whether a given set of parameters induces valid stochastic processes. Nonetheless, there are some options that come to mind immediately. First of all, we could use corrective training methods, such as gradient descent to minimize a squared error measure. (Niles [niles90] suggests such a procedure in the context of his HMM-net.) Furthermore, despite their exotic underlying chains, GMMs still define true probability distributions on their output sequences. Consequently, it still makes sense to think about Maximum Likelihood methods where we would attempt to set the parameters of the model to maximize the likelihood of a database of examples. The easiest way to derive a method for updating the parameters would be to follow the derivation of Levinson et al., who treat Maximum Likelihood Estimation in the framework of classical constrained optimization.[levinson83]

4.4 Approximate Equivalence

Although the results of the previous chapter are a complete characterisation of equivalence and reduction of GMMs, they can be a little unsatisfying, as the following example shows. Suppose $\mathcal{M}$ is a model whose transition amplitudes are all equal and all of whose output distributions are linearly independent of each other. According to our results, this model is not reducible because there is no degeneracy in the output distributions. Indeed, it is true that we cannot build a smaller model that agrees with $\mathcal{M}$ at all times. This is because it is always possible to pick priors in such a way as to explore the entire valid span of the output distributions of $\mathcal{M}$, while a smaller model could not span a space of the same dimension. Yet, it is clear that after the first output, the distribution over states will be uniform and the probability of emitting

various outputs will always be unchanged. We might like to ignore the first output, and say that $\mathcal{M}$ can be reduced to a single state, since what happens in the beginning is an artifact of the prior. Minor modifications to our results would accomodate this - whenever a theorem evaluated a condition for every $x \in \mathcal{O}^* \cup \{\epsilon\}$, we would instead evaluate the condition for $\{x : |x| > 1\}$. We would also need to appropriately modify the vector spaces whose properties were checked in various algorithms developed in this thesis.

However, this brings up the more general question of approximation algorithms. Often, we may not care about what happens at early or late times. Or we may not care if the statistics defined by two models are exactly the same so long as they are close. Approximate equivalence in this sense of "closeness" of models is particularly important because the parameters of probabilistic models are usually estimated from data. Consequently, exact equivalence will be a rare event. Equivalence ignoring late or early times can be easily handled within our methods by various slight modifications of our results. The interesting and difficult problem is to define "closeness" of stochastic processes appropriately and to prove under what conditions the two GMMs are "close" under the definition.

4.5 Practical Applications

Finally, we should mention the possible practical applications of this work, particularly since it was originally begun in the context of building better practical classifiers. Statitical methods and models are being increasingly used in pattern recognition and other fields. The models built in some applications can be very large ([kupiec90]) and reducing them to equivalent models of smaller sizes would be computationally useful. However, since the parameters of models are typically estimated from data, they will very rarely be exactly reducible and the approximation algorithms mentioned in the previous section will be crucial. Since we do not currently have a provably good al-

gorithm for approximate reduction, a reasonable preliminary course to take would be to substitute tests for linear dependence with tests for "almost" linear dependence in all the algorithms and results of the previous chapters. Of course, it is also possible to simply simply build a smaller model and retrain it from data rather than reducing a larger model. However, if the model would take a long time to train (e.g., if the database of examples is very large), or if the large model was constructed for human readability and manual fine tuning ([kupiec90]), reduction of a large model would be a better course of action. Even if we prefer to retrain a smaller model, the canonical dimension defined in Chapter 3 could be evaluated as a way of testing whether a smaller model should be built and retrained. The reduction algorithm could also be used as a way of finding the structure of a good smaller model that is equivalent or nearly equivalent to the original. Even if we retrain the parameters of the reduced model, the reduction step would tell us how many states we are likely to need to get a good representation of the statistics modelled by the larger model.

