Department of Computing

Efficient Duration Modelling in the Hierarchical Hidden Semi-Markov Models and Their Applications

by

Thi V. T. Duong

This thesis is presented for the degree of
Doctor of Philosophy
of

Curtin University of Technology

August 2008

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledge has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

.....

August 2008

Thi V.T. Duong

Contents

List of Figures

List of Tables

Abstract

Modeling patterns in temporal data has arisen as an important problem in engineering and science. This has led to the popularity of several dynamic models, in particular the renowned hidden Markov model (HMM) [Rabiner, 1989]. Despite its widespread success in many cases, the standard HMM often fails to model more complex data whose elements are correlated hierarchically or over a long period. Such problems are, however, frequently encountered in practice. Existing efforts to overcome this weakness often address either one of these two aspects separately, mainly due to computational intractability. Motivated by this modeling challenge in many real world problems, in particular, for video surveillance and segmentation, this thesis aims to develop tractable probabilistic models that can jointly model duration and hierarchical information in a unified framework. We believe that jointly exploiting statistical strength from both properties will lead to more accurate and robust models for the needed task.

To tackle the modeling aspect, we base our work on an intersection between dynamic graphical models and statistics of lifetime modeling. Realizing that the key bottleneck found in the existing works lies in the choice of the distribution for a state, we have successfully integrated the discrete Coxian distribution [Cox, 1955], a special class of phase-type distributions, into the HMM to form a novel and powerful stochastic model termed as the *Coxian Hidden Semi-Markov Model* (CxHSMM). We show that this model can still be expressed as a dynamic Bayesian network, and inference and learning can be derived analytically. Most importantly, it has four superior features over existing semi-Markov modelling: the parameter space is compact, computation is fast (almost the same as the HMM), close-formed estimation can be derived, and the Coxian is flexible enough to approximate a large class of distributions. Next, we exploit hierarchical decomposition in the data by borrowing analogy from the hierarchical hidden Markov model in [Fine et al., 1998, Bui et al., 2004] and introduce a new type of shallow structured graphical model that

combines both duration and hierarchical modelling into a unified framework, termed the *Coxian Switching Hidden Semi-Markov Models* (CxSHSMM). The top layer is a Markov sequence of switching variables, while the bottom layer is a sequence of concatenated CxHSMMs whose parameters are determined by the switching variable at the top. Again, we provide a thorough analysis along with inference and learning machinery. We also show that semi-Markov models with arbitrary depth structure can easily be developed. In all cases we further address two practical issues: missing observations to unstable tracking and the use of partially labelled data to improve training accuracy.

Motivated by real-world problems, our application contribution is a framework to recognize complex *activities of daily livings (ADLs)* and detect *anomalies* to provide better intelligent caring services for the elderly. Coarser activities with self duration distributions are represented using the CxHSMM. Complex activities are made of a sequence of coarser activities and represented at the top level in the CxSHSMM. Intensive experiments are conducted to evaluate our solutions against existing methods. In many cases, the superiority of the joint modeling and the Coxian parameterization over traditional methods is confirmed. The robustness of our proposed models is further demonstrated in a series of more challenging experiments, in which the tracking is often lost and activities considerably overlap. Our final contribution is an application of the switching Coxian model to segment education-oriented videos into coherent topical units. Our results again demonstrate such segmentation processes can benefit greatly from the joint modeling of duration and hierarchy.

Acknowledgments

The dedication of this thesis is split five ways. First and foremost, this is to my main-supervisor *Prof. Svetha Venkatesh* for truly making it possible. You were always there for me in many different ways, guiding me through my PhD, helping me to structure my thesis, returning my chapters with beneficial questions and comments, providing the flexibility I vitally need and much more. To *Mum*, you came a very far distance to help me in taking care of my young family and I'm forever indebted to you for the sacrifice you made in bringing me up and giving me an education. It is to my late *Father* for all the things you taught and inspired me in my childhood. To my husband *Dinh*, you were being so wonderfully understanding and supportive. Last and surely not least, it is to my little *Gau* for coming into our life. It was an awesome joy just to watch you smiling and toddling around the house.

Also, I'm deeply grateful to my co-supervisors Dr. Hung Bui and Dr. Dinh Phung for their invaluable help in formulating important ideas for the thesis. I specially thank Dr. Nam Nguyen for sharing his vision tracking system and Mary for English correction and other stuffs. Finally, I would like to thank Viet, my brother-in-law, for the time he have spent with us, and I'm sending a heartfelt thank-you to my sister's family for their support and hospitality and to my brother Tam for the good time we shared when I was in Vietnam working on the first part of my PhD.

Relevant Publications

Part of this thesis and some related work has been published or documented elsewhere. The list of these publications is provided below.

- **Duong, V.T.**, Bui, H., Phung, D.Q., and Venkatesh, S. (2008). Efficient Duration and Hierarchical Modeling for Human Activity Recognition, *Artificial Intelligence Journal* (JAI), 2008. (to appear).

- **Duong, V.T.**, Phung, D. Q., Bui, H.H., and Venkatesh, S. (2006), Human Behavior Recognition with Generic Exponential Family Distribution Modeling in the Hidden Semi-Markov Model Track Number. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 202-207, August 2006, Hong Kong.

- **Duong, V.T.**, Bui, H., Phung, D.Q., and Venkatesh, S. (2005). Activity Recognition and Abnormality Detection with the Switching Hidden Semi-Markov Model. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition* (CVPR'05), p. 838-845, June 2005, San Diego, USA.

- Phung, D.Q., **Duong, V.T.**, Venkatesh, S., and Bui, H. (2005). Topic Transition Detection Using Hierarchical Hidden Markov and Semi-Markov Models. In *Proceedings of ACM International Conference on Multimedia* (ACM-MM), pages 11-20, November 2005, Singapore.

- **Duong, V.T.**, Phung, D. Q., Bui, H.H., and Venkatesh, S. (2005), Efficient Coxian Duration Modelling for Activity Recognition in Smart Environment with the Hidden semi-Markov Model. In *Proceedings of International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 5-6 December 2005, Melbourne, Australia.

- **Duong, V.T.**, Bui, H.H., Phung, D.Q., and Venkatesh, S. (2004), Learning and Recognizing Human Activities with the Switching Hidden Semi-Markov Model, in *Workshop on Activity Recognition*, *International Conference on Advances in Neural Information Processing* (NIPS), December 2004, Vancouver, Canada.

Abbreviations

Abbreviation	Meaning
PH	Phase-Type
Geom	Geometric
ADLs	Activities of Daily Living
ESS	Expected Sufficient Statistic
ML	Maximum Likelihood
EM	Expectation Maximization
BN	Bayesian Network
DBN	Dynamic Bayesian Network
HMM	Hidden Markov Model
HSMM	Hidden Semi-Markov Model
CxHSMM	Coxian Hidden Semi-Markov Model
$\mathcal{M}$-ph.CxHSMM	$\mathcal{M}$-phase Coxian Hidden Semi-Markov Model
HHMM	Hierarchical Hidden Semi-Markov Model
SHSMM	Switching Hidden Semi-Markov Model
CxSHSMM	Coxian Switching Hidden Semi-Markov Model

Notations

Symbol	Meaning
$\boxplus$	end of an example
$\blacksquare$	end of definition
$\square$	end of proof

Chapter 1

Introduction

Pattern recognition in *sequential* and *stream* data has become an increasingly important research topic. The ability to *represent*, *infer* and *learn* high-level patterns is crucial in a wide range of real-world applications such as event mining in intelligent sensor network processing, activity recognition in smart assistive technologies, abnormality detection in public surveillance, mining temporal patterns on the web, or discovery of genome structures in computational biology. The key challenge in this process is to deal with the *uncertainty* that arises at difference phases, in particular, during modeling processes and in sensor measurement errors.

Motivated by this pressing challenge, this thesis seeks robust probabilistic models to tackle the problem of high-level pattern recognition from the most popular form of sensory data - *streaming videos*. Recognizing events in video data, such as activity recognition, has been a long standing research topic for the last decade. However, most existing approaches are limited to simple events since modeling complex events, such as those that can capture long-range correlations in the data, is often prohibited, mainly due to the computational bottleneck at the representation and inference stages. Realizing this obstacle, this thesis exploits two important properties of the data, namely *duration* and *hierarchical* information, for our modeling purpose. To our knowledge, such modeling efforts pose great difficulties and have not been rigorously explored in the past.

One obvious choice for temporal modeling is the hidden Markov models [Rabiner, 1989]. This celebrated model is simple, compact and has proven to work well in many domains. Nonetheless, it is limited to simple modeling and fails to capture

long-correlation in the data due to the strict first-order Markov assumption. The hidden semi-Markov model is an attempt to relax this strictness by further specifying the lifespan distribution for each state. But, it is still limited in practice due to the computational cost, which grows linearly with the maximum possible duration of a state, and optimisation problem when the duration distribution is continuous. This particularly becomes problematic when dealing with sequential data since duration can easily grow unmanageable. Hierarchical hidden Markov models [Fine et al., 1998, Bui et al., 2004] is another effort to go beyond the simple HMM. Hierarchical modeling can address not only the complex correlations in the data but also the semantic decomposition often found in video data (e.g., goals and sub-goals in activity recognition or episodes, scenes, shots in films). Clearly, probabilistic models that can jointly model both duration and hierarchy, with efficient inference machinery, can potentially be extremely useful in many domains. We are motivated to explore this type of modeling and its potential applications, in particular, for two domains: recognition of normal/abnormal activities in surveillance video and high-level segmentation of education-oriented video.

1.1 Aims and Approach

This thesis aims to develop robust probabilistic models for pattern recognition in sequential and streaming data. It fosters applications of these models for smart home surveillance and video segmentation. In particular, we seek the answer for the following questions:

- Can we develop a new duration modeling framework that overcomes the computational bottleneck presented in current existing hidden semi-Markov models? Can we represent it in a dynamic Bayesian network form and can we construct suitable machinery for learning and inference?
- Can we develop new forms of stochastic models that seamlessly incorporate duration and hierarchical information into a unified framework? How can we construct inference and learning algorithms? To what extent are they tractable and efficient?
- How can we use our solutions to provide better sensing intelligence for surveillance environments and segmentation in video?

Our approach, from a modelling perspective, is driven by the advantages provided by probabilistic graphical modelling. Here, we revisit the existing hidden semi-Markov models in the form of dynamic Bayesian networks to provide better intuition and understanding into their representation and complexity. The key difference with existing work is that we conduct a thorough investigation and provide a unified exponential family duration modeling framework for both continuous and discrete distributions. In seeking better alternative modeling choices, we research the mature branch of statistics concerned with lifetime modeling [Barlow and Proschan, 1981] and in particular, a class of discrete phase-type distributions that is known to be flexible in approximating any arbitrary distribution [Johnson and Taaffe, 1988]. For hierarchical modeling our approach is motivated by the success of the hierarchical hidden Markov models recently proposed in [Fine et al., 1998] and later extended with general state hierarchy in [Bui et al., 2004, Phung, 2005b]. Again, using the language of dynamic Bayesian networks, we seek to construct novel forms of stochastic models that can jointly model flexible duration distributions and hierarchy. Machinery for tractable inference and parameter learning is subsequently investigated, again, under the umbrella of directed graphical models.

From the application perspective, our first motivating application is the construction of a safe and smart house for the aged that facilitates automatic monitoring and support of its occupants, aiming to increase the opportunity for aging in the family home. This is a growing and important research area, in particular, for countries with increasing aging populations such as Australia [Ball, 2003]. Our aim is to focus on specific types of daily activities, or *activities of daily living* (ADLs) [Katz et al., 1963], because they are the measure of both cognitive and physical functions of the occupants and can be used to assess the fitness of the elderly living independently [Lawton, 1990, Rogers et al., 1998]. Our specific aims are:

- To develop a robust framework to define and recognise a set of complex activities performed routinely by the elderly from camera monitoring videos. In particular, to seek solutions to distinguish activities with rich hierarchical and duration information.
- To detect any anomalies that may arise in the activities so that prompt attention can be given, if needed.

Our second motivating application is to learn, segment and possibly annotate education-oriented videos into coherent units of topical content. This is an important step to enabling abstraction, summarization, and browsing of educational content, particularly useful in building e-services for learning and training. It is important in this modeling process to exploit the long-term, multiple-scale correlations of video dynamics. This task, however, is complicated since the semantics in videos are often organized hierarchically and their duration distributions vary greatly. This poses a similar challenge to the previous activity recognition, and again, we aim to use our modeling framework to tackle these issues.

1.2 Significance and Contribution

The significance of this work can be divided into two parts. *The theoretical significance* includes the development of a set of novel, temporal, stochastic models with efficient computation to model hierarchical and duration properties in sequential data. *The application significance* includes: (a) a system to recognise normal activities and detect anomalies from video data, and (b) a probabilistic framework to segment educational videos into units of topical contents. In particular, our theoretical contributions are:

- *A thorough investigation into the aspect of duration modeling in the HSMM under the generic representation of exponential family distributions.* This helps to flesh out the pros and cons in using continuous and discrete distributions as well as to identify the key drawbacks of existing computational bottlenecks.

- *The innovative integration of the Coxian distributions*[1] *for modeling state durations in the HSMM.* The Coxian parameterization offers several advantages over traditional modeling: (a) it is computationally attractive since inference complexity scales linearly with the number of Coxian phases, which is typically much smaller compared with the length of the data sequence, (b) it has a small number of free parameters, again scaling linearly with the number of phases, (c) Parameter learning can be done analytically with closed-form solutions, and (d) it is theoretically flexible enough to approximate any generic distributions.

[1]The Coxian distributions, a subfamily of the Phase-Type distributions, was introduced by David Cox in [Cox, 1955].

- *The development of a novel Coxian Hidden Semi-Markov Model* and its complete analysis including dynamic Bayesian network representation, inference and maximum likelihood estimation under a partially observed data case. We also address how the model can be adapted to deal with *missing* observation often caused by imperfect camera tracking.

- *A novel formulation of the Coxian Switching Hidden Semi-Markov Model* (CxSHSMM)[2] that seamlessly integrates hierarchy, structure sharing and information into a unified framework. The CxSHSMM has a shallow hierarchical structure where at the bottom layer is a set of concatenated Coxian semi-Markov models, each initiated by a state at the top level, whose states switch in a Markovian manner. Again, this model is accompanied by a complete analysis for its DBN representation, inference and learning under missing observation and partially labelled data cases.

- In conjunction with the previous model, *we introduce a set of Switching Hidden Semi-Markov Models* (SHSMMs) having the same structure as the CxSHSMM but state duration is modelled in a generic class of exponential family distribution. This work helps to compare and contrast the difference between the Coxian and other duration models and to broaden our investigation for the sake of completeness.

- To move beyond shallow structures we show that hierarchical hidden semi-Markov models with arbitrary depth can be constructed. We detail how they can be represented by a DBN and discuss relevant inference algorithms when the depth is high.

In applying our modeling solutions, main contributions are:

- *A system that uses the CxHSMM as the main representation and inference machinery to learn and recognize activities of daily livings in a smart home context.* We also make comparisons with other modeling choices (including Multinomial, Poisson, and Inverse Gaussian) and demonstrate that: (a) duration information is essential for accurately modeling activities and can be most effectively exploited by the Coxian parameterization, and (b) high recognition

[2]In fact, by the time of writing this thesis, this model has received an enormous attention from the community. Its early result in the CVPR paper has been cited by more than 45 times, reported by GoogleScholar as of 3rd March 2007.

accuracy can be achieved with a relatively small number of Coxian phases. The latter point is particularly significant since it implies a great reduction in parameter space to overcome the computational bottleneck encountered by traditional methods.

- *A system that employs the CxSHSMM as the main representation and inference machinery to learn and recognize a set of* ***complex*** *activities under challenging conditions, in particular, unlabeled and partially labelled data with missing observations.* To the best of our knowledge, our system is the first to tackle activity recognition with both duration and hierarchy jointly. We empirically confirm that combinations of both types of information improves the performance, and high accuracy can be achieved with a small number of phases, thus complexity is greatly reduced.

- *A novel scheme for anomaly detection by making use of the Coxian expressiveness to distinguish duration activity patterns.* In this framework, 'normal' activities are learned from training data using the CxSHSMM, the 'abnormal' activity is defined as any substantial deviation from the normal pattern. We develop a method to take the complement of the trained normal models to facilitate detecting abnormalities. This scheme offers several advantages: (a) it focuses on anomalies in the duration patterns of activities, a rather important type of anomaly in the elderly-care[3], but often ignored by the research community, (b) abnormality can be detected at an early stage and monitored when it returns to normality, and thus, alerts can be raised on time and false alarms are minimized, (d) lastly, there is no need to manually construct or train abnormal models.

- *A probabilistic framework to exploit joint hierarchy, structure sharing and duration information for segmenting educational videos into topical units.* This presents the first investigation of duration and hierarchical modeling for this task. We demonstrate that such segmentation processes benefit greatly from the joint modelling.

Finally, even though applied in two specific domains, the models developed in this thesis are generic and have a much wider implications. In fact, any existing work

[3]E.g., stay still unusually long at the dining table could be associated with a heart-attack.

that use semi-Markov modelling can be revised to enjoy computational benefits from our developments.

1.3 Outline of the Thesis

The rest of this thesis is organized as follows. In chapter 2, we provide related background and literature, starting with a revision of Bayesian networks, dynamic Bayesian networks, and exponential family distributions since they form the building blocks. In particular, we focus on the issues of representation, inference and maximum likelihood estimation. The latter part of this chapter provides a review of related applications organized into three main themes: *activity recognition, anomaly detection*, and *video segmentation and annotation.*

Chapter 3 investigates the problem of duration modeling in the HSMM. We start with descriptions of the HSMM followed by a study of the existing modeling choices for state duration in the HSMM. This includes the Multinomial, Poisson and the (continuous) Inverse Gaussian distributions. We show how they can all be represented in the generic exponential family representation. We detail the computation for inference in discrete and continuous cases, provide an analysis on these models and point out the key computational drawbacks that need to be overcome.

Chapter 4 presents the first main theoretical and application contribution. We provide an analysis for the family of discrete phase-type distributions with a focus on the Coxian. We then show how the Coxian can be used to model state duration in the HSMM, essentially leading to the new modelling form, termed the Coxian Hidden Semi-Markov Model (CxHSMM). Model definitions and its DBN representation are provided followed by the inference procedure, including a scaling technique to prevent the numerical underflow problem. Parameter learning is then discussed. Next, we present an application of the CxHSMM to learn and recognize ADLs in a smart home environment. The CxHSMM is evaluated against the MuHSMM (Multinomial duration distribution), the PsHSMM (Poisson), the IgHSMM (Inverse Gaussian) and the standard HMM. We also address model selection by using different numbers of Coxian phases. The performance is judged against classification accuracy, early detection rate and computational time.

Chapter 5 begins by explaining intuitively how hierarchy can be in incorporated into the HSMM to form the Switching Hidden Semi-Markov Model (SHSMM). It then presents the second main theoretical contribution of this thesis – formulation of the Coxian Switching Hidden Semi-Markov Model (CxSHSMM). We provide its model definition, how to represent it as a dynamic Bayesian network and discuss the issues of inference learning. Similar to the CxHSMM case, we address these issues under different settings (e.g. supervised, partially supervised and missing observation). For the sake of completeness and comparison, we also present, in parallel, a study on the SHSMM whose state durations at the bottom layer are modeled by distributions other than the Coxian. Finally, while the hierarchy in the CxSHSMM is shallow, we show that our models can easily be adapted to accommodate arbitrary depth.

In chapter 6 we present our major contributions in terms of applications in two different areas: activity recognition and video segmentation. We conduct three sets of experiments in the first area. We first apply the CxSHSMM to automatically learn, segment, and classify complex ADLs, in which the problem of phase number selection for the Coxian is also addressed. The CxSHSMM performance is compared with that of a MuSHSMM, a two-layer HHMM, and a flat MuHSMM. Given the success of the Coxian parameterization at capturing duration patterns, we next employ the CxSHSMM to construct a novel scheme to detect anomalies in durations of ADLs. The next set of experiments tackles activity recognition in more challenging scenarios (e.g. lossy observations and activities with significant overlapped trajectories) using partially labelled data. The second part of this chapter presents a two-phase framework to detect topic transitions in education-oriented videos. The CxSHSMM is again evaluated against the HHMM, the HSMM and the HMM.

Finally, chapter 7 provides a summary of work in the thesis, its contributions, and discusses potential directions for future work.