Another potential practical application involves the implementation and evaluation of GMMs as pattern classifiers. There is some reason to suspect that given a GMM and an HMM with n states each, the GMM could perform better as a pattern classifier. This is plausible because, given a fixed number of states, a GMM can model a wider class of processes than an HMM. In practical applications we are typically dealing with the problem of *approximating* stochastic sequences. There may be processes modelled by n-state GMMs that are much closer to the true process than the best approximation we can find in the HMM formalism. In order to understand this question from the theoretical viewpoint we would need to make progress along several fronts including understanding the approximation properties of HMMs. For example, we would need to be able to compare how accurately a given stationary process can be represented by HMMs and GMMs with n states each. This is a difficult problem worthy of being studied. From the point of view of practical applications, the question of the usefulness of GMMs is best resolved empirically in the domain of application.

4.6 Conclusion

This thesis arose from an attempt to build part of a good foundation for pattern recognition using Hidden Markov Models. There is a need for analytical tools that will enable us to compare different formalisms for pattern recognition and in order to predict their relative effectiveness. In this thesis, we have proved several theorems that uncover the source of the intrinsic expressiveness of Hidden Markov Models. We have shown how to detect equivalence of prior distributions on a model and given a geometric characterization of equivalent priors. This led to a characterization of equivalent Initialized Hidden Markov Models and then of equivalent HMMs. We have given theorems that detect these equivalencies in polynomial time. Next, empirical and theoretical motivations led us to define the class of Generalized Markov Models which contain HMMs as a subclass. We used the definition to reduce HMMs and GMMs to minimal, canonical representations which remove all redundancy from a model. We also developed a geometric characterization of the minimal representations that gave insight into the source of the expressiveness of GMMs and HMMs. This characterization also led to a polynomial time reduction algorithm for Generalized Markov Models. These results lay part of a foundation for the principled use of finite-state models of stochastic processes in pattern recognition.

Appendix A

HMMs and Probabilistic Automata

There have been some recent results in the theory of Probabilistic Automata (PAs) that use methods very similar to ours to decide the equivalence of PAs in polynomial time.[tzeng] Tzeng's work also discusses a result on approximate equivalence of PAs that may provide leads on ways to proceed towards understanding approximate equivalence of HMMs and GMMs. In this appendix we will show how HMMs and PAs are related. First of all, we will define Probabilistic Automata in Tzeng's formulation.

Definition A.1 *Probabilistic Automata*

Let $\mathcal{M}(i,j)$ denote the set of all $i \times j$ stochastic matrices. A Probabilistic Automaton U is a 5-tuple (S, Σ, M, ρ, F), where $S = \{s_1, s_2, \ldots, s_n\}$ is a finite set of states, Σ is an input alphabet, M is a function from Σ into $\mathcal{M}(n,n)$, ρ is a prior distribution on the states, and $F \subseteq S$ is a finite set of final states. $M(\sigma)_{ji}$ is is the probability that U moves from state s_i to s_j after reading the symbol $\sigma \in \Sigma$. We say that $x \in \Sigma^*$[1] is accepted by U with probability $p_U(x)$ if U ends up in a final state with probability $p_U(x)$ on reading x.

It is clear from this definition that PAs are closely related to HMMs. The matrices $M(\sigma)_{ji}$ are similar in meaning to HMM transition operators, barring the complication

[1] We are using the standard notation that Σ^* is the set all finite length strings that can be produced by concatentating symbols in Σ together.

of the final states, the rest of the model is similar also. We will show that with an appropriate definition of equivalence, Initialized HMMs are a subclass of Probabilistic Automata.

Definition A.2 *Equivalence of PAs and HMMs*

Let U be a Probabilistic Automaton and let $(\mathcal{M},\vec{p})$ *be a Hidden Markov Model with* $\mathcal{O} = \Sigma$*, where* $\mathcal{O}$ *is the output set of* $\mathcal{M}$*, and* Σ *is the input alphabet of U. We will say that U and* $\mathcal{M}$ *are equivalent* $(U \Leftrightarrow (\mathcal{M},\vec{p}))$ *if* $\Pr(x|\mathcal{M},\vec{p}) = p_U(x)$ *for every* $x \in \mathcal{O}^* \cup \{\epsilon\}$*.*

Definition A.2 says that a Probabilistic Automaton U is equivalent to a Hidden Markov Model $\mathcal{M}$when U accepts strings with the same probability that $\mathcal{M}$ emits them. We will now show that under this definition of equivalence, HMMs are contained in the class of Probabilistic Automata.

Theorem A.1 *HMMs* $\subset$ *PAs*

The class of Hidden Markov Models is contained in the class of Probabilistic Automata when equivalence is defined by Definition A.2.

The basic idea of the proof is shown in Figure A. Given an HMM $\mathcal{M}$, we will build a Probabilistic Automaton U with the same states plus one extra state to leak away excess probabilities. We will then set up the matrices $M(\sigma)_{ji}$ to mimic the action of the HMM transition operators on the states that the two models share. If every one of the shared states is a final state of U, this will guarantee that $\mathcal{M}$ and U are equivalent.

Proof: Suppose $\mathcal{M} = (\mathcal{S},\mathcal{O},\mathbf{A},\mathbf{B})$ is any HMM with prior $\vec{p}$. Construct a PA $U = (S_U,\Sigma,M,\rho,F)$ where:

$$S_U = \mathcal{S} \cup \{s_U\} \tag{A.1}$$

$$\Sigma = \mathcal{O} \tag{A.2}$$

$$F = \mathcal{S} \tag{A.3}$$

$$\rho_i = \left\{ \begin{array}{l} p_i \text{ if } s_i \in \mathcal{S} \\ 0 \; otherwise \end{array} \right\} \tag{A.4}$$

$$M(\sigma)_{ij} = \left\{ \begin{array}{l} \mathbf{B}_{\sigma i}\mathbf{A}_{ij} \text{ if } s_i, s_j \in \mathcal{S} \\ 1 - \sum_{s_i \in \mathcal{S}} M(\sigma)_{ij} \text{ if } s_j \in \mathcal{S},\ s_i = s_U \\ 0 \text{ if } s_j = s_U, s_i \in States \\ 1 \text{ if } s_i = s_j = s_U \end{array} \right\} \tag{A.5}$$

We will prove by induction on the length of strings x that $p_U(x) = \Pr(x|\mathcal{M}, \vec{p})$ for every string x. First of all, both models accept the null string with probability 1, since both start with all the mass of a stochastic vector on the states $\mathcal{S}$. Furthermore, $\Pr(s_i, \epsilon|U) = \Pr(s_i, \epsilon|\mathcal{M}, \vec{p})$. Now suppose that for every string x, such that $|x| \leq t$, it is true that $p_U(x) = \Pr(x|\mathcal{M}, \vec{p})$ and that for every state $s_i \in \mathcal{S}$, $\Pr(s_i, x|U) = \Pr(s_i, x|\mathcal{M}, \vec{p})$. Then for any symbol $\sigma \in \Sigma = \mathcal{O}$, and every state s_j, it will be true that:

$$\Pr(s_j, x\sigma|U) = \sum_{i=1}^{|\mathcal{S}|} (\mathbf{B}_{\sigma i}\mathbf{A}_{ji} \Pr(s_i, x|U)) + \Pr(s_U, x|U) M(\sigma)_{jU} \tag{A.6}$$

$$= \sum_{i=1}^{|\mathcal{S}|} \mathbf{B}_{\sigma i}\mathbf{A}_{ji} \Pr(s_i, x|\mathcal{M}, \vec{p}) \tag{A.7}$$

$$= \Pr(s_j, x\sigma|\mathcal{M}, \vec{p}) \tag{A.8}$$

Furthermore, since the accepting states of U are exactly the states in $\mathcal{S}$ it also follows that $p_U(x\sigma) = \sum_{s_j \in \mathcal{S}} \Pr(s_j, x\sigma|U) = \Pr(x\sigma|\mathcal{M}, \vec{p})$. By induction on $t = |x|$, we can conclude that for every string x, the probability that x is produced by $\mathcal{M}$ is equal to the probability that x is accepted by U. Hence, $(\mathcal{M}, \vec{p}) \Leftrightarrow U$ and, therefore, $HMMs \subseteq PAs$. On the other hand, it is easy to show that there are probabilistic automata that cannot be implemented as Hidden Markov Models. For example, define the *support* of a PA to the set of strings that are accepted with non-zero probability. The support of an HMM would be the set of strings that are emitted with

non-zero probability. Because of the definitions of the models, a PA could have a finite support, or even a support that consists only of strings longer than a fixed length. Neither of these two cases is possible for an HMM. Furthermore, the various $M(\sigma)$ matrices of a PA need not bear any relationship to each other, while the corresponding transition operators of HMMs are closely related to each other, via the **A** and **B** matrices. For this reason also it is possible to define PAs that are not equivalent to any Hidden Markov Model in our formulation of equivalence. So we conclude that $HMMs \subset PAs$. □