Chapter 2

Related Background

In this chapter we review relevant literature and background to the work presented in this thesis. Since this thesis is mainly concerned with dynamic stochastic models and their applications to activity recognition and video segmentation, we plan the reviews around these areas. In most cases our models are directed graphical models, and thus can be viewed in more generic classes of probabilistic models known as Bayesian Networks[1] (BNs), Dynamic Bayesian Networks (DBNs), and exponential families. Sections 2.1 to 2.3 provide a brief account for these models where we highlight the problems of inference and maximum likelihood (ML) estimation in the general setting. The inference and ML estimation are then investigated in more detail when we study the Hidden Markov Model (HMM) in section 2.4. Also, in section 2.4, we briefly analyse the limitations of the HMM at temporal and hierarchical modelings and existing solutions to these problems (section 2.4.5). Section 2.5 then provides a look at the Coxian and Phase-Type distributions in the literature. Sections 2.6 and 2.7 present reviews on activity recognition and anomaly detection, while section 2.8 provides the literature on video segmentation and annotation. Finally, the chapter ends with some closing remarks in section 2.9.

2.1 Bayesian Networks

Bayesian networks (BN) [Jensen, 1996, Pearl, 1988, 1998] is a popular class of probabilistic *directed graphical models* widely used to model *casual relationships without loops* among a set of random variables. Each random variable is represented by a

[1] also alternatively known as probabilistic belief networks or casual probabilistic networks.

vertex[2]. Each random variable takes values from a countable or continuous state space, however, we restrict our model to finite (countable) state sets. The statement on the certainty of the variable state is called evidence, while an exact state assigned to it is an *instantiation*. The casual relationships between random variables are shown by their directed links; however as the graph has no loop (*acyclic*), there is no directed path from a random variable back to itself. If the link is directed from vertex X_i to vertex X_j, then the random variable X_i is a *child* of X_j, and the random variable X_j is a *parent* of X_i. A child can have more than one parent and vice versa, and a set of all parents pointing to a child X_i is denoted as X_{π_i} where π_i denotes the set of indices of the parents of i. By default the *root* has no parent. Formally, a BN is a directed graphical model defined as follows.

Definition 2.1. A Bayesian Network (BN) consists of a set of N random variables $\{X_1, ..., X_N\}$, each taking values from a finite set of mutually exclusive states, and an acyclic directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ defined on them, such that each node in the set of vertices $\mathcal{V}$ corresponds to a random variable X_i. An edge (i, j) in the set of edges $\mathcal{E}$ represents a directed child-parent relation from X_i to X_j. By enumerating all directed links pointing to a node, each X_i is associated with a set of its parent nodes X_{π_i}, and the resulting set $\{X_i, X_{\pi_i}\}$ is assigned with a *conditional distribution* $\Pr(X_i \mid X_{\pi_i})$.

■

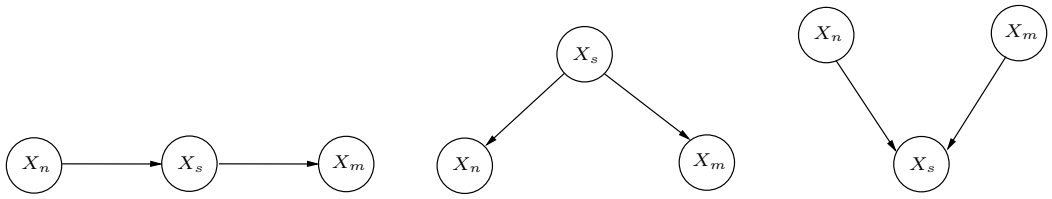
The main property of a BN is that the graphical structure allows joint distribution of these N random variables to be factorized into a product of local conditional probability forms:

$$\Pr(X_1, \ldots, X_N) = \prod_i^N \Pr(X_i \mid X_{\pi_i}) \tag{2.1}$$

where the conditional probabilities $\Pr(X_i \mid X_{\pi_i})$ are specified by the parameters θ of the BN model.

Assuming each X_i is a binary random variable, the normal way to assign the joint probabilities of N variables would take 2^N parameters, which is usually impractical

[2]thus the terms 'vertex' and 'random variable' are used interchangeably.



(a) serial. (b) diverging. (c) converging.

Figure 2.1: The three elemental connections in a BN.

in real world applications due to the exponential blow in parameters when N is high. For the BN, it is clear from Eq. (2.1) that this number reduces to less than $2^k N$, where k is the maximum number of parents each node can have. This saving is a result of built-in independence assumptions in the network.

An important concept in BN is *d-separation* [Jensen, 1996] which allows conditional independence between variables to be asserted directly from the graph structure. Asserting d-separation is based on examining the three elemental connections in BN, namely *serial*, *diverging*, and *converging* as shown in Fig. (2.1). The two variables X_n and X_m are d-separated if for all paths connecting them there is an in-between variable X_s such that one of the two following conditions holds: (i.) the connection is either serial or diverging and X_s is instantiated, $X_n \perp\!\!\!\perp X_m \mid X_s$; or (ii.) the connection is converging and the states of neither X_s nor any of its child nodes are known. Another convenient way to work out conditional independency is to follow the Baye's Ball algorithm as detailed in [Jordan, 2004].

Example 2.1. Fig. (2.2) shows a toy example of a BN in smart-home context. The activity monitor M asserts the current state of activities of daily livings (ADLs) in the house. If any abnormalities occur it will trigger the home alert system A, which will in turn send messages simultaneously to carer C and the emergency monitoring center E. The elderly occupant O can also send a message to the carer C if she needs to. This graph depicts all three different connections: serial ($M-A-E$ and $M-A-C$), converging ($O-C-A$), and diverging ($C-A-E$). The state of ADLs in the house (M) affects the emergency monitoring center E through the home alert system A (serial connection). However, if A is turned on (its state is known), the emergency center E then knows that abnormality is detected, and thus there is no need to know the state of M: $E \perp M \mid A$. Similarly, the carer C can affect the

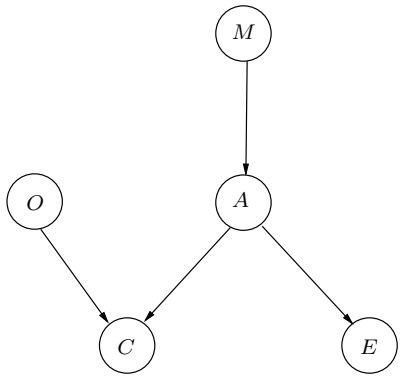


Figure 2.2: An example of BNs.

emergency center E through the home alert system A (diverging connection). For instance, the knowledge of C just receiving a message will increase the possibility of alert system A being triggered, thus, it will increase the chance of the emergency center E also getting a message. Nevertheless, if the state of A is known to be off, then C probably receives its message from the occupant O, and the chance of E expecting a message remains unchanged: $E \perp\!\!\!\perp C \mid A$. Finally, we look at the connection $O-C-A$. If the state of the carer C is known, then there is no affiliation between the occupant O and the home alert system A. On the other hand, if C receives a message then it must be sent from either O or A; equivalently O and A become dependent.

⊞

The discussion on d-separation shows how evidence is propagated through the network. The next question is how to quantitatively update the certainty of the states of some nodes X_h given the exact states of other nodes X_o, i.e. to compute the conditional probability $\Pr(X_h \mid X_o)$. This is called the problem of *inference*. Fortunately, the factorization of the joint probability into local conditional probabilities in Eq. (2.1) allows inference to be performed with the *junction tree* algorithm [Jensen, 1996] consisting of three steps: the BN is firstly converted into an undirected graph by a linking all nodes which have a common child (moralization step) and dropping arrows on directed edges; the undirected graph is then triangulated and converted into a junction tree, for example by Kruskal's algorithm [see Jensen, 1996, chapter 4]; and lastly, a message passing procedure is conducted on the resulting clique tree. More details can be found in [Jordan, 2004, Jensen, 1996, Pearl, 1988]. However,

except for special cases, the junction tree algorithm is often problematic in practice due to the large clique size during triangulation step.

A central issue in BN is how to *learn* the model parameters θ, commonly known as *parameter estimation* problem. We consider two different cases: (a) the BN is *fully observed* and (b) the BN has *hidden* (*latent*) variables. In the fully observed case learning is done via Maximum Likelihood (ML) estimation; whereas in the hidden case the Expectation Maximization (EM) algorithm [Dempster et al., 1977] is used. However, as BN belongs to the Exponential Family distributions [Dan, 1998], perhaps a better way to express the parameter estimation problem in BN is to view it as a form of an Exponential Family. That is because the Exponential Family provides the sufficient statistics readily and thus facilitates the learning process. Our next section is dedicated to the Exponential Family and ML/EM estimation in this context.

2.2 The Exponential Family

Exponential Family distributions arise popularly in many problems and encompass a rich class of distributions such as Bernoulli, Poisson and Multinomial in the discrete case as well as Gaussian, Beta and Gamma in the continuous case. The list is much longer and indeed more sophisticated graphical models such as Bayesian networks, Markov random fields, conditional random fields can also be represented in Exponential Family forms. This class of distributions possesses some very important properties that make them useful.

Definition 2.2. A family of probability distributions is said to belong to an Exponential Family if it can be expressed in the following form:

$$\Pr(x \mid \theta) = h(x) \exp\left\{\theta^{\mathsf{T}} T(x) - A(\theta)\right\} \tag{2.2}$$

where θ is the parameter (often referred to as *natural* or *canonical* parameter), $T(x)$ is the vector of sufficient statistics, $\theta^{\mathsf{T}} T(x)$ is the usual inner product, $A(\theta)$ is the log-partition function serving as the normalization term to make $\Pr(x \mid \theta)$ a proper probability distribution (sum or integration to one), and $h(x)$ is the base function and independent of the parameter θ.

■

The log-partition arises from Eq. (2.2) as:

$$A(\theta) = \log \int_x h(x) \exp\left\{\theta^\mathsf{T} T(x)\right\} dx \tag{2.3}$$

and in the discrete case, the integration sign is simply replaced by the sum. Let d be the dimension of the random vector x, then the parameter space Θ is the set of all parameters θ such that Eq. (2.2) is defined, which in turn implies that $A(\theta)$ in Eq. (2.3) is finite:

$$\Theta = \left\{\theta \in \mathbb{R}^d \mid A(\theta) < \infty\right\}$$

Example 2.2. As a simple example, the (discrete) Poisson distribution can be expressed in Exponential Family form as follows:

$$\begin{aligned} \Pr(x \mid \lambda) &= \frac{\lambda^x \exp\{-\lambda\}}{x!} \\ &= \frac{1}{x!} \exp\{x \log \lambda - \lambda\} \end{aligned} \tag{2.4}$$

with

the natural parameter $\theta = \log \lambda$;

the sufficient statistic $T(x) = x$;

the log-partition function: $A(\theta) = \lambda = \exp\{\theta\}$;

and the base function: $h(x) = \frac{1}{x!}$.

⊞

Example 2.3. Another simple example is the (continuous) univariate Gaussian distribution:

$$\begin{aligned} \Pr(x \mid \mu, \sigma) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \\ &= \frac{1}{\sqrt{2\pi}} \exp\left\{\frac{\mu}{\sigma^2}x - \frac{1}{2\sigma^2}x^2 - \frac{\mu^2}{2\sigma^2} - \log \sigma\right\} \end{aligned} \tag{2.5}$$

with

the natural parameter $\theta = \begin{bmatrix} \frac{\mu}{\sigma^2} \\ -\frac{1}{2\sigma^2} \end{bmatrix}$;

the sufficient statistic $T(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix}$;

the log-partition function $A(\theta) = \frac{\mu^2}{2\sigma^2} + \log \sigma = -\frac{\theta_1^2}{4\theta_2} - \frac{1}{2}\log\{-2\theta_2\}$;

and the base function $h(x) = \frac{1}{\sqrt{2\pi}}$.

⊞

From the definition in Eq. (2.2) it is easy to see that the product of two or more distributions in the Exponential Family also belong to the Exponential Family. Thus, if each local conditional distribution in a BN can be expressed in an Exponential Family form, then so can the joint distribution of the whole network. We use the simple network in example (2.4) to illustrate this concept.

Example 2.4. Consider a simple network in Fig. (2.3) where X and Y are binary random variables and further let $\Pr(X=0) = a$, $\Pr(Y=0 \mid X=0) = b$, and $\Pr(Y=1 \mid X=1) = c$, then the joint probability is given as:

$$\Pr(X,Y) = \Pr(X)\Pr(Y \mid X) \tag{2.6}$$

Let $\delta_X^{(i)}$ be the event $\{X=i\}$ (i.e. $\delta_X^{(i)} = 1$ if $X=i$, and $\delta_X^{(i)} = 0$ otherwise), the local probabilities in Eq. (2.6) can be expressed accordingly as:

$$\begin{aligned} \Pr(X) &= \prod_{x=0,1} \Pr(X=x)^{\delta_X^{(x)}} \\ &= \exp\left\{\sum_x \delta_X^{(x)} \log \Pr(X=x)\right\} \\ &= \exp\left\{\delta_X^{(0)} \log a + \delta_X^{(1)} \log(1-a)\right\} \end{aligned} \tag{2.7}$$

Thus, $\Pr(X)$ belongs to Exponential Family with the natural parameter $\theta_1 = [\log a \quad \log(1-a)]^\mathsf{T}$, and the sufficient statistic $\mathrm{T}_1(X) = \left[\delta_X^{(0)} \quad \delta_X^{(1)}\right]^\mathsf{T}$. Similarly,

$$\begin{aligned} \Pr(Y \mid X) &= \prod_{x=0,1}\prod_{y=0,1} \Pr(Y=y \mid X=x)^{\delta_X^{(x)}\delta_Y^{(y)}} = \exp\left\{\sum_{x,y} \delta_X^{(x)}\delta_Y^{(y)} \log \Pr(Y \mid X)\right\} \\ &= \exp\left\{\delta_X^{(0)}\delta_Y^{(0)} \log b + \delta_X^{(0)}\delta_Y^{(1)} \log(1-b) + \delta_X^{(1)}\delta_Y^{(1)} \log c + \delta_X^{(1)}\delta_Y^{(0)} \log(1-c)\right\} \end{aligned} \tag{2.8}$$

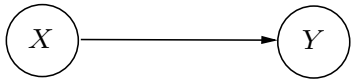


Figure 2.3: The BN of example (2.4).

That means $\Pr(Y \mid X)$ is an Exponential Family distribution with the natural parameter $\theta_2 = [\log b \;\; \log(1-b) \;\; \log c \;\; \log(1-c)]^{\mathsf{T}}$, and the sufficient statistic $\mathrm{T}_2(X,Y) = \left[\delta_X^{(0)}\delta_Y^{(0)} \;\; \delta_X^{(0)}\delta_Y^{(1)} \;\; \delta_X^{(1)}\delta_Y^{(1)} \;\; \delta_X^{(1)}\delta_Y^{(0)}\right]^{\mathsf{T}}$. It then follows that the joint probability $\Pr(X,Y)$ belongs to the Exponential Family with the natural parameter $\theta = [\theta_1 \;\; \theta_2]^{\mathsf{T}}$, and the sufficient statistic $T(X,Y) = [\mathrm{T}_1(X) \;\; \mathrm{T}_2(X,Y)]^{\mathsf{T}}$ as shown below:

$$\begin{aligned}\Pr(X,Y) &= \exp\{\log\Pr(X) + \log\Pr(Y \mid X)\} \\ &= \exp\Big\{\delta_X^{(0)}\log a + \delta_X^{(1)}\log(1-a) + \delta_X^{(0)}\delta_Y^{(0)}\log b \\ &\qquad + \delta_X^{(0)}\delta_Y^{(1)}\log(1-b) + \delta_X^{(1)}\delta_Y^{(1)}\log c + \delta_X^{(1)}\delta_Y^{(0)}\log(1-c)\Big\} \qquad (2.9)\end{aligned}$$

⊞

2.2.1 Maximum-likelihood with fully observed model

Assume for clarity in the discrete case that each $X_i \in \{1,\ldots,K\} = \mathcal{X}_i$ and further let $\mathcal{X}_{\pi_i}$ be the set of all values that its parent X_{π_i} can take, then the joint distribution of the BN is given as:

$$\begin{aligned}\Pr(X_1,\ldots,X_N) &= \exp\left\{\sum_{i=1}^{N}\log\Pr(X_i \mid X_{\pi_i})\right\} \\ &= \exp\left\{\sum_{i=1}^{N}\log\left\{\prod_{k\in\mathcal{X}_i}\prod_{v\in\mathcal{X}_{\pi_i}}\delta_{X_i}^{(k)}\delta_{X_{\pi_i}}^{(v)}\Pr(X_i = k \mid X_{\pi_i} = v)\right\}\right\} \\ &= \exp\left\{\sum_{i=1}^{N}\left[\sum_k\sum_v \delta_{X_i}^{(k)}\delta_{X_{\pi_i}}^{(v)}\right]\log\theta_{k,v}^{i}\right\} \qquad (2.10)\end{aligned}$$

where $\theta_{k,v}^{i}$ denotes $\Pr(X_i = k \mid X_{\pi_i} = v)$ (consequently $\sum_k \theta_{k,v}^{i} = 1$). Eq. (2.10) shows that the joint probability distribution belongs to the Exponential Family with the canonical parameters $\log\theta_{k,v}^{i}$, and the sufficient statistics are a set of identity

functions $T\left(\theta^i_{k,v}\right) = \delta^{(k)}_{X_i}\delta^{(v)}_{X_{\pi_i}}$:

$$\Pr\left(X_1, \ldots, X_N\right) = \exp\left\{\sum_{i=1}^{N}\sum_{v}\sum_{k} T\left(\theta^i_{k,v}\right)\log\theta^i_{k,v}\right\} \tag{2.11}$$

Given this form it is easy to show that Maximum Likelihood (ML) estimation in the fully observed case is *decoupled* into local maximization involving each X_i and its parent X_{π_i}. Using Lagrange multipliers [Arfken, 1985] (theorem (2.1)) on the constraint that each local conditional distribution must sum to one, ML in this case is equivalent to counting the frequency of each local configuration in the observed data.

Theorem 2.1. *Given two vectors* $\boldsymbol{a} = [a_1 \ldots a_N]^\mathsf{T}$, $\boldsymbol{z} = [z_1 \ldots z_N]^\mathsf{T}$ *and an objective function* $f\left(\boldsymbol{z}\right) = \sum_{n=1}^{N} a_n \log z_n$, *the solution to the optimization problem* $\hat{\boldsymbol{z}} = \underset{z}{argmax}\ f\left(\boldsymbol{z}\right)$ *subject to the constraint* $\sum_{n=1}^{N} z_n = 1$ *is* $\hat{z_n} = a_n \diagup \sum_{n=1}^{N} a_n$.

Proof. Adding the Lagrange multiplier λ into function $f\left(\boldsymbol{z}\right)$ results in:

$$f\left(\boldsymbol{z}\right) = \sum_{n=1}^{N} a_n \log z_n + \lambda\left(1 - \sum_{n=1}^{N} z_n\right) \tag{2.12}$$

Taking the derivative of Eq. (2.12) with respect to z_n:

$$\frac{\delta f\left(\boldsymbol{z}\right)}{\delta z_n} = \frac{a_n}{z_n} - \lambda$$

Setting the derivative to zero:

$$\begin{aligned} & \frac{a_n}{z_n} - \lambda = 0 \\ \Rightarrow \qquad & z_n = \frac{a_n}{\lambda} \end{aligned} \tag{2.13}$$

Summing both sides of Eq. (2.13) over all values of z_n and a_n:

$$\begin{aligned} & \sum_{n=1}^{N} z_n = \sum_{n=1}^{N} \frac{a_n}{\lambda} \\ \Rightarrow \qquad & \lambda = \frac{\sum_{n=1}^{N} a_n}{\sum_{n=1}^{N} z_n} = \sum_{n=1}^{N} a_n \end{aligned} \tag{2.14}$$

Substituting λ from Eq. (2.14) into Eq. (2.13) leads to the estimation formula for z_n:

$$\hat{z_n} = \frac{a_n}{\sum_{n=1}^{N} a_n} \tag{2.15}$$

□

2.2.1.1 The Maximum-Likelihood algorithm

Formally, the solution for the ML estimation in the fully observed case is stated as follows. Given the observed data consists of M iid sequences $D = \left\{D^{(1)}, \ldots, D^{(M)}\right\}$ where $D^{(m)}$ is an instantiation of $\{X_1, \ldots, X_N\}$, the complete log likelihood $\mathcal{L}^C(D \mid \theta)$ is computed as:

$$\mathcal{L}^C(D \mid \theta) = \log \prod_{m=1}^{M} \Pr\left(D^{(m)} \mid \theta\right) = \sum_{m=1}^{M} \log \Pr\left(D^{(m)} \mid \theta\right) \tag{2.16}$$

$$= \sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{v \in \mathcal{X}_{\pi_i}} \sum_{k \in \mathcal{X}_i} \delta^{(k)}_{X_i^{(m)}} \delta^{(v)}_{X_{\pi_i}^{(m)}} \log \theta^i_{k,v} \tag{2.17}$$

$$= \sum_{i=1}^{N} \sum_{v \in \mathcal{X}_{\pi_i}} \sum_{k \in \mathcal{X}_i} \sum_{m=1}^{M} \delta^{(k)}_{X_i^{(m)}} \delta^{(v)}_{X_{\pi_i}^{(m)}} \log \theta^i_{k,v} \tag{2.18}$$

$$= \sum_{i=1}^{N} \sum_{v \in \mathcal{X}_{\pi_i}} \sum_{k \in \mathcal{X}_i} T\left(\theta^i_{k,v}\right) \log \theta^i_{k,v} \tag{2.19}$$

where the step from Eq. (2.16) to Eq. (2.17) is based on the exponential form of the joint distribution in Eq. (2.10), and

$$T\left(\theta^i_{k,v}\right) = \sum_{m=1}^{M} \delta^{(k)}_{X_i^{(m)}} \delta^{(v)}_{X_{\pi_i}^{(m)}} \tag{2.20}$$

is the sufficient statistic of $\log \theta^i_{k,v}$. Thus, the sufficient statistic of iid observations is equal to the *sum* of individual sufficient statistics. Given the above expression, the ML solution can be now solved for each local conditional probability using theorem (2.1):

$$\hat{\theta}^i_{k,v} = \frac{T\left(\theta^i_{k,v}\right)}{\sum_{k \in \mathcal{X}_i} T\left(\theta^i_{k,v}\right)} \tag{2.21}$$

and the sufficient statistic $T\left(\theta^i_{k,v}\right)$ is the count of configurations. Assuming $m(x_i)$ to be the count that X_i is assigned to $x_i \in \{1, \ldots, K\}$ and similarly $m(x_{\pi_i})$ be the configuration count for its parents, then the ML solution in Eq. (2.21) can be re-written as:

$$\hat{\theta}^i_{k,v} = \frac{m(x_i = k, x_{\pi_i} = v)}{\sum_k m(x_i = k, x_{\pi_i} = v)} \tag{2.22}$$

Also note that as the ML solution is based on the empirical counts, it is vulnerable to overfitting.

We now come back to example (2.4) to show how ML solutions can be computed for this particular BN. Suppose we have seven idd observations of $\{X, Y\}$: $D = \{\{1,0\},\{0,1\},\{0,0\},\{1,1\},\{0,1\},\{1,1\},\{1,1\}\}$. Using Eq. (2.22), it then follows that:

$$\hat{a} = \frac{m(x=0)}{\sum_{k=0,1} m(x=k)} = \frac{3}{7}$$

$$\hat{b} = \frac{m(y=0, x=0)}{\sum_{k=0,1} m(y=k, x=0)} = \frac{1}{3}$$

$$\hat{c} = \frac{m(y=1, x=1)}{\sum_{k=0,1} m(y=k, x=1)} = \frac{3}{4}$$

2.2.2 Maximum-Likelihood with Hidden Variables

The ML problem becomes more challenging with the presence of latent variables as it can no longer be straightforwardly decoupled into local maximization. To overcome this, the Expectation-Maximization (EM) algorithm [Dempster et al., 1977] is used, which consists of two iterative steps: the E-step and the M-step as briefly outlined in the following sub-section. For full explanation on EM, readers are referred to references such as [Dempster et al., 1977, Prescher, 2003, Jordan, 2004].

2.2.2.1 The Expectation-Maximization algorithm

The set of variables $\mathbf{X} = \{X_1, \ldots, X_N\}$ is partitioned into two subsets: a subset of observed variables $\mathbf{O}$, and a subset of latent variables $\mathbf{H}$, then our objective in this case is to maximize the function:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \Pr(\mathbf{O} \mid \theta) \tag{2.23}$$

To utilize the advantage of decoupling the ML problem into local maximization as in the fully observed case, we first have to "fill in" the unobserved variables [Jordan,

2004]. Let $\mathcal{L}(\mathbf{O} \mid \theta)$ be the incomplete log likelihood, then:

$$\begin{aligned}
\mathcal{L}(\mathbf{O} \mid \theta) &= \log\{\Pr(\mathbf{O} \mid \theta)\} = \log\left\{\sum_{\mathbf{H}} \Pr(\mathbf{O}, \mathbf{H} \mid \theta)\right\} \\
&= \log\left\{\sum_{\mathbf{H}} \left(\frac{\Pr(\mathbf{O}, \mathbf{H} \mid \theta)}{Q(\mathbf{H} \mid \mathbf{O})} Q(\mathbf{H} \mid \mathbf{O})\right)\right\} \\
&\geq \sum_{\mathbf{H}} Q(\mathbf{H} \mid \mathbf{O}) \log\left\{\frac{\Pr(\mathbf{O}, \mathbf{H} \mid \theta)}{Q(\mathbf{H} \mid \mathbf{O})}\right\} \qquad (2.24) \\
&= \sum_{\mathbf{H}} Q(\mathbf{H} \mid \mathbf{O}) \log \Pr(\mathbf{O}, \mathbf{H} \mid \theta) - \sum_{\mathbf{H}} Q(\mathbf{H} \mid \mathbf{O}) \log Q(\mathbf{H} \mid \mathbf{O}) \\
&= \left\langle \mathcal{L}^C(\mathbf{O}, \mathbf{H} \mid \theta) \right\rangle_Q - \sum_{\mathbf{H}} Q(\mathbf{H} \mid \mathbf{O}) \log Q(\mathbf{H} \mid \mathbf{O}) \qquad (2.25) \\
&\triangleq \digamma(Q, \theta)
\end{aligned}$$

where the notation $\left\langle \mathcal{L}^C(\mathbf{O}, \mathbf{H} \mid \theta) \right\rangle_Q$ denotes the expected value of $\log \Pr(\mathbf{O}, \mathbf{H} \mid \theta)$ over the function $Q(\mathbf{H} \mid \mathbf{O})$. Eq. (2.24) is obtained from Jensen's inequality resulting directly from the concavity of the log function. The EM algorithm is then preformed on the lower bound of the incomplete log likelihood:

$$\text{E-step:} \quad Q^{(t+1)} = \underset{Q}{\operatorname{argmax}} \digamma\left(Q, \theta^{(t)}\right) \qquad (2.26)$$

$$\text{M-step:} \quad \theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} \digamma\left(Q^{(t+1)}, \theta\right) \qquad (2.27)$$

As pointed out in [Jordan, 2004], the solution for the E-step is $Q^{(t+1)} = \Pr\left(\mathbf{H} \mid \mathbf{O}, \theta^{(t)}\right)$ since:

$$\begin{aligned}
\digamma\left(\Pr\left(\mathbf{H} \mid \mathbf{O}, \theta^{(t)}\right), \theta^{(t)}\right) &= \sum_{\mathbf{H}} \Pr\left(\mathbf{H} \mid \mathbf{O}, \theta^{(t)}\right) \log\left\{\frac{\Pr\left(\mathbf{O}, \mathbf{H} \mid \theta^{(t)}\right)}{\Pr\left(\mathbf{H} \mid \mathbf{O}, \theta^{(t)}\right)}\right\} \\
&= \sum_{\mathbf{H}} \Pr\left(\mathbf{H} \mid \mathbf{O}, \theta^{(t)}\right) \log \Pr\left(\mathbf{O} \mid \theta^{(t)}\right) \\
&= \mathcal{L}\left(\mathbf{O} \mid \theta^{(t)}\right)
\end{aligned}$$

Also we note that maximizing $\digamma\left(\Pr\left(\mathbf{H} \mid \mathbf{O}, \theta^{(t)}\right), \theta\right)$ with respect to θ is equal to maximizing the expected complete log likelihood $\left\langle \mathcal{L}^C(\mathbf{O}, \mathbf{H} \mid \theta) \right\rangle_{\Pr\left(\mathbf{H} \mid \mathbf{O}, \theta^{(t)}\right)}$ because the second term in Eq. (2.25) does not depend on θ. Thus, the M-step at iteration $t+1$ is essentially equivalent to:

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} \left\langle \mathcal{L}^C(\mathbf{O}, \mathbf{H} \mid \theta) \right\rangle_{\Pr\left(\mathbf{H} \mid \mathbf{O}, \theta^{(t)}\right)} \qquad (2.28)$$

and can be solved locally.

2.2.2.2 The expected sufficient statistic and ML solution

As shown in Eq. (2.28), the maximization step in the presence of hidden variables is performed on the expected log likelihood with respect to the probability of hidden variables given observed ones, instead of on the log likelihood itself. We now see how this change affects the sufficient statistic.

Using the expression of the complete log likelihood in Eq. (2.19) the expected complete log likelihood can be written as:

$$\begin{aligned}\left\langle \mathcal{L}^C\left(\mathbf{O},\mathbf{H} \mid \theta\right)\right\rangle_{\Pr(\mathbf{H} \mid \mathbf{O},\theta)} &= \left\langle \sum_{i=1}^{N}\sum_{v\in\mathcal{X}_{\pi_i}}\sum_{k\in\mathcal{X}_i} T\left(\theta^i_{k,v}\right)\log\theta^i_{k,v}\right\rangle_{\Pr(\mathbf{H} \mid \mathbf{O},\theta)} \\ &= \sum_{i=1}^{N}\sum_{v\in\mathcal{X}_{\pi_i}}\sum_{k\in\mathcal{X}_i}\left\langle T\left(\theta^i_{k,v}\right)\right\rangle_{\Pr(\mathbf{H} \mid \mathbf{O},\theta)}\log\theta^i_{k,v} \end{aligned} \tag{2.29}$$

The above equation shows that the sufficient statistic now becomes the expected sufficient statistic $\left\langle T\left(\theta^i_{k,v}\right)\right\rangle_{\Pr(\mathbf{H} \mid \mathbf{O},\theta)}$, which is henceforth referred to as $\left\langle T\left(\theta^i_{k,v}\right)\right\rangle$ for simplicity:

$$\left\langle T\left(\theta^i_{k,v}\right)\right\rangle = \sum_{\mathbf{H}}\Pr\left(\mathbf{H} \mid \mathbf{O},\theta\right)T\left(\theta^i_{k,v}\right) = \sum_{\mathbf{H}}\Pr\left(\mathbf{H} \mid \mathbf{O},\theta\right)\sum_{m=1}^{M}\delta^{(k)}_{X^{(m)}_i}\delta^{(v)}_{X^{(m)}_{\pi_i}} \tag{2.30}$$

Similar to the fully observed case, the ML solution for the expected complete log likelihood is solved locally using the Lagrange multiplier (theorem (2.1)), which leads to:

$$\hat{\theta}^i_{k,v} = \frac{\left\langle T\left(\theta^i_{i,k}\right)\right\rangle}{\sum_{k\in\mathcal{X}_i}\left\langle T\left(\theta^i_{i,k}\right)\right\rangle} \tag{2.31}$$

2.3 Dynamic Bayesian Networks

Having shown that BN in the general case can be expressed in the Exponential Family form, we now have a clearer picture of the structure of the solution for parameter estimation. The nature of the data we work with in the thesis is sequential. The models developed in this work need to be dynamic and we frame these developments in a more generic class of probabilistic models known as Dynamic Bayesian Networks (DBN) [Dean and Kanazawa, 1989, Murphy, 2002].

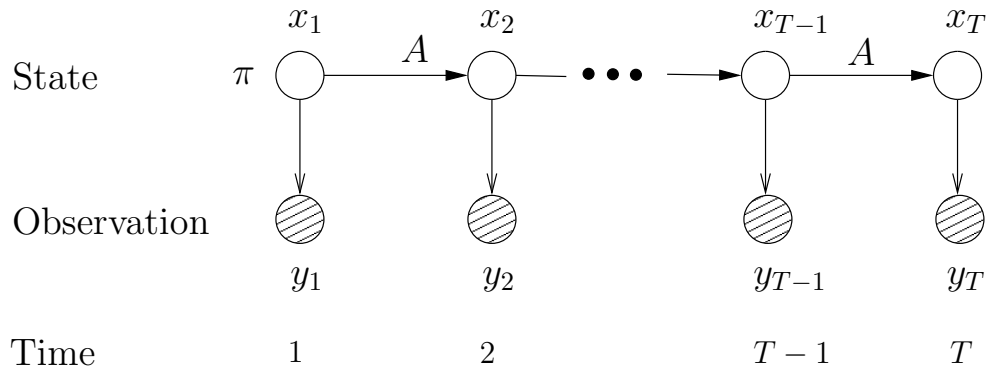


Figure 2.4: DBN representation for the HMM.

Intuitively, a DBN is a Bayesian network defined for temporal data in which a single Bayesian network structure $\mathcal{B}$ is replicated over time. Let $\mathcal{V}_t$ be the set of an amalgamated set of random variables at time t and let $\mathcal{B}_t(\mathcal{V}_t)$ be network structure at time t defined for $\mathcal{V}_t$, then the network structure for the next time slice $\mathcal{B}_{t+1}(\mathcal{V}_{t+1})$ is *identical* to $\mathcal{B}_t(\mathcal{V}_t)$ and thus it allows the use of a single $\mathcal{B}(\mathcal{V})$ to denote the network structure at any time slice. To complete the definition for DBN, we further specify a transition network structure that flows from $\mathcal{V}_t$ to $\mathcal{V}_{t+1}$ and we denote as $T(\mathcal{V}_t \rightarrow \mathcal{V}_{t+1})$. For example, the simplest form of DBN is the Hidden Markov Models (HMM) shown in Fig. (2.4). Each time slice $\mathcal{V}_t$ consists of the state X_t and observation Y_t; the one-slide $\mathcal{B}(\mathcal{V}_t)$ has a simple structure of $X_t \rightarrow Y_y$ and the two-slide $T(Z_t \rightarrow Z_{t+1})$ has the simple structure[3] of $X_t \rightarrow X_{t+1}$. With respect to parameter specification, DBN assigns two types of probability: the initial probability model at the first time slice for $\Pr(\mathcal{V}_1)$ and the transition probability for $\Pr(\mathcal{V}_{t+1} \mid \mathcal{V}_t)$, and this transition model is *copied*[4] over time which is known more technically as a form of parameter *tying* across time (e.g., see [Phung, 2005b] for a discussion on parameter tying and relationship to Exponential Family).

DBN has the same demands for inference and learning as in the BN, but since it is dynamic the techniques take the dynamic nature into account. The inference problem includes computing the *filtering* distribution $\Pr(H_t \mid o_{1:t})$, which is the probability of current hidden state H_t given the observation sequence $o_{1:t}$; the *smoothing* distributions $\Pr(H_t \mid o_{1:T>t})$; and the prediction distribution $\Pr(H_t \mid o_{1:\tau<t})$. For learning,

[3]Since Y_t and Y_{t+1} have no role in the transition network while only X_t and X_{t+1} do, Murphy [Murphy, 2002] refers to X_t as the interface nodes.

[4]To be more precise, we are considering *time-invariant* DBN here.

we need to compute the parent-child distribution $\Pr\left(H_t, X_{\pi_{H_t}} \mid o_{1:T}\right)$ where any of of the parents $X_{\pi_{H_t}}$ could be in the time slice $t-1$. As any discrete-state DBN can be converted into a HMM [Murphy, 2002], its inference can be effectively done by the well-known forward/backward procedures applied in the HMM [Rabiner, 1989], which are detailed in section 2.4.3. Parameter estimation (ML and EM) in the DBN is similar to that of the BN, except here we have to collect the sufficient statistics of all nodes which share the same parameters over time. There is, however, an exemption for the parameters showing the initial condition of the network ($\Pr\left(\mathcal{V}_1\right)$), whose sufficient statistics only arise at the first time slice.

For a comprehensive survey on many aspects of DBN we refer readers to the excellent work of [Murphy, 2002]. In particular, details on several approximate inference techniques including the Boyen-Koller (BK), factor frontier (FF), particle filters (PF) and Rao-Blackwellised particle filter (RBPF) can be found therein. Even though not discussed fully, these techniques can be readily applied for the models developed in this thesis when computational speed is a more pressing issue, e.g., in large-scale systems, but of course, at the trade-off of accuracy.

2.4 The Hidden Markov Models

This section is devoted to familiarizing the readers with the Hidden Markov Model (HMM) as it is the baseline for all probabilistic models investigated in this thesis.

2.4.1 Model and definition

The well-known HMM [Rabiner, 1989] is defined as follows. The state space is a set of discrete states, numbered sequentially: $Q = \{1, \ldots, |Q|\}$, and elements in Q are referred as i, j. The initial probability π_i specifies the starting state of the Markov chain defined over states in Q, while the transition matrix A_{ij} governs the transitions within states. At each time point an observation v in the alphabet set V is generated with an emission probability $B_{v|i}$ with i being the current state[5]. Thus, a HMM is completely parameterized by $\theta_{HMM} \triangleq \{\pi,\ A,\ B\}$, and stochastic constraints require $\sum_{i \in Q} \pi_i = 1$, $\sum_{j \in Q} A_{ij} = 1$, and $\sum_{v \in V} B_{v|i} = 1$. Table (2.1) shows a summary of

[5]Note that when the observation is continuous, the emission probability is usually modeled by a mixture of Gaussians. However, we only consider discrete observations in our work.

Symbols	Meanings
Q	The state space includes $\|Q\|$ mutually exclusive state: $Q = \{1,\ 2,\ \ldots,\ \|Q\|\}$.
V	The observation space consists of $\|V\|$ distinguished alphabets, $V = \{1,\ 2,\ \ldots,\ \|V\|\}$.
π_i	The probability that the semi-Markov chain will start with state i, $\sum_{i\in Q} \pi_i = 1$.
A_{ij}	The probability that the next state will be j given the . current state is i, $\sum_{j\in Q} A_{ij} = 1$
$B_{v\|i}$	The probability that an alphabet v is generated given the current state is i, $\sum_{v\in V} B_{v\|i} = 1$.
θ_{HMM}	The HMM parameter set: $\theta_{\text{HMM}} \triangleq \{\pi,\ A,\ B\}$.

Table 2.1: Parameters of a HMM.

HMM parameters. Finally, it is important to note that in the HMM, the probability $D_i(d)$ for which a state i remains the same for a positive duration d, has a geometric distribution: $D_i(d) \sim f_{\text{Geom}\left(1-A_{ii}\right)}(d) = (A_{ii})^{d-1}(1 - A_{ii})$.

2.4.2 DBN Representation

The HMM is a simple generative model which can be viewed as a special case of the DBN (section 2.3). Figure (2.4) in section 2.3 shows the DBN representation of a HMM when unrolled over T time slices. Each time slice t has a simple BN (section 2.1) showing the environment at the specific time t. Each BN consists of a state variable x_t taking a single value from the state set Q, and an observation y_t generated from the state x_t. In general, the state variables $x_{1:T}$ are hidden and represented by clear nodes, the observations $y_{1:T}$ are observed and represented by shaded nodes.

Fig. (2.5) shows the three core cliques in the DBN representation of the HMM. All the three parameters $(\pi,\ A,\ B)$ in θ_{HMM} can be transformed to causal relationship in these cliques:

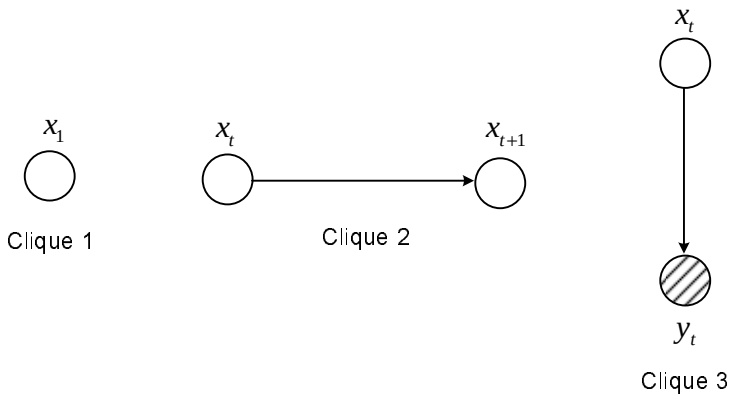


Figure 2.5: Cliques of the HMM.

$$\text{Clique 1:} \qquad \pi_i \triangleq \Pr(x_1 = i), \quad \sum_{i \in Q} x_i = 1.$$

$$\text{Clique 2:} \qquad A_{ij} \triangleq \Pr\left(x_{t+1} = j \mid x_t = i\right), \quad \sum_{j \in Q} A_{ij} = 1.$$

$$\text{Clique 3:} \qquad B_{v|i} \triangleq \Pr(y_t = v \mid x_t = i), \quad \sum_{v \in V} B_{v|i} = 1.$$

2.4.3 Inference

Inference in the HMM includes the computation of a forward variable $\alpha_t(i) \triangleq \Pr(x_t^i, y_{1:t})$, and a backward variable $\beta_t(i) \triangleq \Pr(y_{t+1:T} \mid x_t^i)$ that in turn are used to compute the two smoothing distributions required in learning: $\gamma_t(i) \triangleq \Pr(x_t^i \mid y_{1:T})$, and $\xi_t(i,j) \triangleq \Pr(x_t^i, x_{t+1}^j \mid y_{1:T})$. Note that for simplicity we have written s_t^i as shorthand for the event $s_t = i$ with s_t being an arbitrary node in the DBN representation.