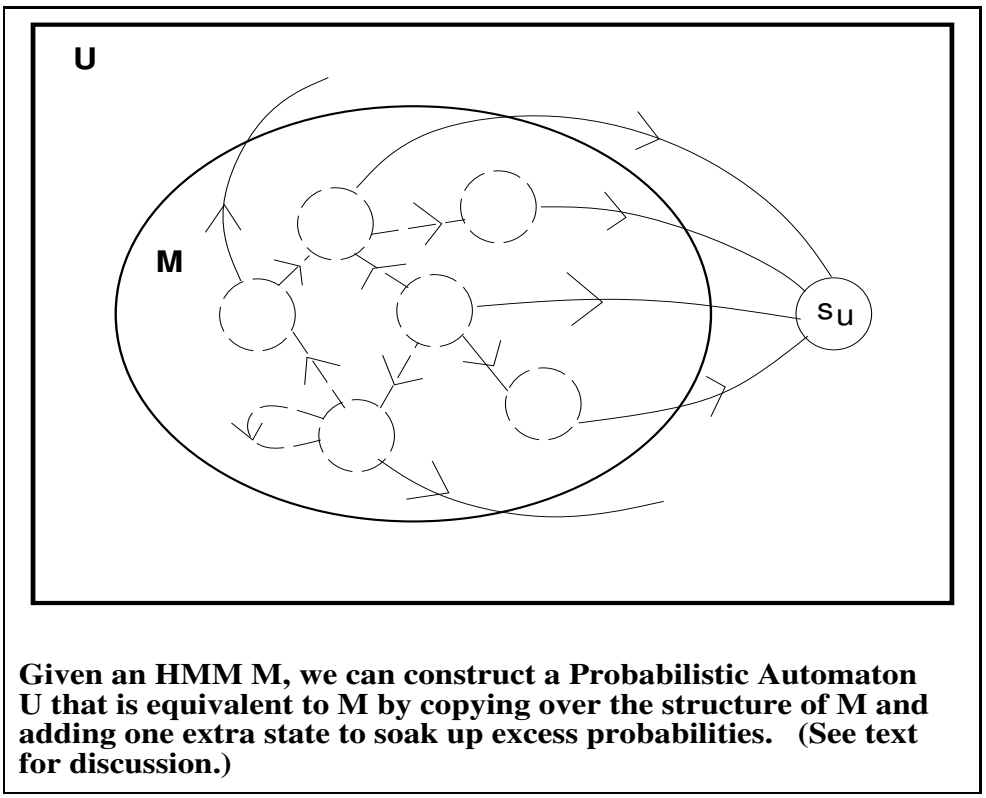


Figure A-1: Constructing a Probabilistic Automaton Equivalent to An HMM

Theorem A.1 shows that HMMs can be considered a subclass of Probabilistic Automata. However, if the number of outputs is large, an n-state PA will require

many more parameters to describe it than an n-state HMM. W.Tzeng proves results concerning equivalence of Probabilistic Automata using methods that are similar to ours.[tzeng] He also discusses the problem of approximate equivalence of PAs and arrives at some interesting results. Since HMMs and PAs are so closely related, the methods used by Tzeng to extract results concerning approximate equivalence can guide us in studying the same question for HMMs.

Bibliography

[bahl88] L.R.Bahl, P.F.Brown, P.V. de Souza, R.L.Mercer and M.A.Picheny, *Acoustic Markov Models Used In The Tangora Speech Recognition System*

[bodreau68] P.E.Bodreau, *Functions of Finite Markov Chains and Exponential Type Processes*, Annals of Mathematical Statistics, Vol.39, No.3, pp.1020-1029, 1968.

[burke58] C.J.Burke and M.Rosenblatt, *A Markovian Function of A Markov Chain*, Annals of Mathematical Statistics, 1958.

[chvatal80] V.Chvátal, *Linear Programming*, W.H.Freeman and Company, New York, 1980.

[dharma63a] S.W.Dharmadhikari, *Functions of Finite Markov Chains*, Annals of Mathematical Statistics, Vol.34, pp.1022-1032, 1963.

[dharma63b] S.W.Dharmadhikari, *Sufficient Conditions For A Stationary Process To Be A Function Of A Finite State Markov Chain*, Annalas of Mathematical Statistics, Vol.34, 1963.

[dharma68] S.W.Dharmadhikari, *SPlitting A Single State Of A Stationary Process Into Markovian States*, Annals of Mathematical Statistics, Vol.38, No.3, pp.1069-1077, 1968.

[blackwell57] D.Blackwell and L.Koopmans, *On The Identifiability Problem for Functions of Finite Markov Chains*, Annals of Mathematical Statistics, Vol.28, pp.1011-1015, 1957.

[gilbert59] E.Gilbert, *On The Identifiability Problem For Functions of Finite Markov Chains*, Annals of Mathematical Statistics, Vol.30, pp.688-697, 1959.

[iosifescu80] M.Iosifescu, *Finite Markov Processes and Their Applications*, John Wiley & Sons, New York, 1980.

[ito92] H.Ito, S.Amari and K.Kobayashi, *Identifiability of Hidden Markov Information Sources and Their Minimum Degrees of Freedom*, IEEE Transactions on Information Theory, Vol.38, No.2, March 1992.

[juang91] B.H. Juang and L.R.Rabiner, *Hidden Markov Models for Speech Recognition*, Technometrics, Vol.33, No.3, August 1991.

[kamp85] Y.Kamp, *State Reduction in Hidden Markov Chains Used for Speech Recognition*, IEEE Transactions of Acoustics, Speech and Signal Processing, Vol.ASSP-33, No.4, October 1985.

[karmarkar84] N.Karmarkar, *A New Polynomial Time Algorithm For Linear Programming*, Combinatorica, Vol.4, No.4, pp.373-395, 1984.

[kemeney65] J.G.Kemeney and J.L.Snell, *Finite Markov Chains*, D.Van Nostrand Company Inc., Princeton, New Jersey, 1965.

[kopec91] G.E.Kopec and P.A.Chou, *Document Image Decoding Using Markov Sources*, submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992.

[kupiec90] J.Kupiec, *Part-of-Speech Tagging Using a Hidden Markov Model*, Xerox PARC System Sciences Lab Tech. Report No. SSL-90-39, 1990.

[lee88] Kai-Fu Lee, *Large-Vocabulary Speaker Independent Continuous Speech Recognition: The SPHINX System*, Carneige-Mellon University Computer Science Department, PhD Thesis, 1988.

[levinson83] S.E.Levinson, L.R.Rabiner and M.M.Sondhi. *An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition*, The Bell System Technical Jounal, Vol.62, No.4, April 1983.

[niles90] L.T.Niles, *Modelling and Learning in Speech Recognition: The Relationship Between Stochastic Pattern Classifiers and Neural Networks*, Technical Report LEMS-79, Brown University, August 1990.

[paz71] A.Paz, *Introduction to Probabilistic Automata*, Aacademic Press, London, 1971.

[poggio89] Tomaso Poggio and Federico Girosi, *A Theory of Networks for Approximation and Learning*, Artificial Intelligence Laboratory, M.I.T., A.I.Memo No.1140, July 1989

[press90] W.H.Press, B.P.Flannery, S.A.Teukolsky, and W.T.Vetterling, *Numerical Recipes in C*, Cambridge University Press, New York, 1990.

[rabiner86] L.R.Rabiner and B.H.Juang, *An Introduction to Hidden Markov Models*, IEEE ASSP Magazine, pp.4-16, January 1986.

[rosenblatt71] M.Rosenblatt, *Markov Processes - Structure and Asymptotic Behaviour*, Springer-Verlag, New York, 1971.

[tzeng] W.Tzeng, *A Polynomial Time Algorithm for the Equivalence of Probabilistic Automata*, SIAM Journal on Computing, to appear.