Computing the forward and backward variables

The forward/backward variables can be computed recursively [Rabiner, 1989] via dynamic programming by decomposing the variables using the conditional independence property of the Markov chain. The recursive computation for the forward

variable is as follows:

$$\alpha_{t+1}(j) \triangleq \Pr(x_{t+1}^j, y_{1:t+1}) \tag{2.32}$$
$$= \sum_i \Pr(x_t^i, x_{t+1}^j, y_{1:t+1}) \tag{2.33}$$
$$= \sum_i \Pr(y_{t+1} \mid x_{t+1}^j) \Pr(x_{t+1}^j \mid x_t^i) \Pr(x_t^i, y_{1:t}) \tag{2.34}$$
$$= B_{y_{t+1}|j} \sum_i A_{ij} \alpha_t(i) \tag{2.35}$$

The step from equation (2.33) to equation (2.34) is a result of the conditional independency in the Markov chain (Fig. (2.4)). The variable x_{t+1} separates the observation y_{t+1} from all previous values: $y_{t+1} \perp \{x_t, y_{1:t}\} \mid x_{t+1}$, which leads to $\Pr(y_{t+1} \mid x_{t+1}, x_t, y_{1:t}) = \Pr(y_{t+1} \mid x_{t+1})$. Similarly the state variable x_t isolates x_{t+1} from all previous observations $y_{1:t}$: $\Pr(x_{t+1} \mid x_t, y_{1:t}) = \Pr(x_{t+1} \mid x_t)$. Since $\alpha_t(i)$ is a forward recursion, it needs a definition at time $t = 1$ to start with:

$$\alpha_1(i) = \Pr\left(x_1^i, y_1\right) = \Pr\left(y_1 \mid x_1^i\right) \Pr\left(x_1^i\right) = B_{y_1|i} \pi_i$$

Note that the likelihood is readily available from the forward variable as: $\Pr(y_{1:T}) = \sum_i \Pr(x_T^i, y_{1:T}) = \sum_i \alpha_T(i)$. The backward variable $\beta_t(i)$ can be computed recursively in an analogous manner:

$$\beta_t(i) \triangleq \Pr\left(y_{t+1:T} \mid x_t^i\right) \tag{2.36}$$
$$= \sum_j \Pr\left(x_{t+1}^j, y_{t+1:T} \mid x_t^i\right) \tag{2.37}$$
$$= \sum_j \Pr\left(y_{t+1} \mid x_{t+1}^j\right) \Pr\left(y_{t+2:T} \mid x_{t+1}^j\right) \Pr\left(x_{t+1}^j \mid x_t^i\right) \tag{2.38}$$
$$= \sum_j B_{y_{t+1}|j} A_{ij} \beta_{t+1}(j) \tag{2.39}$$

The backward calculation needs an initialization at time $t = T$:

$$\beta_T(i) = \Pr\left(y_{T+1:T} \mid x_T^i\right) = \Pr\left(\varnothing \mid x_T^i\right) = 1$$

The scaled forward and backward variables

In real-world applications we may be faced with the problem of numerical underflow that occurs when the observation sequence is long, as α_t and β_t will become the products of a large number of terms, each less than 1. To avoid this, we compute

the *scaled versions* [Rabiner, 1989] of the forward/backward variables instead of the original joint probabilities.

The scaled forward variable is defined as: $\tilde{\alpha}_t(i) \triangleq \Pr(x_t^i \mid y_{1:t})$, which is a filtering distribution[6], and computed by introducing two extra variables: the partially scaled variable $\ddot{\alpha}_t(i) \triangleq \Pr(x_t^i, y_t \mid y_{1:t-1})$, and the scaling factor $\psi_t \triangleq \Pr(y_t \mid y_{1:t-1})$, so that $\tilde{\alpha}_t(i) = \ddot{\alpha}_t(i)/\psi_t$. Assume that at time t we have computed $\tilde{\alpha}_t(i)$, the recursion at time $t+1$ is then given as:

$$\begin{aligned}
\ddot{\alpha}_{t+1}(j) &\triangleq \Pr(x_{t+1}^j, y_{t+1} \mid y_{1:t}) \\
&= \sum_i \Pr(x_{t+1}^j, x_t^i, y_{t+1} \mid y_{1:t}) && (2.40) \\
&= \sum_i \Pr(y_{t+1} \mid x_{t+1}^j) \Pr(x_{t+1}^j \mid x_t^i) \Pr(x_t^i \mid y_{1:t}) && (2.41) \\
&= B_{y_{t+1}|j} \sum_i A_{ij} \tilde{\alpha}_t(i) && (2.42)
\end{aligned}$$

$$\psi_{t+1} \triangleq \Pr(y_{t+1} \mid y_{1:t}) = \sum_j \Pr(x_{t+1}^j, y_{t+1} \mid y_{1:t}) = \sum_j \ddot{\alpha}_{t+1}(j) \qquad (2.43)$$

Then

$$\tilde{\alpha}_{t+1}(j) = \frac{\ddot{\alpha}_{t+1}(j)}{\psi_{t+1}} \qquad (2.44)$$

Similar to the unscaled version the scaled forward variable also starts with an initialization at time $t = 1$:

$$\tilde{\alpha}_1(i) = \Pr\left(x_1^i \mid y_1\right) = \frac{\Pr(x_1^i, y_1)}{\Pr(y_1)} = \frac{\alpha_1(i)}{\sum_i \alpha_1(i)} = \frac{B_{y_1|i}\pi_i}{\sum_i B_{y_1|i}\pi_i}$$

For the scaled backward variable, we also introduce a scaled factor $\phi_t \triangleq \Pr\left(y_{t+1:T} \mid y_{1:t}\right)$, which can be computed as:

$$\phi_t \triangleq \Pr\left(y_{t+1:T} \mid y_{1:t}\right) = \Pr\left(y_{t+2:T} \mid y_{1:t+1}\right) \Pr\left(y_{t+1} \mid y_{1:t}\right) = \phi_{t+1}\psi_{t+1} \qquad (2.45)$$

The scaled backward variable then readily follows as:

$$\tilde{\beta}_t(i) \triangleq \frac{\beta_t(i)}{\phi_t} \qquad (2.46)$$

and the initialization at time $t = T$ is given by:

$$\tilde{\beta}_T(i) = \frac{\beta_T(i)}{\phi_T} = \frac{\Pr(\varnothing \mid x_T^i)}{\Pr(\varnothing \mid y_{1:T})} = \frac{1}{1} = 1$$

[6] i.e., the probability of a current state given the observation up to the current time.

Given the scaled forward and backward variables, the smoothing distributions can be easily computed. The (one-time slice) gamma distribution is derived as:

$$\gamma_t(i) = \Pr\left(x_t^i \mid y_{1:T}\right) = \frac{\Pr\left(x_t^i, y_{t+1:T} \mid y_{1:t}\right)}{\Pr\left(y_{t+1:T} \mid y_{1:t}\right)} \tag{2.47}$$

$$= \frac{\Pr\left(y_{t+1:T} \mid x_t^i\right) \Pr\left(x_t^i \mid y_{1:t}\right)}{\Pr\left(y_{t+1:T} \mid y_{1:t}\right)} \tag{2.48}$$

$$= \tilde{\beta}_t(i)\tilde{\alpha}_t(i) \tag{2.49}$$

Similarly, the two-time slice ξ_t distribution is computed as:

$$\xi_t(i,j) = \Pr\left(x_t^i, x_{t+1}^j \mid y_{1:T}\right) \tag{2.50}$$

$$= \frac{\Pr\left(x_t^i, x_{t+1}^j, y_{1:T}\right)}{\Pr\left(y_{1:T}\right)} \tag{2.51}$$

$$= \frac{\Pr\left(y_{t+1} \mid x_{t+1}^j\right) \Pr\left(y_{t+2:T} \mid x_{t+1}^j\right) \Pr\left(x_{t+1}^j \mid x_t^i\right) \Pr\left(x_t^i \mid y_{1:t}\right)}{\Pr\left(y_{t+1:T} \mid y_{1:t}\right)} \tag{2.52}$$

$$= \frac{B_{y_{t+1}|j}\beta_{t+1}(j)\, A_{ij}\tilde{\alpha}_t(i)}{\phi_t} \tag{2.53}$$

$$= \frac{B_{y_{t+1}|j}A_{ij}\tilde{\alpha}_t(i)\,\tilde{\beta}_{t+1}(j)}{\psi_{t+1}} \tag{2.54}$$

Finally, it is clear from the recursive formulae that the inference for the HMM has a complexity of $O(|Q|^2 T)$.

2.4.4 Parameter Estimation

Maximum-likelihood estimation $\theta^* = \text{argmax}_\theta \Pr(y_{1:T} \mid \theta)$ can be conducted using the Expectation-Maximization algorithm (section 2.2.2.2) along with expected sufficient statistics (ESS's) collected over time (section 2.3). The procedure is iterative between two steps: compute the ESS's from the smoothing distributions (E-step) followed by setting the newly re-estimated parameters to the normalized expected sufficient statistics (M-step).

Following the results in section 2.2.2.2, the expected complete log likelihood for the HMM can be written in exponential form, analogous to equation (2.29), as:

$$\left\langle \mathcal{L}^C(x_{1:T}, y_{1:T} \mid \theta) \right\rangle = \sum_{i \in Q} \left\langle T(\pi_i) \right\rangle \log \pi_i + \sum_{j \in Q} \sum_{i \in Q} \left\langle T\left(A_{ij}\right) \right\rangle \log A_{ij} + \sum_{i \in Q} \sum_{v \in V} \left\langle T\left(B_{v|i}\right) \right\rangle \log B_{v|i} \tag{2.55}$$

in which the three terms correspond to the three local conditional probabilities depicted in the three cliques in Fig. (2.5), and the ESS's $\langle T(.)\rangle$ are computed as follows:

$$\langle T(\pi_i)\rangle = \left\langle \delta_{x_1}^{(i)} \right\rangle = \sum_{x_{1:T}} \Pr(x_{1:T} \mid y_{1:T})\, \delta_{x_1}^{(i)} = \Pr\left(x_1^i \mid y_{1:T}\right) = \gamma_1(i) \tag{2.56}$$

$$\begin{aligned}\left\langle T\left(A_{ij}\right)\right\rangle &= \left\langle \sum_{t=2}^{T} \delta_{x_t}^{(i)} \delta_{x_{t+1}}^{(j)} \right\rangle = \sum_{x_{1:T}} \Pr(x_{1:T} \mid y_{1:T}) \sum_{t=2}^{T} \delta_{x_t}^{(i)} \delta_{x_{t+1}}^{(j)} \\ &= \sum_{t=2}^{T} \Pr\left(x_t^i, x_{t+1}^j \mid y_{1:T}\right) = \sum_{t=2}^{T} \xi_t(i,j)\end{aligned} \tag{2.57}$$

$$\begin{aligned}\left\langle T\left(B_{v|i}\right)\right\rangle &= \left\langle \sum_{t=1}^{T} \delta_{y_t}^{(v)} \delta_{x_t}^{(i)} \right\rangle = \sum_{x_{1:T}} \Pr(x_{1:T} \mid y_{1:T}) \sum_{t=1}^{T} \delta_{y_t}^{(v)} \delta_{x_t}^{(i)} \\ &= \sum_{t=1}^{T} \Pr\left(x_t^i \mid y_{1:T}\right) \delta_{y_t}^{(v)} = \sum_{t=1}^{T} \gamma_t(i)\, \delta_{y_t}^{(v)}\end{aligned} \tag{2.58}$$

Note that all the sufficient statistics in Eqs. (2.56), (2.57), and (2.58) have natural interpretations: $T(\pi_i) = \delta_{x_t}^{(i)}$ states the chance of having the hidden state x_1 starting with a value i; $T\left(A_{ij}\right) = \sum_{t=2}^{T} \delta_{x_t}^{(i)} \delta_{x_{t+1}}^{(j)}$ counts the number of times a state i making a transition to state j; and $T\left(B_{v|i}\right) = \sum_{t=1}^{T} \delta_{y_t}^{(v)} \delta_{x_t}^{(i)}$ shows how many times a state i generates an observation v.

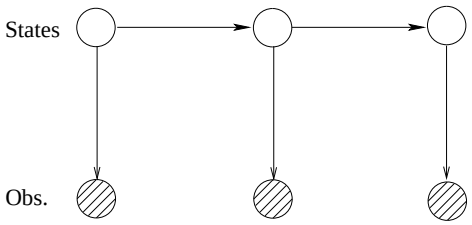
Using the results of section 2.2.1, the ML solutions are given by:

$$\hat{\pi}_i = \frac{\langle \pi_i \rangle}{\lambda} = \frac{\langle \pi_i \rangle}{\sum_{i \in Q} \langle \pi_i \rangle} = \frac{\gamma_1(i)}{\sum_{i \in Q} \gamma_1(i)} = \gamma_1(i) \tag{2.59}$$

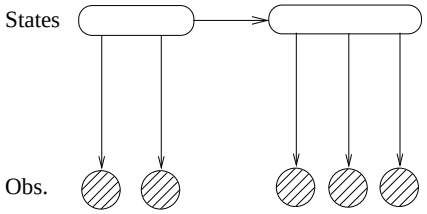
$$\hat{A}_{ij} = \frac{\langle A_{ij} \rangle}{\sum_{j \in Q} \langle A_{ij} \rangle} = \frac{\sum_{t=2}^{T} \xi_t(i,j)}{\sum_{t=2}^{T} \sum_{j \in Q} \xi_t(i,j)} \tag{2.60}$$

$$\hat{B}_{v|i} = \frac{\langle B_{v|i} \rangle}{\sum_{v \in V} \langle B_{v|i} \rangle} = \frac{\sum_{t=1}^{T} \gamma_t(i)\, \delta_{y_t}^{(v)}}{\sum_{t=1}^{T} \gamma_t(i)} \tag{2.61}$$

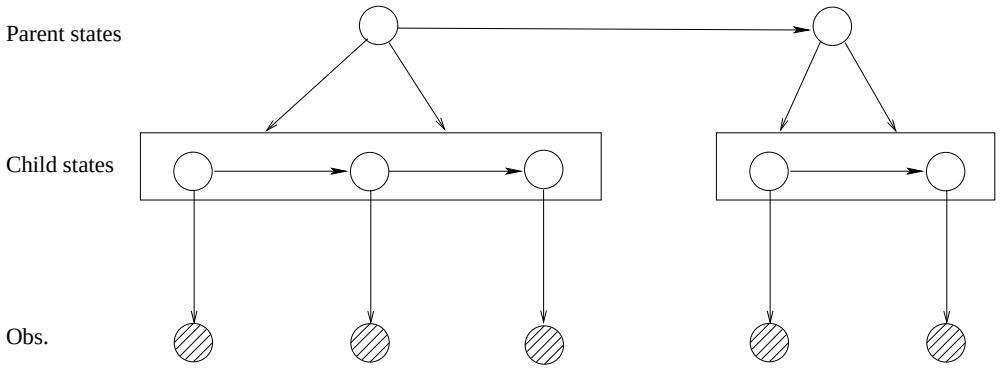
Finally, if there are M idd observations then the ESS becomes the sum of the ESS of every single observation (as a result of Eq. (2.20)).



(a) Hidden Markov Models.



(b) Hidden semi-Markov Models.



(c) Hierarchical Hidden Markov Models.

Figure 2.6: Block diagrams of the HMM, the HSMM and the HHMM.

2.4.5 Duration and Hierarchical Extensions

As identified in [Rabiner, 1989], the major limitation of the HMM is its modeling of state duration. As the state duration is defined by the state's self transition probability and thus follows a geometric distribution, the probability always decreases with longer durations. This is unsuitable for many natural sequences, therefore, extensions have been made into the conventional HMM by allowing the state durations to take general forms. This is done by setting the state's self transition probability to zero and specifying a separate distribution for its duration, such as a Multinomial or a Gaussian. In other words, a state remains unchanged for some duration of time[7] (Fig. (2.6)(c)) that is to be defined by its own duration distribution before transiting to a new state. The new model is called the Hidden Semi-Markov Model (HSMM) as it violates the strong Markov assumption. That means the next state depends not only on the current state but also on how long the chain has been in that state. By representing the HSMM in a DBN, inference and learning are done using the same techniques as in other DBN structures (e.g. HMM). Detailed descriptions of the HSMM are provided in chapter 3. It is, however, important to point out that the

[7] or equivalently emits a sequence of observations.

expressive power of incorporating duration modeling brings with its two important drawbacks: the increasing number of parameters associated with each state that need to be learned and the burden of additional computation. As detailed in chapter 3, while the first disadvantage can be solved using compact parameterization (e.g. the Gaussian or Gamma distributions instead of the Multinomial), the second one remains challenging. In particular, for all the state-of-the-art modeling choices of state durations (e.g. Multinomial and other distributions from the Exponential Family), state durations need to be explicitly counted at each time slice in the DBN representation. Hence, inference complexity in the HSMM is in order of the maximum duration and thus poses a serious issue in many applications.

Furthermore, by having a flat structure, the HMM is unable to model many physical signals with complex multi-scale hierarchical structures. Hence, another important extension to the HMM is the incorporation of hierarchical knowledge such as the hierarchical HMM (HHMM) [Fine et al., 1998] and the layered HMM [Oliver et al., 2002b, Sebe et al., 2005]. Fine et al. [Fine et al., 1998] were the first to introduce the HHMM by generalizing the HMM and viewing each state as an autonomous sub-HMM model. Fig. (2.6)(c) shows the block diagram for a two-layer HHMM. In the HHMM only the lowest layer *actually* generates observations, and hence is referred to as the production layer. States at layers other than the production layer "generate" a sequence of observations rather than a single observation by recursive activation of its sub-states. The state hierarchy proposed by [Fine et al., 1998] is, however, restricted to a tree structure, thus, it does not allow the sharing of lower-level states by higher-level states. This may be inappropriate for some applications. Bui et al. [Bui et al., 2004] introduce the concept of a general state hierarchy to allow the sharing of common substructures in the HHMM, hence providing more flexibility in the model. Inference and learning in the HHMM (with tree or general structure) can also be done in the usual DBN framework [Murphy, 2002, Phung, 2005b]. The layered HMM [Oliver et al., 2002b], on the other hand, does not strictly model hierarchy. It can be viewed as a cascade of HMMs where each layer is trained independently and the results of the lower layer are used as inputs to train the higher layer.

Finally, although hierarchy and durations are both important extensions on the HMM, there has not been any formal work to integrate these extensions into a

unified probabilistic model to exploit both the duration and hierarchical aspects often encountered naturally in many applications.

2.5 The Coxian and Phase-Type distributions

In an attempt to overcome the limitations of existing duration models in the HSMM, this thesis considers the Coxian distribution [Cox, 1955], a member of Phase-Type (PH) distributions as a promising candidate for modeling state durations. Thus, this section is devoted to summarizing current work on estimations and applications of the PH distributions, and the Coxian in particular. Detailed descriptions of the PH and Coxian distributions are presented in chapter 4.

As the PH distribution family is dense [Johnson and Taaffe, 1988], there are many existing works on fitting data into general PH distributions, and some on Coxian in particular. Most of these works [Johnson and Taaffe, 1990, 1991, Bobbio and Trivedi, 1990, Soren et al., 1996, Faddy, 1994, 1998, 2002] are continuous PH distributions; however, there is some recent work on discrete PH such as [Bobbio et al., 2003, Horvath and Telek, 2002, Isensee and Horton, 2005] . The continuous PH distribution is a generalization of the exponential distribution, therefore, it includes distributions such as exponential, Erlang, (continuous) Coxian, and hyper-exponential; whereas the discrete PH is a generalization of geometric distributions and examples are geometric, (discrete) Coxian, hyper-geometric and negative binomial distributions.

Early work on continuous PH distribution estimation was based on moment matching. Johnson and Taaffe develop methods for matching the first three moments [Johnson and Taaffe, 1988] or more generally the first k moments [Johnson and Taaffe, 1989] of non-degenerate distributions by a mixture of Erlang distributions of common order. The same group later extend their work to include a mixture of two Erlang distributions of not necessarily the same order, a Coxian distribution and also a PH distribution with no mass at zero [Johnson and Taaffe, 1990]. They also investigate the appropriateness of moment matching methods in queuing models [Johnson and Taaffe, 1991]. Bobbio and Trivedi [Bobbio and Trivedi, 1990] work on certain classes of acyclic PH distributions based on minimum distance fitting. However, more recent fitting work aims to maximize the data D likelihood: $\mathcal{L}_{(\boldsymbol{\mu},\boldsymbol{\lambda}) \mid D} = \Pr(D \mid \boldsymbol{\mu}, \boldsymbol{\lambda})$. Soren et al. [Soren et al., 1996], developed into a pack-

age called EMPHT-program [Haggstrom et al., 1992], view the PH distribution as a multi-parameter Exponential Family when the underlying Markov process is completely observed, and employ the EM algorithm [Dempster et al., 1977] to deal with incomplete observations. An interesting finding from their work is that except for the Erlang distribution with feedback, a Coxian is as good a fit as any general PH distribution with the same number of transient states. The Coxian distribution also finds its popularity in Faddy's work [Faddy, 1994, 1998], which uses a simple method (the Nelder-Mead method) to maximize the likelihood and compensates the algorithm non-convergence by a penalized function [Faddy, 2002]. Riska et al. [Riska et al., 2002] use a divide and conquer approach which involves two steps: partitioning data into sets and employing the EM algorithm to fit each data set to a PH distribution, then constructing a final fitting for the whole data from the resulting PH distributions. Most recently, Thummler and Telek [Thummler and Telek, 2006] claim a novel approach with more efficiency and numerical stability.

The discrete PH distribution has started to gain more attention recently. Bobbio et al. [Bobbio et al., 2003] are the first to introduce a discrete PH fitting method and restrict their work to the acyclic PH class. Their method employs an ML estimation procedure to compute the PH parameters in canonical form. They present a fitting package named PhFit [Horvath and Telek, 2002] to estimate both discrete and continuous PH distributions, which could become useful for heavy tail distributions as it separates the body and tail parts in the fitting process. However, the user is required to define the number of phases for both the body and tail fittings. Isensee and Horton [Isensee and Horton, 2005] implement three different optimization methods (Gradient Descent, Nelder-Mead Simplex, Simulated Annealing) for discrete PH approximation, and study the performance of the three algorithms and the effects of the PH distribution size.

Regarding applications, both the continuous and discrete PH distribution families (including the Coxian distribution) find their most common applications in queuing models in communication networks [Neuts, 1981, 1989, Ishay, 2002, Soong and Barria, 2000]. Recently, the continuous PH distributions are used to model state durations in Continuous Time Bayesian Network [Nodelman et al., 2005] and the discrete PHs are employed for the classic problem of sequence discrimination [Callut and Dupont, 2006]. With respect to the Coxian distribution, its continuous version

has lately become a useful practical tool in representing the survival of patients in hospitals [Faddy and McClean, 1999] and modeling patient duration of stay in hospitals [Faddy and McClean, 2000, Marshall and McClean, 2004], which also assists in estimating the cost for groups of aged patients [Marshall et al., 2007]. In [Marshall and McClean, 2004], based on the data collected on the patients, the continuous Coxian is fitted with different numbers of phases using a series of likelihood ratio testing to find the best fit model. Nevertheless, there has not been any significant applications reported for the discrete Coxian distribution.

2.6 Activity Recognition

In the literature, many different terminologies have been used by different authors to refer to human *intentional actions* such as *actions, motions, temporal textures, activities, behaviours, plans, primative/composite events,* e.g. [Polana, 1994, Brand et al., 1997, Bui, 2003, Tapia, 2003, Wang et al., 2003b, Wilson, 2005, Price, 2007]. In their survey, Moeslund and Granum [Moeslund and Granum, 2001] view definitions of human actions in current research under a hierarchy: action/motor primitives, actions, and activities. Action primitives are the smallest meaningful entities used to compose actions, which in turn are used to build activities. The authors gave *playing tennis* as an example of activity, while action primitives are *run left* and *back-hand*, and actions are a *sequence of primitive actions* performed to *return the ball successfully.* In the work of this thesis, the term *activity* (or sometimes high-level activity) carries the same meaning as in [Moeslund and Granum, 2001] since it refers to a series of intended human actions to complete a designated task, for example *"preparing dinner"*; while *atomic activity* is a primitive action or a sequence of primitive actions carried within an activity, for example "*cooking at stove*" .

More specifically we are working with *activities of daily living (ADLs)* [Katz et al., 1963], which are daily routines of house occupants in the independent-living home environment. An index of ADLs was introduced in [Katz et al., 1963] to measure both cognitive and physical functions of participants. Examples include bathing, exercising, cleaning and meal preparation. Many researchers [Lawton, 1990, Kempen et al., 1996, Greiner et al., 1996, Rogers et al., 1998] have continued to confirm the importance and effectiveness of ADLs on assessing the fitness of the elderly living independently in their home, inspiring research on automatic modeling and recogniz-

ing ADLs as well as detecting anomalies, e.g. [Mynatt et al., 2000, Tapia, 2003, Bao and Intille, 2004, Wilson, 2005, Rivera-Illingworth et al., 2005]. Readers are referred to [Haigh and Yanco, 2002] for a good survey on current research and technologies on automatic monitoring and assistance for the elderly to live independently in their home. Existing research tends to further categorize everyday activities into three classes: activities of daily living (ADLs) are activities related to personal care and essential for people living independently, e.g. eating, bathing or dressing; instrumental activities of daily living (IADLs) involve the use of instruments, e.g. preparing meals, washing dishes or dusting; and enhanced activities of daily living (EADLs) are the more challenging types of activities, e.g. using the Internet to communicate with others, pursuing further education or doing voluntary work. As we look at the activity domain from an activity modelling and recognition perspective, rather than a medical perspective, we use a loose definition on ADLs, which covers both basic ADLs and IADLs.

In particular, our motivating application is the construction of safe and smart houses for the aged that facilitate automatic monitoring and support of their occupants, aiming to increase the opportunities for aging in the family home. There are two main problems in building such a system. First, the system needs to learn, understand, and automatically build a model of the occupant's ADLs by observing what the occupant usually does during the day. Second, the system needs to be able to use its learned knowledge to monitor the person's current activity, i.e. to recognize what the occupant is doing and detect if there are any deviations from the normal activity patterns and alerting the carer if necessary.

2.6.1 Activity Recognition with Dynamic Stochastic Models

Dynamic stochastic models have been widely chosen for the problem of modeling and recognizing activities as they possess the capability of statistically describing how a state, modeled by a statistical model, can transit into another, making them particularly useful at capturing the dynamics in time-series activity data. This section presents a review of the use of dynamic stochastic models in the field of activity recognition, particularly focusing on the Hidden Markov Models and their variants as they are more closely relevant to our work.

2.6.1.1 Approaches using the HMMs and their variants

The HMMs and their variants are attractive approaches for learning and recognizing activities as they possess the following appealing properties. First, they can be represented by DBNs, which have clear Bayesian semantics and model conditional in/dependencies in a natural way, and thus are effective in handling time-varying and/or incomplete data. Second, simple and efficient inference and learning algorithms are available [Rabiner, 1989, Murphy, 2002]. Third, prior knowledge, as well as new domain knowledge can be incorporated, especially as it is more intuitive with the availability of graphical representation. The HMMs and their variants, however, are not generally applied directly to noisy raw video/sensor data. Essential features are first extracted from data via various means, e.g. Bayesian Networks to map video data into event concepts [Hongeng and Nevatia, 2003], particle flow to extract motion information [Niu and Abdel-Mottaleb, 2004], principle component analysis coefficients to represent activity trajectories [Bashir et al., 2007], or more generally an independent data conversion scheme accompanied by observation models [Nguyen, 2004b]. The HMMs and their extensions then operate on these features to exploit their temporal properties. This section presents the literature review on approaches to Activity Recognition using the HMMs and their variants[8] with a focus on duration and hierarchical extensions.

HMM-based Approaches

Initially the HMMs were mainly used in speech recognition, and its usefulness in understanding time-sequential data was first brought to the attention of the visual community in the early 90s by Yamato et al. [Yamato et al., 1992]. Yamato et al. opened new potential applications in activity recognition for the HMMs and their variants, successfully applying the HMMs for learning and recognizing different strokes in tennis games. The HMMs became popular with activity recognition work in the 90s, such as recognizing American Sign Language [Starner and Pentland, 1995], simple tasks: pick-up, put-down, push, pull, drop, and throw [Siskind and Morris, 1996], human gait [Bregler, 1997], and gestures [Lee and Kim, 1999]. For instance, Lee and Kim [Lee and Kim, 1999] argue that a simple threshold is often not sufficient for recognizing hand gestures and thus make use of the internal

[8] Readers may be also interested in [Cappe, 2001], which provided a good list of papers developed between 1989 and 2000 on general study and extensions on the HMMs and their applications in various other areas.

segmentation property of the HMM, saying that each state of a trained HMM represents a sub-pattern and state transitions follow sequential order of sub-patterns, to construct an HMM-based threshold model. The HMM-based threshold model is a simplified version of an ergodic HMM that copies states from all trained HMMs and fully connects them. Only gestures whose best likelihoods given all gesture models are higher than their likelihoods given the threshold model are accepted as valid ones.

HMMs continued their popularity in this decade and have been exploited in different perspectives by various researchers [Brand and Kettnaker, 2000, Hamid et al., 2003, Niu and Abdel-Mottaleb, 2004, Wyatt et al., 2005, Lester et al., 2006, Bashir et al., 2007]. For example, Brand and Kettnaker [Brand and Kettnaker, 2000] argue for the use of entropy minimization over the conventional Baum-Welch formula during learning to better reveal the hidden structures in the data, as minimization is not only performed on the entropy of the data's expected sufficient statistics (ESS's) but also on the cross-entropy between the ESS's of the data and the model, and entropy of the model itself. The authors apply the HMM learned with this modified EM algorithm for the application of office activities and monitoring traffic. In addition, as the latter application involves a varying number of objects, the authors use a mixture observation model to handle the variable-length observations. Addressing multiple observation problems, Xiang and Gong [Xiang and Gong, 2007] simply fix the number of observation symbols emitted at a given time. The observation space is then factorized by assuming each observation symbol is independent of the others. The authors use different Multi-Observation HMMs (MOHMMs) to model different behaviour patterns and then build a composite behaviour model using the mixture of learned MOHMMs. Their target applications are in surveillance scenarios including both indoor, e.g. entering/existing building, and outdoor, e.g. aircraft docking procedure. Peursum et al. [Peursum et al., 2004] use the (flat) left-right HMMs to segment relatively complex hierarchical activities, e.g. printing and retrieving documents or making tea, by viewing the segmentation problem as missing observations. Labels of sub-activities are supplied during training as observation features and treated as missing observations during testing. Nodes in the HMM's DBN representation associated with missing observations are omitted during computation of forward and backward variables, and the most probable sub-activity labels y_t (dubbed missing observations) are inferred from the distribution $\Pr(y_t \mid \text{Observation}_{1:T})$. This approach, however, does not allow online segmenta-

tion. As the author has pointed out, more flexible structures are required for activities with less restricted temporal order and a hierarchical HMM would be better at modeling complex hierarchical activities. Aiming for view-invariant recognition, Niu and Abdel-Mottaleb [Niu and Abdel-Mottaleb, 2004] use a bank of HMMs, each trained for one activity captured from different camera views. It is shown that the recognition can benefit when both optical flow motion information and eigen-based shape features are used (88.3% accuracy). However, the set of activities considered are still relatively simple and primitive. Bashir et al. [Bashir et al., 2007] segment object motion trajectories at changing points in their curvatures and represent sub-trajectories by their principle component analysis (PCA) coefficients. Each state in the HMM is then modeled by a mixture of Gaussians representing the PCA coefficients of sub-trajectories. It was found that this HMM-based trajectory modeling delivered more consistent performance than Gaussian Mixture Models, when recognizing activities with temporal orders, such as in Australian Sign Language and sport activities. However, most of these works are based on the assumption that activities have flat structures with simple temporal orders, and thus, are inadequate when it comes to learning, classifying and segmenting composite activities with complex structures and temporal signatures.

While using HMMs is suitable and efficient for learning simple sequential data, its performance seriously degrades when the range of activities becomes more complex, or the activities exhibit long-term temporal dependencies that are difficult to deal with under the strong Markov assumption. Further, activities that are hierarchical in nature cannot be adequately modeled by the flat HMM structure. As mentioned in section 2.4.5, to overcome these limitations two popular classes of extensions to the HMM have been proposed: the HSMM and HHMM.

HSMM-based Approaches

The HSMM has the advantage of being able to model non-exponential/non-geometric state durations and has been reported to achieve higher recognition rates in several papers, e.g. [Hongeng and Nevatia, 2003, Luhr et al., 2004, Tweed et al., 2005, Natarajan and Nevatia, 2007a]. Common choices of distributions used for modeling state durations include Multinomial [Luhr et al., 2004, Marhasev et al., 2006, Natarajan and Nevatia, 2007a] and Gaussian [Hongeng and Nevatia, 2003]. This advantage, however, carries a heavy computational burden in both training and

classification because the inference complexity depends on maximum duration span of a state (as briefly explained in section 2.4.5); that is, $O\left(|Q|^2 MT\right)$ where $|Q|$ is the number of states, M is the maximum duration and T is the activity length. In activity recognition the maximum sub-activity/activity duration can be arbitrarily large. Realizing this computational drawback, several attempts have been made to improve the inference complexity. Hongeng and Nevatia [Hongeng and Nevatia, 2003] try to limit the duration span to a small value during training and classification, which may work for some typical durations: uniform by taking the maximum upper bound, or normal by considering only the region around the mean. The authors, however, apply their models to activities from various domains, such as ground and airborne surveillance, whose temporal variances are not guaranteed to follow uniform or Gaussian distributions. Alternatively, narrowing their purpose to decoding only, Tweed et al. [Tweed et al., 2005] impose concave monotonic conditions on the duration distribution of the HSMM to achieve an $O\left(|Q|^2 T\right)$ most-likely sequence inference algorithm, and apply it for recognizing behaviours from a British breakfast television program.

Computation costs become more severe as more structure is incorporated into the HSMM. An example is the Non-stationary Hidden semi-Markov Models (NHSMM) used in [Marhasev et al., 2006] to learn and recognize normal and abnormal human behaviours in large-scale surveillance, e.g. activities of passengers in airport. The NHSMM extends the HSMM by allowing transition probabilities between states to depend on state durations. That means the probability of going from state i to state j depends on the duration an agent has spent in state i. The authors report important improvements in recognizing both normal and abnormal activities as compared with the HSMM and HMM. As expected, the drawback of this model is its computational cost since the likelihood computation complexity is $O\left(|Q|^2 M^2 T\right)$ in comparison with $O\left(|Q|^2 MT\right)$ of the HSMM, or $O\left(|Q|^2 T\right)$ of the HMM. Another example is the Coupled Hidden semi-Markov Models (CHSMM) proposed in [Natarajan and Nevatia, 2007a] to model activities involving multiple agents. The Coupled HMM [Brand, 1996] is an extension of the HMM to model multiple dependent processes by running multiple Markov chains in parallel, connecting their states across time slices. The CHSMM extends the CHMM in the way a HSMM expands the HMM, i.e. allowing state duration to be non-exponential. By incorporating both duration and multiple channels the CHSMM brings with it a very heavy com-

putation cost. The authors then adopt Brand's approximate learning algorithm for the CHMM [Brand, 1996] with some important modification to achieve an inference complexity $O\left(C^2 |Q|^2 M^3 T\right)$, which is still expensive. In testing their models the authors had to assume uniformly distributed duration models when building their visual synthetic surveillance and normal duration distribution models for learning American Sign Language. Again, it is clear from our review of the HSMM that there is a strong need for a new type of duration parameterization to solve the computational setback of current duration models.

HHMM-based approaches

The HHMM was first applied to the problem of learning multi-level structure in text and detecting stroke patterns in handwriting in [Fine et al., 1998]. It is also used for video analysis in [Xie et al., 2002], and later extended in [Xie and Chang, 2003]; however, hierarchy is not effectively modeled as the HHMM is "collapsed" into a flat HMM during inference and learning. Luhr et al. [Luhr et al., 2003] are the first to employ the HHMM in modeling and recognizing human activities. Nevertheless, in these models the state hierarchy in the HHMM is restricted to a tree structure. This is not suitable for complex activities as they sometimes share common atomic activities, for example, both "cooking dinner" and "making coffee" may involve the use of kitchen cabinets. This problem in solved in [Bui et al., 2004] wherein state hierarchy can be an arbitrary lattice structure. This model is first used to learn activity trajectories using simulated data in [Bui et al., 2004] and then real surveillance data in [Nguyen et al., 2005, Nguyen and Venkatesh, 2005]. Nguyen et al. [Nguyen et al., 2006] later use a set of multiple HHMMs integrated with the joint probabilistic data association filters (JPDAFs) to track and recognize high-level behaviours of multiple people in a home environment. Each HHMM is associated with a single person with high-level states for complex behaviours and low-level states for atomic behaviours, and the assignment of people to observation data is handled by the JPDAFs. Another type of hierarchical extension is the layered HMMs used in [Oliver et al., 2002b, Sebe et al., 2005] for activity recognition. At each layer a bank of HMMs (dubbed discriminative HMMs) are run in parallel and the model with the highest likelihood is selected at each time slice. Outputs from the lower layer are used as input to the higher layer. The layered HMMs are found to be useful in enhancing the robustness in activity recognition by reducing training and tuning requirements via re-training the lowest layer, and keeping the higher-level

layers unchanged. The LHMM decomposes activities into different time granularities by using a sliding time window at each layer. An observation is generated for each processed time window and passed as input to the next layer. The authors [Sebe et al., 2005] use their intuition and knowledge on the activity types being modeled at each layer to decide on sliding window sizes. Finally, even though hierarchical structures have been efficiently modeled in these works, temporal variances at both atomic and complex activities are overlooked in the HHMMs, and not sufficiently modeled in the LHMMs.

Approaches using other variants of the HMM

Apart from the two important extensions above, the HMMs have been extended in several other ways tailored to the need of different applications. A common extension is to supply the HMMs with parallel running Markov chains [Vogler and Metaxas, 1999, Brand et al., 1997] to accommodate multiple interacting agents. Model variants and their applications in human activity recognition include: (i.) the parallel HMMs for American Sign Language [Vogler and Metaxas, 1999] and later extended to human gait [Vogler et al., 2000], (ii.) the coupled HMMs for Tai Chi martial art [Brand et al., 1997] and the occurrences of human interactions and types of interactions [Oliver et al., 2000], (iii.) the factorial HMMs for human gaits [Chen et al., 2007], and (iv.) the dynamically multi-linked HMMs for group activities at outdoor scenes [Gong and Xiang, 2003]. In multi-channel HMMs, inference complexity is exponential in the number of Markov channels C because of the composite state $x_t \triangleq \left\{x_t^{(1)}, \ldots, x_t^{(C)}\right\}$ and observation $y_t \triangleq \left\{y_t^{(1)}, \ldots, y_t^{(C)}\right\}$. The typical approach to reduce the model complexity is to make assumptions on the interaction between channels as well as between channels and their observations, so that the composite transition probabilities and emission probabilities can be factorized. For example, in the coupled HMMs (CHMMs) [Brand, 1996] states from parallel HMMs are connected over one time slice and the composite state transition probability is factorized into products of individual $\Pr\left(x_{t+1}^{(c)} \mid x_t^{(c')}\right)$ with $c,\ c' \in [1, \ldots, C]$. This could become erroneous as C increases since the composite transition probability is set to the product of a large number of terms, each less than 1. The author also assumes each chain has its own observation sequence to factorize the observation probability as a product of individual $\Pr\left(y_t^{(c)} \mid x_t^{(c)}\right)$. Consequently, a deterministic $O\left(C^2 |Q|^2 T\right)$ approximation for MAP state estimation for C fully coupled chains of $|Q|$ states is derived, making the CHMM computationally feasible. A CHMM with

two chains is then later used [Brand et al., 1997] to perform visual classification of Tai Chi Chuan two-hand gestures. Gong and Xiang [Gong and Xiang, 2003] simplify the state factorization problem in the CHMMs by disconnecting irrelevant causal relations between state variables across processes when constructing their dynamically multi-linked HMMs (DML-HMMs). In order to achieve this simplification the authors have to first learn the topology of their DML-HMMs using Schwarz's Bayesian Information Criterion. Their application is to model and recognize airport cargo loading and unloading activities.

There are applications where multiple observations either are not or cannot be assumed to come from multiple interacting processes. For example, observations from various sources such as videos and sensors describing ADLs of a single house occupant. In these cases the probabilities of observing multiple symbols given a hidden state at a time in the HMMs can be factorized [Brand and Kettnaker, 2000, Xiang and Gong, 2007] or constructed using a combinatorial method [Li et al., 2000]. Also related to observation models, Wilson and Bobick [1999] extend the HMMs by introducing a global parameter into the output probabilities, forming the parametric HMMs. The authors then apply the parametric HMMs for gesture recognition.

Another extension is to enrich the HMM with context [Bui et al., 2002], transforming it from a type of probabilistic context free grammar to context dependent grammar. The Abstract HMM (AHMM) proposed in [Bui et al., 2002] is a multi-scaled probabilistic model, consisting of multi-layer abstract policies where a policy is similar to a high-level state in the HHMM. The policy selection process follows a top-down decomposition. The higher policy selects the lower one and the execution continues to the bottom level, where the bottom level policy does not select another policy but models a Markov chain. The observations are then generated directly from this Markov chain. Unlike the HHMM, the AHMM allows the refinement of an abstract state into lower-level states to be dependent on the current context, modeled by the current state at the bottom level, and thus it belongs to the class of context-dependent models. The AHMM is first applied to activity tracking and recognition [Nguyen et al., 2004], then used to model movements in indoor [Osentoski et al., 2004] and outdoor [Liao et al., 2004] environments. Bui [Bui, 2003] later introduces memory into the AHMM to form the Abstract Hidden Markov Memory Model (AHMEM) that allows the choice of the next sub-plans to depend not only on

the current state, but also the sub-plans chosen in the past. The AHMEM is found to be effective at handling noisy data from multiple sources with its applications in learning behaviour model from multiple cameras [Nguyen et al., 2004]. It is also later extended to handle multi-agent policies, called the MultiAgent AHMEM, for recognizing simple actions like walking and jumping, in which different body parts are represented as cooperative agents [Kosta et al., 2006].

The HMMs have been extended in various ways as discussed above, however, as our current interest is to build stochastic models that are capable of automatically learning and recognizing complex ADLs of a single house occupant as well as detecting any anomalies, we focus only on extensions of HMMs which help towards this goal. To this end we recognize that the best models for ADLs are those competent to exploit both temporal variations and hierarchical decompositions (with shared substructures) in a computationally effective fashion. Previous work [Kautz et al., 2003] has recognized the need to combine both the hierarchical and semi-Markov extensions to form the Hierarchical Hidden Semi-Markov Model. However, there has been no attempt at formalizing such a model or in demonstrating its usefulness empirically over other existing models.

2.6.1.2 Approaches using other dynamic stochastic models

Apart from the HMMs and their variants, activity recognition involves the use of other probabilistic finite state automatons and many different structures of Dynamic Bayesian Networks. This section provides a brief review on these approaches, focusing on their durational and hierarchical modeling aspects.

The Variable Length Markov Model (VLMM) is a probabilistic finite state automaton capable of capturing processes with variable memory lengths (as opposed to the fixed memory Markov model) wherein states are not hidden. Galata et al. [Galata et al., 2001] use the VLMMs to learn, recognize, animate and predict exercise routines. By having variable memory length, the VLMM is effective at capturing large scale temporal dependencies and is particularly good for behaviours with both short- and long-term temporal dependencies. However, it does not model state durations. Medioni et al. [Medioni et al., 2001] use a finite-state automaton to recognize hierarchically complex activities (dubbed multi-state scenarios such as “a car is avoiding

the checkpoint"), where each state represent a sub-activity. Rule-based methods are used to compute the likelihood of occurrences of activities. The authors later extend their work in [Hongeng et al., 2004] enabling activity likelihood to be calculated rigorously via Bayesian and first-order logic.

A substantial amount of work on activity recognition has been based on DBNs; nevertheless, they generally do not explore both durational and hierarchical properties embedded in human actions. First, we examine some research efforts that incorporates duration information. For example, the DBN used in [Du et al., 2006] to recognize a few different types of interacting activities between two people from video data, can be viewed as a simplified version of a coupled HSMM with two channels: one consists of global activity states with learned uniform duration distributions, while the other includes local activity states without durations. A simple one-way causal relationship directs global states to local states. The global states generate global observation features containing velocities of people, distances between them, and angles between moving directions. The local states generate local features comprising of the aspect ratio of the tracked bounding box and inclination of torso. Another example is the Activity Graph proposed in [Patterson et al., 2004] to present ADLs in terms of the gross manipulation of household objects supplied by RFID tags. An Activity Graph consists of a set of disjunctive and conjunctive arcs (allowing partially ordered activities) with probabilities on disjunctive arcs and nodes with Gaussian durations. An Activity Graph has to be constructed for each ADL, and can be represented by a DBN and inference done with Rao-Blackwellized particle filters [Murphy and Russell, 2001]. Durations of sub-activities (nodes in the Activity Graph) are modeled by Gaussians but parameter learning is not supported. The authors also intend to extend their work to model hierarchically complex activities as well as address interrupted and resumed activities. Shi et al. [Shi et al., 2004] take a different approach by looking at (finite) temporal intervals comprising activities instead. Component temporal intervals in activities are allowed to be partially ordered or in parallel. The authors propose a Propagation Network (P-Nets), which is a form of DBN with the ability to model duration explicitly. Each node in P-Nets is associated with a temporal interval and its state (i.e. active or inactive) once initiated, depends solely on its Gaussian duration model, which is to be learned during training. The authors apply P-Nets to model the task of calibrating a blood glucose monitor, commonly used by the elderly with diabetes. They later extend their

work to a new scheme called P-Nets+Boosting (by introducing a boosting-based learning method) [Shi et al., 2006] aimed at reducing the burden of manual work in constructing the network. In addition, they broaden their experiment to include two more data sets: indoor activities (e.g. making a phone call, reading a book) and weight-lifting exercises. However, common choices of distributions for modeling activity durations are still multinomial and Gaussian, therefore, they suffer from the same disadvantages as in the HMM-based approaches discussed previously.

DBNs have been effectively exploited in revealing the natural hierarchical organization of human actions in a number of research works. For instance, Kitani et al. [Kitani et al., 2005] address the hierarchical nature in human activities via the innovative combination of stochastic context free grammar (SCFG) and Bayesian networks. SCFG is used for its expressive power. Hence, descriptions of hierarchical structures in activities is first given in a SCFG, which is then used to generate a hierarchical BN, as the BN is particularly good at handling the uncertainty of human actions and allows complex probabilistic queries across the grammar. Deleted interpolation, a smoothing technique in natural language processing, is carried out on the DBN and used to recognize various activities including temporally overlapped ones such as passing through the scene and departing the scene; nevertheless, the problem of duration modeling is not discussed. Hoey [Hoey, 2001] use a hierarchical Bayesian Network in which the lowest level is a mixture of Gaussian distributions, whereas upper levels are a mixture of Markov chains for learning and recognizing facial expression events in video. A DBN representation is obtained by unrolling the hierarchical BN at different time scales at each level, with the higher level operating at lower time scales. Although the hierarchy is nicely modeled and learned, event durations are not incorporated, furthermore, manual segmentation of video streams is required, making it unsuitable for real-time applications. Oliver and Horvitz [Oliver and Horvitz, 2005] extend their Layered HMMs [Oliver et al., 2002a] by combining the use of both HMMs and DBNs. This is done by replacing the bank of discriminative HMMs at the highest layer by a DBN with hidden "activity" states while keeping the lower-layer HMMs. Office activity labels inferred from this DBN-HMMs structure give better accuracy rates than the Layered HMMs, nevertheless, as the authors point out, it is more computationally expensive. Again, activity hierarchy is well modeled but their temporal characteristics are overlooked.

To sum up, these approaches have the same problems with those based on the HMMs. They generally either do not model duration characteristics of activities in a computationally efficient fashion or integrate both duration and hierarchical properties.

2.6.2 Non-dynamic Approaches to Activity Recognition

There are few research works based on relatively simple techniques such as template matching, nearest neighbor and decision tree classifiers, to recognize simple human actions. The approaches generally cannot incorporate hierarchical/temporal information or support online recognition. For example, Collins et al. [Collins et al., 2002] present a view-dependent method for human identification based on template matching of 2D silhouettes extracted from the gait sequences. Rao et al. [Rao and Shah, 2001, Rao et al., 2002], on the other hand, propose a view-invariant representation of actions composed of atomic units called dynamic instants and intervals. Instants are instantaneous entities showing motion changes: speed, direction, acceleration, and curvature, while intervals are time periods between two instants during which no important motion changes occurs. Distances between activities' instants are then computed and used to decide a match, and activities used are relatively simple like opening a cabinet, picking up an object or erasing the board. Alternatively, Ben-Arie et al. [Ben-Arie et al., 2002] recognize simple activities such as walking or jumping using multidimensional indexing of body poses. [Bao and Intille, 2004] apply C4.5 decision tree, nearest neighbour, decision table and naive Bayes classifiers to recognize relatively complicated activities such as riding escalators or vacuuming, using features extracted from acceleration data. The acceleration data is obtained from several accelerometers worn on different body parts of the participants.

Graphical representations are common in handling more complex activities, especially those with hierarchical orders. Hongeng et al. [Hongeng et al., 2000] use a Bayesian Network comprising of several naive Bayesian classifiers to represent hierarchically complex activities at parking bay and checkpoint monitoring, with a bottom-up inference process. This representation of hierarchy is very intuitive, however, the network expands expensively in accommodating different scenarios. Also, temporal dependencies cannot be incorporated. Minnen et al. [Minnen et al., 2003] propose a system based on stochastic parsing to automatically annotate well-

ordered activities such as the Towers of Hanoi task from video data. Ghanem et al. [Ghanem et al., 2004] use Petri Nets with extensions to represent composite (hierarchical) events (e.g. car activities in car park) from simple events by combining their logical, temporal and spatial relations. Liao et al. [Liao et al., 2005] use relational Markov networks for location-based activity recognition. While duration information can be incorporated as feature functions, activity duration distributions cannot be modelled and learned explicitly. Avrahami-Zilberbrand et al. [Avrahami-Zilberbrand et al., 2005] face similar problems with their plan library representation, in which durations of plan-steps cannot be modeled explicitly but are defined by a set of temporal constraints.

2.7 Detection of Anomalies in Activity

Anomalies are deviations from the common forms and are referred to with different names such as abnormality, outlier, irregularity, deviation, rarity, unusualness, etc. Due to its importance in various areas, e.g. assistive technologies for the elderly and patients, and security surveillance in airports and buildings, anomaly detection has been extensively researched. However, we only discuss the most relevant works. Our focus is to deliver an anomaly detection framework in a smart home context for the aged, which is *firstly* able to detect a subtle (thus, harder to spot) form of abnormality, namely *duration abnormality*, which are deviations in durations (longer or shorter) spent at some locations in the activity sequence, or equivalently the unusualness in the pace of conducting activity. This kind of anomaly, if detected, can provide important clues in alert systems. For instance, a person staying at a location for a longer duration than usual might indicate the onset of illness or disability. *Secondly*, the detection scheme needs to be able to recognize an anomaly as soon as it appears, and it is also important to detect when anomalous activities return to normal. This ensures alerts to be raised on time and false alarms to be minimized. *Thirdly*, the system needs to be capable of dealing with practical issues including insufficient negative training data (as abnormalities are rare and varied) and the requirement of a defined abnormality threshold for detection decision (since it is sometimes difficult to specify in practice). The remainder of this section is devoted to discussing anomaly detection work along these three criteria.

Most research on anomaly detection is to detect abnormalities in “apparent” forms,

e.g. deviations in terms of activity trajectories or sequential orders; however, some of them are also able to detect duration anomalies. Examples in large scale environments include [Grimson et al., 1998, Chellappa et al., 2003, Vaswani et al., 2004]. Using visual data returned from a distributed set of cameras, Grimson et al. [Grimson et al., 1998] detect unusual speeds of moving vehicles in a surveillance area as outliers of the clusters representing normal activity patterns. Chellappa et al. [Chellappa et al., 2003] use statistical shape theory to model the changing configurations of interacting objects and examine their mean and dynamic deviations to spot abnormal behaviour in the tracked objects. In particular, abnormalities in the walking pace of a passenger (e.g. sudden stop in his track) in an airport is detected via the changes in shape formed by all passengers. The importance of detecting duration anomalies for indoor activities has also been recognized. For instance, Rivera-Illingworth et al. [Rivera-Illingworth et al., 2005] use an Adaptive Neural Architecture, which, differing from most neutral networks, can grow to accommodate new samples in data without the need for re-training the whole network. The network consists of an input layer, a hidden layer (with or without an accompanied memory layer) and an output layer. It detects unusualness by using a threshold in the hidden layer or growing a new class node to interpret the unfamiliar sample. The importance of detecting deviations in the sequence, frequency and also durations of learned activities is recognized by the author as they have introduced a memory layer. Nevertheless, evaluation on the effectiveness of this memory layer is left for further work as their current experiment only tests activities that are totally new to the network. Hara et al. [Hara et al., 2002] first cluster sensory data obtained from ubiquitous small motion detectors installed in a smart house into a manageable number of states using the nearest neighbour method. They use first-order Markov chains to construct templates for daily activities (e.g. one template for every hour) defined on the state transition probabilities and state transition duration time distributions. Anomalous patterns are detected by comparing either the average log likelihoods of the transition probabilities and transition duration time computed from their cluster sequence, or their Kullback-Leibler/Euclidian distances from the activity templates against some thresholds. This framework can detect unusual duration patterns, for instance, the house occupant faints (stops long at a place) while walking, but detection may fail if this happens in areas such as a living room where stationary patterns are common. More closely related to our work is that of [Luhr et al., 2004], who have done some primitive work on detecting duration abnormal-

ities in daily activities. Activities whose decomposed sub-activities have unusual durations are detected as an abrupt decline in the likelihood function given their trained normal models. Good performance is reported when a left-to-right explicit duration HSMM is used. Nevertheless, apart from modeling state durations expensively by using multinomial distributions, this approach is unable to recognize the return of abnormal activities to normal. Moreover, it requires decisions on threshold values, which is generally hard to define in practice.

Currently there is not much activity recognition work that supports detection of abnormal state returning to normal. Hu et al. [Hu et al., 2006] offer a system in a large scale surveillance scenario which can achieve this goal. Based on the learned (normal) statistical motion patterns, the system uses statistical methods to detect anomalies in real traffic scenes as soon as they appear and labels the tracked object with an abnormality probability. Thus, it is able to recognize when the object behavior comes back to normal. Regarding anomaly detection in home domains, Yin et al. [Yin et al., 2007] also recognize the importance of being able to realize when unusual activities have returned to normal but have yet implemented this function in their framework. Hence, they still face with the risk of generating a large number of unnecessary abnormal models as their framework uses the learned normal activity models to derive one new model for every new type of unusual activity.

In dealing with the scarcity of abnormal data available for training, there are two common approaches: the *unsupervised* approach makes use of the scarcity property of unusual activities to filter out activities with the most discrepancies in the data set, while the *model-based* approach first builds normal models from training data, then uses them to infer abnormal models or to set thresholds. Less common is to *manually* construct anomalous models. For example, Zhong et al. [Zhong et al., 2004] take a completely unsupervised approach in detecting anomalies. They do not model normal activities but view them as patterns that are repeated over time and develop a similarity-based framework to detect unusual activities. Applications include monitoring patients in a hospital dining room, detecting cheats in poker games, detecting unusual car and pedestrian activities in road surveillance, and analyzing the crowd via web cam. However, this approach is unsuitable for online detection and at the same time requires a large data set for sufficient differentiation.

With respect to the model-based approach, Zhang et al. [Zhang et al., 2005] has empirically shown that a semi-supervised framework is superior than both unsupervised and supervised ones, provided that only a small amount of abnormal data is available for training in the supervised case. The authors detect unusual events (cheating) from a poker video game, e.g. "passing cards under tables" and "hiding a card" using a semi-supervised adapted HMM framework, in which a usual event model is first learned from training data, and then use it to produce a number of unusual event models (one at each iteration) by using Bayesian adaptation techniques in an unsupervised fashion. This adapted framework is shown to yield better detection rates than both an unsupervised HMM-based clustering approach and a supervised HMM approach when given little abnormal training data. Even though their work shows promising results, the optimal number of iterations has to be found experimentally, and the task of optimizing the number of iterations has not yet been investigated. The biggest disadvantage of this approach is probably the excessive growth in the network as a new abnormal event model is added at every iteration. Some of the work in the model-based approach does not derive abnormal models from normal ones, but instead sets a threshold to classify anomalies. For instance, in their building surveillance task, Hamid et al. [Hamid et al., 2005] learn normal activity models (represent activities as bags of n-grams) and identify abnormal activities based on discrepancies. For the application of unusual event detection in crowds (large groups of people), Andrade et al. [Andrade et al., 2005] train a number of HMMs to model normal events using features extracted from particle flow patterns of scenes, with decisions on classifying normal or abnormal events based on comparing the likelihoods of the observed events obtained from the bank of normal HMMs against a threshold. Xiang and Gong [Xiang and Gong, 2007] build a composite normal behaviour model using a mixture of MOHMMs from training data from surveillance video, and compute an online anomaly measure that is a weighted sum of the normalized log likelihood of the unseen behaviour given the composite model. Liao et al. [Liao et al., 2004] do not derive abnormal models from normal ones but use a prior model for anomalous behaviours. The authors detect anomalies in a user's daily transportation routines by comparing the log likelihoods of two models: a hierarchical Markov model learned from (normal) training data and a flat model parameterized by prior knowledge of general physical constraints and not subject to the user's daily routines. This framework has successfully identified the occurrence of anomalies when the user had missed her typical bus stop. Another

example of using prior models is in [Chan et al., 2004]. To deal with the exceptional rarity of unusual events in aerial data, Chan et al. [Chan et al., 2004] show that semantic observations are more effective in generalizing unseen scenarios than direct continuous observations. They use HMMs to model the spatial and temporal relations between interest objects with observations based on their binarized distances. However, the rare (unusual) activities models are manually constructed, making this approach only applicable to expected, simple and well defined activities.

2.8 Video Segmentation and Annotation

Analogous to video surveillance where we are able to segment a raw surveillance video into coherent units of activities (e.g., making breakfast, cooking dinners), this thesis also explores to another application area that is to segment and possibly annotate professionally made videos into coherent units of topics (e.g., video segment about safety rules in office) using the model developed. Different from raw videos captured from fixed cameras, professionally made videos are more sophisticated, often intensively edited, to create entertaining experiences or to convey certain messages to the viewers. Duration as well as hierarchical information can be exploited and modeled to fulfill some common tasks such as segmentation and annotation. We provide a brief review on video segmentation and annotation, focusing on approaches related to the models developed in this thesis.

The goal of the video segmentation is to characterize the temporal dynamics of the video whereby it can be segmented into coherent units, possibly at different levels of hierarchical abstraction. Slightly different from the computer vision community, video segmentation in the multimedia field is generally concerned with edited videos such as broadcast news, motion pictures, educational videos, documentary films and so on. As they are professionally made, the structure and content usually adhere to a predefined story board and are thus often rich in content. Video content-based indexing and segmentation has been one of the central problems for decades in multimedia computing and is still an active problem. Starting from individual image frames, the next coherent segmental unit of a video is a *shot* which is defined as the video segment 'resulting from a single run of the camera [Phung, 2005b] (e.g., the portion in news where the anchor starts to speak until it switches to a different scene). The transition from one shot to another usually causes an abrupt change

in the histogram and thus can be detected reliably with simple methods. Editing optical transitions to create more pleasant experiences such as fades or dissolves are more challenging to detect, but nevertheless they can also be detected at a satisfactory level [Hanjalic, 2002, Phung, 2005b].

While shot detection is generally considered as a solved problem, seeking high-level semantics that move beyond the shots to carry longer correlation between shots is a challenging problem. Research into this problem is fast growing and depending on the investigating domain, the high-level units appear under different names such as *scene*, *story*, *episode* in motion pictures; *topic*, *subtopic*, *macro segments*, *story units* for information-oriented videos (news, documentaries, training and educational videos), or in more general terms such as *logical story units* used in [Hanjalic et al., 1999, Vendrig and Worring, 2002]. Otherwise stated, we shall use 'scene' in this section to mean all of those aforementioned names, and formally, a scene is defined as "a sequence of consecutive shots whose contents are unified in terms of time, locale and dramatic structures" [Truong et al., 2002].

Some of the earliest work extracts scene-level concepts in broadcast programs, in particular, news videos [Ide et al., 1998, Liu and Huang, 1999, Shearer et al., 2000]. The semantic extraction problem is usually cast as the classification problem in these works. The authors in [Shearer et al., 2000], for example, combine a number of visual and aural low-level features with shot syntax in news videos to group shots into different narrative structures and label them as anchor-shot, voice-over, or interview. Liu and Huang [Liu and Huang, 1999] propose a video/audio fusion approach to segment news reports from other categories in broadcast programs with different types of classifiers (simple threshold method, Gaussian mixture classifier, and support vector machine). Ide et al. [Ide et al., 1998] propose an automatic indexing scheme for news video where shots are indexed based on the image content and keywords into five categories: speech/report, anchor, walking, gathering, and computer graphics. Caption text information is then used with classified shots to build the indices.

Segmentation of the *news story* is the second major theme explored in the broadcast domain. The common underlying approach used in these works is the use of explicit 'rules' about the structure of news programs to locate the transitions of a news

story. Commonly accepted heuristics are for example: a news story often starts and finishes with anchor-person shots [Truong, 2004]; the start of a news story is usually coupled with music [Aigrain et al., 1998]; or a relatively long silent period is the indication of the boundary between two news stories [Wang et al., 2003a]. More complicated rules via temporal analysis are also exploited such as in the work of [Zhu et al., 2001] which utilizes detection results of anchor-persons and captions to form a richer set of rules (i.e, if the same text caption appears in two consecutive anchor-person shots, then they belong to the same news story). There is also a body of work which casts the segmentation problem of news stories in a HMM framework [Iurgel et al., 2001, Chaisorn et al., 2004]. The authors in [Iurgel et al., 2001], for example, propose the news segmentation as the problem of decoding the maximum state sequence of a trained HMM whose transition matrix is tailored by explicit rules about the news program. A somewhat similar approach to the work in this thesis is [Chaisorn et al., 2004] (whose results came first in the TRECVID2003 story segmentation benchmark). Shots are first classified into a set of common labels in news (e.g, anchor, 2anchor, text-scene, etc.). These labels are then input to a HMM for segmentation. They report best performances of 74.9% recall and 80.2% precision for the TRECVID dataset. Their work remains limited due to the flat HMM being used, and it is not clear how the set of 'transition' states were chosen. In an effort to move beyond flat structure the authors have raised the need for high-order statistical techniques.

More recent approaches towards scene extraction have shifted to motion pictures, e.g. [Sundaram and Chang, 2002, Wang et al., 2001, Adams et al., 2001, Truong, 2004]. Detecting scenes in motion pictures is in general a challenging problem and there are three main existing approaches as outlined in [Truong, 2004]: temporal clustering-based, rule-based and memory-based detection. In the *clustering-based* approach, shots are grouped into scenes based on visual similarity and temporal closeness (e.g, [Hanjalic et al., 1999, Lin and Zhang, 2000]). Scene breaks in the *rule-based* detection approach are determined based on the semantic and syntactic analysis of audiovisual characteristics, and in some cases further enhanced with more rigorous grammars from film theory (e.g, [Wang et al., 2001, Adams et al., 2001]). The authors in [Sundaram and Chang, 2002] propose a memory-based scene detection framework. Visual shot similarity in these works is determined based on the consistency in color chromaticality, and the soundtrack is partitioned into 'audio

scenes'. Visual and aural data are then fused within a framework of memory and attention span model to find likely scene breaks or singleton events. Further related background on scene detection can be found in many good surveys (e.g, [Sundaram and Chang, 2002, Snoek and Worring, 2004, Truong, 2004, Phung, 2005b]).

Existing HMM-based approaches for modeling long-term temporal dependencies typically use pre-segmented training data at multiple levels, and hierarchically train a pool of HMMs. HMMs at the lower levels are used as input to the HMMs at the upper levels. In principle, some fundamental units are recognized by a sequence of HMMs, and then likelihood values (or labels) obtained from these HMMs are combined to form a hierarchy of HMMs to capture the interactions at higher semantic levels (e.g, [Kijak et al., 2003, Naphade and Huang, 2002]). Analyzing sports videos, Kijak et al. [Kijak et al., 2003] propose a two-tiered classification of tennis videos using two layers of HMMs. At the bottom level, four HMMs are used to model four shot classes ('first missed serve','rally', 'replay', and 'break'). Each HMM is trained separately and subsequently topped up by another HMM which represents the syntax of the tennis video with three states of the game: {'sets', 'games', and 'points'}. Parameters for the top HMM are, however, all manually specified. In [Naphade and Huang, 2002], a generic two-level hierarchy of HMMs is proposed to detect recurrent events in movies and talk shows. Their idea is to use an ergodic HMM at the top level, in which each state is another (non-ergodic) sub-HMM representing a type of signal property. For the case of movies, the top HMM has six states, and each in turn is another three-state non-ergodic HMM. The observations are modelled as a mixture of Gaussians. After training, the authors claim that interesting events such as 'explosion' and 'male speech' can be detected. While being able to overcome the limitation of the flat HMM in modeling long-term dependencies, approaches that use HMMs at multiple levels still suffer from two major problems: (1) pre-segmented and annotated data are needed at all levels for training, and (2) in most existing work parameterization at higher levels has to be manually specified. In many cases, preparing training data at multiple levels is extremely tedious, and in the worst case, may not be possible. With respect to the second problem, as each semantic level has to be modeled separately, the underlying problem is that the interactions across semantic layers are not modeled and thus do not contribute to the learning process.

One framework that integrates the semantics across layers is the Hierarchical Hidden Markov Model (HHMM) proposed in [Fine et al., 1998]. The hierarchical HMM extends the standard HMM in a hierarchical manner to allow each state to be recursively generalized as another sub-HMM, and thus enabling the ability to handle hierarchical modeling of complex dynamic process, in particular "the ability to infer correlated observations over long periods in the observation sequence via the higher levels of hierarchy" [Fine et al., 1998]. The original motivation in [Fine et al., 1998] was to seek better modeling of different stochastic levels and length scales presented in language (e.g, speech, handwriting, or text). However, the model introduced in [Fine et al., 1998] considers only state hierarchies that have tree structures, disallowing the sharing of substructures among the high-level states. Recognizing this need, the authors in [Bui et al., 2004] have extended the strict tree-form topology in the original HHMMs of [Fine et al., 1998] and allowed it to be a general lattice structure. The extension thus permits a state at any arbitrary level of the HHMMs to be shared by more than one parental states at its higher level (i.e, resulting in a compact form of parameter typing at multiple levels). This extended form is very attractive for video content modeling since it allows the natural organization of the video content to be modeled not only in terms of multiple scales, but also in terms of shared substructures existing in the decomposition.

Early application of the HHMM for video analysis is found in [Xie et al., 2002] and later extended in [Xie and Chang, 2003]. In particular, the authors use the HHMM to detect the events of 'play' and 'break' in soccer videos. For inference and learning, the HHMM is 'collapsed' into a flat HMM with a very large product state space, which can then be used in conjunction with the standard forward/backward passes as in a normal HMM. Four methods are compared in [Xie et al., 2002] to detect 'play' and 'break': (1) supervised HMMs, in which each category is trained with a separate HMM, (2) supervised HHMMs, in which bottom level HMMs are learned separately and parameters for the upper levels are manually specified, (3) unsupervised HHMMs without model adaptation, and (4) supervised HHMMs with model adaptation. In (3) and (4), two-level HHMMs are used. Their results have shown a very close match between unsupervised and supervised methods in which the completely unsupervised method with model adaptation performs marginally better. These figures are $75.5\%, 75.0\%, 75.0\%$ and 75.7% respectively for those four methods. While presenting a novel contribution to the feature selection and model

selection procedure, the application of the HHMMs in their work is still limited, both for learning and for the exploitation of the hierarchical structure. Flattening a HHMM into a flat HMM as reported in [Xie et al., 2002, Xie and Chang, 2003] suffers from many drawbacks as criticized in [Murphy and Paskin, 2001]: (a) it cannot provide multi-scale interpretation, (b) it loses modularity since the parameters for the flat HMM get constructed in a complex manner, and (c) it may introduce more parameters, and most importantly it does not have the ability to reuse parameters. In other words parameters for the shared sub-models are not 'tied' during the learning, but have to be replicated and thus lose the inherent strength of hierarchical modeling.

Being able to model shared structures, the extended HHMMs of [Bui et al., 2004] allow us to build more compact models, which facilitates more efficient inference and reduces the sample complexity in learning. This model is applied in [Phung et al., 2004a] and [Phung et al., 2004b] for the problem of topic transition detection and video structure discovery, respectively. The authors in [Phung et al., 2004a] use a three-level HHMM for the detection of topic transitions in educational videos. Differing from our experiments in this thesis, the HHMM in [Phung et al., 2004a] is modified to operate directly with continuous-valued observed data via the use of Gaussian mixture models as the emission probabilities. Each shot-based observed vector consists of seven features extracted from visual and audio streams. They report a 77.3% recall rate and 70.7% precision for the detection task. In another application, with the help of prior knowledge about educational videos, a topology for a three-level HHMM is used in [Phung et al., 2004b] to automatically discover meaningful narrative units in the educational genre. Their experiments have shown encouraging results in which many meaningful structures are hierarchically discovered such as 'on-screen narration with texts', 'expressive linkage', 'expressive voice-over', etc. The work of [Phung et al., 2004b] is somewhat similar to that of [Naphade and Huang, 2002] except the model in [Phung et al., 2004b] allows more domain knowledge to be encoded and the parameters are all learned automatically. Nevertheless, none of these works supports automatic modeling and learning of durational knowledge.

2.9 Closing remarks

We have presented a composition of related background for this thesis. Related theories were provided in the first part (sections 2.1 to 2.4), while the second part (section 2.5 to 2.8) was concerned with relevant applications. The first part included a revision of Bayesian networks, dynamic Bayesian networks, exponential families and the Hidden Markov Models. We particularly paid attention to inference and learning algorithms in general DBN structures. The second part provided a literature review on the Phase-Type distribution and the application domains investigated in this thesis including activity recognition, anomaly detection, and segmentation and annotation of professionally made video.

The next chapter presents the first contribution in which we provide a thorough investigation, treated under a unified framework of exponential families, into different modeling choices for state durations.

Chapter 3

The Hidden Semi-Markov Models

In this chapter we present an investigation into the state-of-the-art modeling choices for the state duration in the HSMM. We revisit and analyze some existing probability density[1] choices for modeling duration. We aim to treat all of these different duration models under a more generic class of the Exponential Family distributions and highlight their key advantages and disadvantages. This will serve as the motivation for our work on duration modeling in the next chapter.

Therefore, our contributions in this chapter include a thorough investigation into the aspect of duration modeling and a generic DBN representation for the HSMM with different duration distributions of the Exponential Family. Even though Mitchell and Jamieson [Mitchell and Jamieson, 1993] have investigated the use of the Exponential Family distributions to model state durations, our work is different from theirs by the use of graphical representation. This representation makes the models intuitive and allows us to use the available tools and techniques in graphical models for inference and learning. The layout of this chapter is as follows. First, a description on the Hidden Semi-Markov Model (HSMM) is provided in section 3.1. Our focus is, however, in section 3.2, when we provide a detailed explanation on current duration modeling and how it can be framed in the generic Exponential Family. Finally, our conclusions are contained in section 3.3.

3.1 The Hidden Semi-Markov Models

The section presents descriptions of the Hidden Semi-Markov Model (HSMM).

[1]Note that the term "probability density" used here is for both continuous and discrete cases.

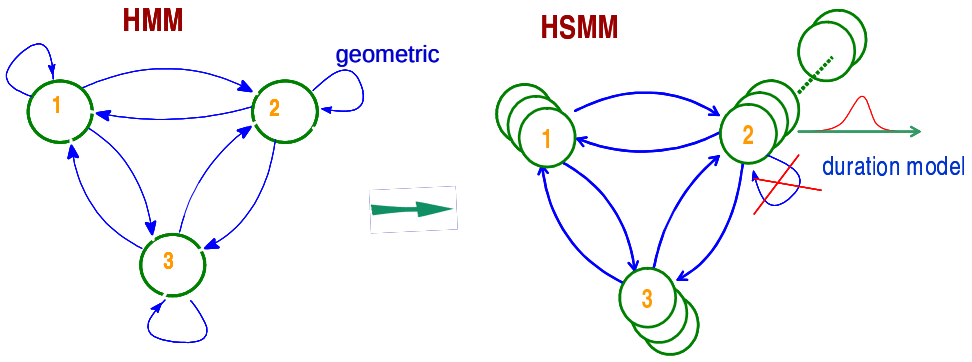


Figure 3.1: From HMM to HSMM.

3.1.1 Model and definitions

In the HMM (section 2.4), the probability mass function for the state duration is characterized by a geometric distribution, which is too restrictive for most cases. Fig. (3.1) shows the evolution from a HMM to a Hidden semi-Markov model (HSMM) by allowing a general distribution for the state duration. While state i remains unchanged during time $t-l+1$ to t, it emits an observation segment $y_{t-l+1:t}$. If $\Pr(y_{t-l+1:t} \mid i) = \prod_{\tau=t-l+1}^{t} \Pr(y_\tau \mid i)$, the model is an explicit HSMM [Rabiner, 1989, Mitchell et al., 1999]. If the factorization also depends on the mean of the segment, then the model is a segmental model [Gales and Young, 1993, Ostendorf et al., 1996]. In this thesis we only consider the explicit HSMM, and for simplicity we henceforth refer to it as HSMM.

As shown in Table (3.1), the HSMM is completely characterized by a state space Q, an observation alphabet V, and a parameter set $\theta_{\text{HSMM}} \triangleq \{\pi, A, D, B\}$. While the initial state distribution π and the observation matrix B are the same as the HMM's, the transition matrix A no longer allows self-transitions. Note that in the HMM the self-transition probability A_{ii} is stochastic and defines the state's inherent geometric duration ($D_i(d) \sim f_{\text{Geom}(1-A_{ii})}(d)$). On the contrary, in the HSMM, the self-transition probability is set to zero and the state duration is characterized by a separate distribution, which is the state duration distribution D_i. If the state duration D_i is geometric (or exponential for continuous time HSMM), the HSMM then reduces to a HMM.

Symbols	**Meanings**
Q	The state space includes $\|Q\|$ mutually exclusive state, $Q = \{1,\ 2,\ \ldots,\ \|Q\|\}$.
V	The observation space consists of $\|V\|$ distinguished alphabets, $V = \{1,\ 2,\ \ldots,\ \|V\|\}$.
M	The maximum duration of any states.
π_i	The probability that the semi-Markov chain will start with state i, $\sum_{i \in Q} \pi_i = 1$.
A_{ij}	The probability that the next state will be j given the current state is i, self transition is not allowed $A_{ii} = 0, \forall i \in Q$, and $\sum_{j \in Q} A_{ij} = 1$.
D_i	The duration distribution for state i.
$B_{v\|i}$	The probability that an alphabet v is generated given the current state is i, $\sum_{v \in V} B_{v\|i} = 1$.
θ_{HSMM}	The HSMM parameter set: $\theta_{\text{HSMM}} = \{\pi,\ A,\ D,\ B\}$.

Table 3.1: Parameters of a HSMM.

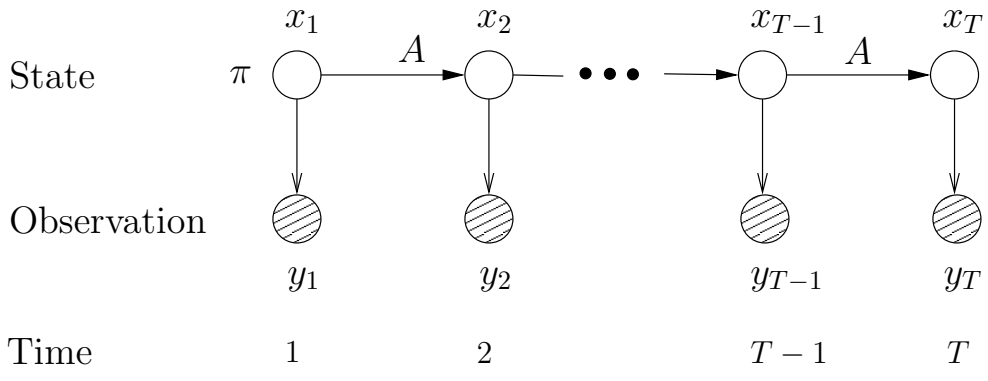


Figure 3.2: DBN representation for the HMM.

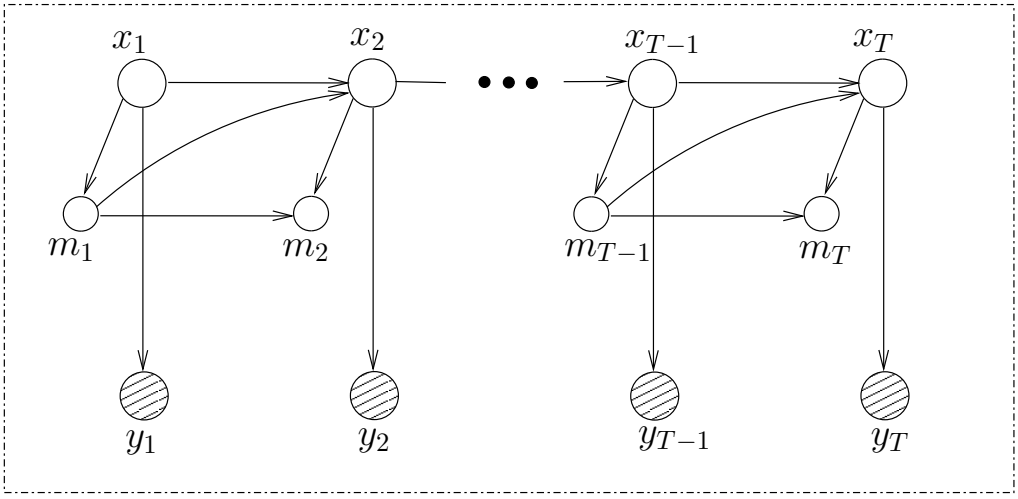


Figure 3.3: DBN representation of a HSMM whose state duration is modeled by a Multinomial or an Exponential Family distribution.

3.1.2 DBN representation, Inference and Learning

In this section we present a generic DBN structure for the HSMM whose state durations can follow the Multinomial or other distributions from the Exponential Family. We introduce a new set of variables $m_{1:T}$, namely the duration variables, into the DBN (Fig. (3.3)). This generic representation of the HSMM makes the duration model more perceptive and enables us to take advantages of available techniques for inference and parameter estimation in a usual DBN setting (e.g., the forward/backward inference and the EM algorithm). Thus, similar to the HMM, learning in the presence of latent variables in the HSMM becomes a learning problem in a generic Bayesian network. Also, for the convenience of comparison we re-present here, Fig. (3.2), the DBN representation of the HMM.

Fig. (3.3) shows that in addition to the state variable x_t and observation variable y_t as in the HMM (Fig. (3.2)), at each time slice we also maintain a duration variable m_t. The duration m_t is a "count-down" variable, which not only specifies how long the current state will last but also acts like the context, defining how the next time slice $t+1$ will be defined from the current time slice t. When $m_t > 1$ the same state x_t carries on to the next time slice, and the state duration reduces by 1: $m_{t+1} = m_t - 1$. On the contrary, when $m_t = 1$, the next state x_{t+1}, where $x_{t+1} \neq x_t$, is drawn from the transition probability $A_{x_t x_{t+1}}$, and the duration variable m_{t+1} is initialized to some random value d with a probability conventionally drawn from a

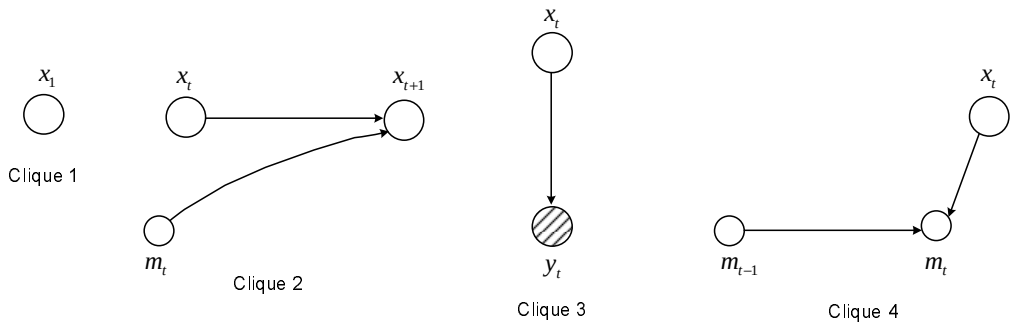


Figure 3.4: Cliques of the HSMM.

Multinomial or an Exponential Family distribution: $\Pr(m_{t+1}^d \mid x_{t+1}^i, m_t^1) = D_i(d)$, for $d > 0$. Again, we use x_t^i as a shorthand for $x_t = i$. The variable m_{t+1} then counts down until it reaches 1. Note that the definition of duration readily satisfies the probabilistic constraints: $\sum_{d=1}^{M} D_i(d) = 1$.

Fig. (3.4) shows the four cliques comprising the HSMM. As compared to cliques of the HMM (Fig. (3.2)), the transition clique (clique 2) is now also affected by the context m_t and a new duration clique (clique 4) is introduced, whose conditional independencies define the duration parameter. We have:

$$\text{Clique 1:} \quad \Pr\left(x_1^i\right) = \pi_i \tag{3.1}$$

$$\text{Clique 2:} \quad \Pr\left(x_{t+1}^j \mid x_t^i,\ m_t^d\right) = \begin{cases} \delta(i,j) & \text{if } d > 1 \text{ (stays in the same state)} \\ A_{ij} & \text{if } d = 1 \text{ (transits to a new state } j\text{)} \end{cases} \tag{3.2}$$

$$\text{Clique 3:} \quad \Pr\left(y_t^v \mid x_t^i\right) \ = B_{v|i} \tag{3.3}$$

$$\text{Clique 4:} \quad \Pr\left(m_t^d \mid x_t^i,\ m_{t-1}^{d'}\right) = \begin{cases} \delta\left(d, d'-1\right) & \text{if } d' > 1 \text{ (stays in the same state)} \\ D_i(d) & \text{if } d' = 1 \text{ (transits to a new state)} \end{cases} \tag{3.4}$$

Inference

The inference tasks for the HSMM include computing the forward variable $\alpha_t(i,d) \triangleq \Pr\left(x_t^i, m_t^d, y_{1:t}\right)$, the backward variable $\beta_t(i,d) \triangleq \Pr\left(y_{t+1:T} \mid x_t^i, m_t^d\right)$, the smoothing distributions $\gamma_t(i,d) = \Pr\left(x_t^i, m_t^d \mid y_{1:T}\right)$ and $\xi_t(i,j,d) \triangleq \Pr\left(x_t^i, x_{t+1}^j, m_{t+1}^d, m_t^1 \mid y_{1:T}\right)$, and also the scaled forward/backward variables: $\tilde{\alpha}_t(i,d) \triangleq \Pr\left(x_t^i, m_t^d \mid y_{1:t}\right)$ and $\tilde{\beta}_t(i,d) \triangleq \frac{\Pr\left(y_{t+1:T} \mid x_t^i, m_t^d\right)}{\Pr\left(t+1:T \mid y_{1:t}\right)}$. All these variables can be computed analogous to that of the HMM in section 2.4, and thus can be directly inferred from the available filtering and smoothing distributions of the HMM by replacing the single hidden

state variable x_t with the grouped variables $\{x_t, m_t\}$. For example, replacing x in Eq. (2.34) by $\{x, m\}$, we obtain the forward recursion for the HSMM case as:

$$\begin{aligned}\alpha_{t+1}(j,d) &\triangleq \Pr\left(x_{t+1}^j, m_{t+1}^d, y_{1:t+1}\right) \\ &= \sum_{i,d'} \Pr\left(y_{t+1} \mid x_{t+1}^j, m_{t+1}^d\right) \Pr\left(x_{t+1}^j, m_{t+1}^d \mid x_t^i, m_t^{d'}\right) \Pr\left(x_t^i, m_t^{d'}, y_{1:t}\right) \\ &= \sum_{i,d'} \Pr\left(y_{t+1} \mid x_{t+1}^j\right) \Pr\left(m_{t+1}^d \mid x_{t+1}^j, m_t^{d'}\right) \Pr\left(x_{t+1}^j \mid x_t^i, m_t^{d'}\right) \Pr\left(x_t^i, m_t^{d'}, y_{1:t}\right)\end{aligned} \tag{3.5}$$

where $\Pr\left(y_{t+1} \mid x_{t+1}^j\right) = B_{y_{t+1}|j}$ and $\Pr\left(x_t^i, m_t^{d'}, y_{1:t}\right) = \alpha_t(i,d')$. The other two terms in Eq. (3.5) are defined based on the conditional independencies of the transition and duration cliques (clique 2 and 4 in Eqs. (3.2) and (3.4)). If from time t to $t+1$: $(i,d') \to (j,d)$ the semi-Markov chain continues in the same state and $\{i = j,\ d' = d+1 > 1\}$, it then follows:

$$\begin{aligned}\Pr\left(x_{t+1}^j \mid x_t^i, m_t^{d'}\right) &= 1 \\ \Pr\left(m_{t+1}^d \mid x_{t+1}^j, m_t^{d'}\right) &= 1\end{aligned}$$

Otherwise, i.e. $i \neq j$ and $d' = 1$:

$$\begin{aligned}\Pr\left(x_{t+1}^j \mid x_t^i, m_t^{d'}\right) &= A_{ij} \\ \Pr\left(m_{t+1}^d \mid x_{t+1}^j, m_t^{d'}\right) &= D_j(d)\end{aligned}$$

Combining the two cases results in:

$$\alpha_{t+1}(j,d) \;=\; B_{y_{t+1}|j}\left[\alpha_t(j,d+1) + \sum_{i \neq j} D_j(d) A_{ij} \alpha_t(i,1)\right] \tag{3.6}$$

Thus, skipping tedious derivations, we list here the final formulae for other recursive

variables:

$$\begin{aligned}
\ddot{\alpha}_{t+1}(j,d) &\triangleq \Pr\left(x_{t+1}^{j}, m_{t+1}^{d}, y_{t+1} \mid y_{1:t}\right) \\
&= B_{y_{t+1}|j}\left[\tilde{\alpha}_t(j,d+1) + \sum_{i\neq j} D_j(d) A_{ij} \tilde{\alpha}_t(i,1)\right] \qquad (3.7)\\
\psi_{t+1} &\triangleq \Pr\left(y_{t+1} \mid y_{1:t}\right) = \sum_{j,d} \ddot{\alpha}_{t+1}(j,d) \qquad (3.8)\\
\tilde{\alpha}_{t+1}(j,d) &= \frac{\ddot{\alpha}_{t+1}(j,d)}{\psi_{t+1}} \qquad (3.9)\\
\beta_t(i,d') &= \begin{cases} \sum_{j,d} B_{y_{t+1}|j} A_{ij} D_j(d) \beta_{t+1}(j,d), & d'=1 \\ B_{y_{t+1}|i} \beta_{t+1}(i,d'-1), & d'>1 \end{cases} \qquad (3.10)\\
\phi_t &\triangleq \Pr\left(y_{t+1:T} \mid y_{1:t}\right) = \phi_{t+1}\psi_{t+1} \qquad (3.11)\\
\tilde{\beta}_t(i,d') &= \frac{\beta_t(i,d')}{\phi_t} \qquad (3.12)
\end{aligned}$$

and the smoothing distributions:

$$\begin{aligned}
\gamma_t(i,d) &= \tilde{\alpha}_t(i,d)\,\tilde{\beta}_t(i,d) \qquad (3.13)\\
\xi_t(i,j,d) &= \frac{B_{y_{t+1}|j} A_{ij} D_j(d) \tilde{\alpha}_t(i,1)\,\tilde{\beta}_{t+1}(j,d)}{\psi_{t+1}} \qquad (3.14)
\end{aligned}$$

Finally, it is clear from the recursive formula in Eq. (3.5) that the inference has a complexity of $O(|Q|^2 M^2 T)$. However, by taking advantage of the deterministic counting process of m_t (i.e. within a given state, $m_{t+1} = m_t - 1$), the complexity is reduced to $O(|Q|^2 MT)$ as shown in Eqs. (3.6) and (3.10).

Learning

Similar to the HMM, the DBN representation of the HSMM enables it to be viewed as a member of the Exponential Family. Hence, in the learning phase, the HSMM parameter set θ_{HSMM} can be estimated using the EM algorithm in a similar fashion. We provide here a summary for parameter estimation formulae, except for state duration distribution which will be detailed separately in the next section.

The initial probability: $\hat{\pi}_i = \frac{\langle T(\pi_i)\rangle}{\sum_i \langle T(\pi_i)\rangle} = \langle T(\pi_i)\rangle = \sum_d \gamma_1(i,d)$

The transition probability: $\hat{A}_{ij} = \frac{\langle T(A_{ij})\rangle}{\sum_j \langle T(A_{ij})\rangle} = \frac{\sum_{t=1}^{T}\sum_d \xi_t(i,j,d)}{\sum_{t=1}^{T}\sum_{j,d}\xi_t(i,j,d)}$

The emission probability: $\hat{B}_{v|i} = \frac{\langle T(B_{v|i})\rangle}{\sum_v \langle T(B_{v|i})\rangle} = \frac{\sum_{t=1}^{T}\sum_d \gamma_t(i,d)\,\delta_{y_t}^{(v)}}{\sum_{t=1}^{T}\sum_{v,d}\gamma_t(i,d)\,\delta_{y_t}^{(v)}}$

3.2 Duration models

Essential to a HSMM is the choice of state duration models. The duration distribution is required to be versatile enough to model complex durations and yet efficient to compute. Existing duration models include the Multinomial [Rabiner, 1989], the Poisson [Russell and Moore, 1985], the Gamma [Levinson, 1986], the Gaussian [Hongeng and Nevatia, 2003] or more generally, the Exponential Family distributions [Mitchell and Jamieson, 1993]. In this section we revisit these duration models, viewing them through the framework of DBN.

Estimation of duration distributions

In the learning phase, we start with the E-step by computing the complete log likelihood. In the expression of the complete log-likelihood $\mathcal{L} = P(x_{1:T}, m_{1:T}, y_{1:T} \mid \theta)$, we collect only terms associated with the duration parameters (i.e., clique 4 of Fig. (3.4)):

$$\begin{aligned}\mathcal{L}_D = \log\prod_{t=1}^{T}\Pr(m_t \mid x_t, m_{t-1}) &= \sum_{t=1}^{T}\log\left\{\prod_{i\in Q}\prod_{d=1}^{M}\Pr\left(m_t^d \mid x_t^i, m_{t-1}^1\right)^{\delta_{m_t}^{(m)}\delta_{x_t}^{(i)}\delta_{m_{t-1}}^{(1)}}\right\} \\ &= \sum_{i,d}\sum_{t=1}^{T}\delta_{m_t}^{(m)}\delta_{x_t}^{(i)}\delta_{m_{t-1}}^{(1)}\log\{D_i(d)\} \qquad (3.15)\end{aligned}$$

The sufficient statistic of the duration parameter $D_i(d)$: $T(D_i(d)) = \sum_{t=1}^{T}\delta_{m_t}^{(m)}\delta_{x_t}^{(i)}\delta_{m_{t-1}}^{(1)}$ counts the number of instances the state duration of state i being initialized to d. Taking the expectation of $\mathcal{L}_D$ over the $\Pr(\text{hidden} \mid \text{observed}) = \Pr(x_{1:T}, m_{1:T} \mid y_{1:T})$ results in:

$$\langle\mathcal{L}_D\rangle = \sum_{i,d}\langle T(D_i(d))\rangle\log\{D_i(d)\} \qquad (3.16)$$

in which the expected sufficient statistics (ESS's) is defined by:

$$\langle T\left(D_i(d)\right)\rangle = \left\langle \sum_{t=1}^{T} \delta_{m_t}^{(d)} \delta_{x_t}^{(i)} \delta_{m_{t-1}}^{(1)} \right\rangle = \sum_{t=1}^{T} \Pr(x_t^i, m_t^d, m_{t-1}^1 \mid y_{1:T}) = \sum_{t=1}^{T} \sum_{j \in Q} \xi_t\left(j, i, d\right) \tag{3.17}$$

In the M-step we need to maximize the expected log-likelihood $\langle \mathcal{L}_D \rangle$ in Eq. (3.17) with respect to the duration parameter D_i, and the maximization method depends on the choice of duration distributions. While a simple Lagrange multiplier method is used for the discrete distributions, the continuous distributions need some other optimization methods, and the choice of optimization methods depends on the distribution itself and the computation costs allowed. In the next sections we will present the M-step separately for the Multinomial and the Exponential Family distributions. Note that even though the Multinomial also belongs to the Exponential Family distributions, it is generally viewed separately as a non-parametric distribution.

3.2.1 The Multinomial Model

The Multinomial is a natural extension of the binomial distribution and has diverse applications in fields such as kinetic theory of classical physics, analysis of contingency tables, population estimation and so on [Johnson et al., 1993]. The Multinomial is the most common choice for duration modeling in the HSMM [Rabiner, 1989, Mitchell et al., 1999, Luhr et al., 2004, Yu and Kobayashi, 2003] due to its simplicity.

Let M be the maximum duration length. The duration of a state i is modeled by a Multinomial as: $D_i \sim Mult\left(D_i(1), D_i(2), \ldots, D_i(M)\right)$, $\sum_{d=1}^{M} D_i(d) = 1$. Using Lagrange theorem (theorem (2.1)) on Eq. (3.16) and subject to the constraint $\sum_{d=1}^{M} D_i(d) = 1$, the re-estimated formula for $D_i(d)$ then follows as:

$$\hat{D}_i\left(d\right) = \frac{\langle T\left(D_i(d)\right)\rangle}{\sum_{d=1}^{M} \langle T\left(D_i(d)\right)\rangle} \tag{3.18}$$

Substituting the ESS's in Eq. (3.17) into Eq. (3.18) results in:

$$\hat{D}_i\left(d\right) = \frac{\sum_{t=1}^{T} \sum_{j \in Q} \xi_t\left(j, i, d\right)}{\sum_{t=1}^{T} \sum_{j,d} \xi_t\left(j, i, d\right)} \tag{3.19}$$

3.2.2 The Exponential Family Model

The Exponential Family includes a rich set of distributions such as binomial, Poisson, Gaussian, Inverse Gaussian, Gamma, etc. The probability of a state i having a duration d following an Exponential Family distribution takes the following form:

$$D_i(d) \triangleq h(d) \exp\left(\mathbf{w}^{\mathsf{T}} T(d) - A(\mathbf{w})\right) \tag{3.20}$$

where the function $h(m)$ is not of fundamental importance as its existence is only to make sure $\sum_d D_i(d) = 1$ (or $\int_d D_i(d) = 1$ for continuous case) and it also plays no role in the M-step. Playing more important roles are the natural parameter $\mathbf{w}$, the sufficient statistics $\mathrm{T}(d)$, and the log partition function $A(\mathbf{w})$. The log partition function is a log normalization factor: $A(\mathbf{w}) = \log \sum_d \exp\left(\mathbf{w}^{\mathsf{T}} \mathrm{T}(d)\right)$ for discrete d.

Substituting $D_i(d)$ in Eq. (3.20) into Eq. (3.16) results in:

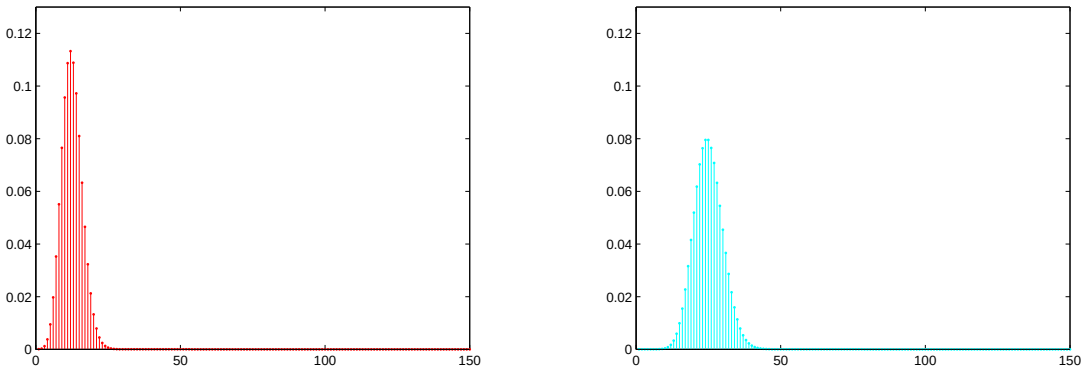
$$\langle \mathcal{L}_D \rangle = \sum_{i,d} \langle T(D_i(d)) \rangle \mathbf{w}^{\mathsf{T}} \mathrm{T}(d) - \sum_{i,d} \langle T(D_i(d)) \rangle A(\mathbf{w}) \tag{3.21}$$

In maximizing the expected log $\langle \mathcal{L}_D \rangle$, we consider two popular and useful distributions from the Exponential Family: the Poisson (discrete) and Inverse Gaussian (continuous).

3.2.2.1 The Poisson distribution

The Poisson distribution arises from considering limiting forms of the binomial distribution. It is useful in situations where the number of independent trials is very large, while the probability of occurrence of an outcome is very small. The Poisson distribution has a wide range of applications such as in measuring the arrivals and departures in data networks, the number of particles emitted by a radioactive source, the number of deaths from being kicked by mules in the Prussian Army Corps [Johnson et al., 1993], etc. In particular, the Poisson is chosen as our example because of its simplicity and its good results in modeling state duration in the HSMM for speech recognition [Russell and Moore, 1985].

The state duration i modeled by a Poisson with mean λ_i has the following form:



(a) $\lambda = 12.5$

(b) $\lambda = 25$

(c) $\lambda = 50$

(d) $\lambda = 100$

Figure 3.5: Examples of Poisson probability mass function with $\lambda = 12.5,\ 25,\ 50,$ and 100.

$$D_i(d) = \frac{1}{d!}\lambda_i^d \exp(-\lambda_i) = \frac{1}{d!}\exp\left(d\log\lambda_i - \lambda_i\right) \tag{3.22}$$

Examples of the probability mass function of the Poisson distribution are shown in Fig. (3.5). Eq. (3.22) shows that Poisson belongs to Exponential Family, with:

$$\begin{aligned} &\text{Natural parameter:} && w_i = \log\lambda_i \\ &\text{Sufficient Statistics:} && T(d) = d \\ &\text{Log partition function:} && A(w_i) = \lambda_i = \exp(w_i) \end{aligned}$$

Therefore, it follows from Eq. (3.21) that:

$$\langle\mathcal{L}_D\rangle = \sum_{i,d}\langle T\left(D_i(d)\right)\rangle w_i d - \sum_{i,d}\langle T\left(D_i(d)\right)\rangle \exp(w_i) \tag{3.23}$$

Differentiating Eq. (3.23) with respect to w_i leads to:

$$\frac{\delta\langle\mathcal{L}\rangle}{\delta w_i} = \sum_{d}\langle T\left(D_i(d)\right)\rangle d - \sum_{d}\langle T\left(D_i(d)\right)\rangle \exp(w_i) \tag{3.24}$$

Setting the derivative to zero $\frac{\delta\langle\mathcal{L}\rangle}{\delta w_i} = 0$, we obtain the re-estimated formula for $\lambda_i = \exp(w_i)$:

$$\hat{\lambda} = \exp(\hat{w}_i) = \frac{\sum_d\langle T\left(D_i(d)\right)\rangle d}{\sum_d\langle T\left(D_i(d)\right)\rangle} = \frac{\sum_{t=0}^{T-1}\sum_{j,d}\xi_t(j,i,d)\,d}{\sum_{t=0}^{T-1}\sum_{j,d}\xi_t(j,i,d)} \tag{3.25}$$

The above expression is intuitive as the Poisson parameter λ represents the duration mean.

3.2.2.2 The Inverse Gaussian distribution

The Inverse Gaussian (IG) distribution was originally named by Tweedie [Johnson et al., 1994] due to the inverse relationship between its cumulant generating function and that of the Gaussian distributions. IG is also called the Wald distribution or Inverse Normal distribution. It was employed to study the movement of particles subject to Brownian motion and often appeared in Russian studies on electronics and radio techniques. Further, the IG distribution is also well-known in sequential analysis [Johnson et al., 1994]. We choose to investigate IG as an example of continuous distributions in the Exponential Family because it is restricted to the positive domain and has been used to model patients' staying time in hospital [Seshadri,

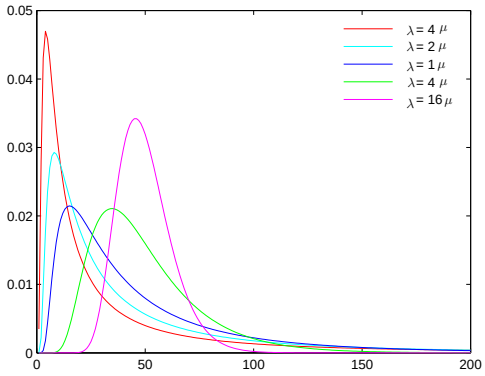


Figure 3.6: Examples of Inverse Gaussian distributions with $\mu = 50$, and $\lambda = 4\mu,\ 2\mu,\ \mu,\ 4\mu,$ and 16μ

.

1993] with successful results. The IG probability density function, illustrated by examples in Fig. (3.6), is defined on $d \in (0, +\infty)$ and takes the following form:

$$\mathrm{IG}(\mu, \lambda) = \Pr(d \mid \mu, \lambda) = \frac{\sqrt{\lambda}}{\sqrt{2\pi d^3}} \exp\left(-\frac{\lambda}{2\mu^2}\frac{(d-\mu)^2}{d}\right)$$

in which μ is the distribution mean and $\frac{\mu^3}{\lambda}$ is its variance. Let $\left(w_i^{(1)}, w_i^{(2)}\right) = \left(-\frac{\lambda}{2}, -\frac{\lambda}{2\mu^2}\right)$ (we start introducing i here to associate the distribution with a state i), then the IG distribution can be written in Exponential Family form as:

$$\begin{aligned}\mathrm{IG}(\mathbf{w}_i) = \Pr(d \mid \mathbf{w}_i) &= \frac{d^{-\frac{3}{2}}}{\sqrt{2\pi}} \exp\left(\frac{w_i^{(1)}}{d} + w_i^{(2)} d - A\left(w_i^{(1)}, w_i^{(2)}\right)\right) \\ &\propto \exp\left(\frac{w_i^{(1)}}{d} + w_i^{(2)} d - A\left(w_i^{(1)}, w_i^{(2)}\right)\right) \end{aligned} \tag{3.26}$$

where the log partition function is:

$$A\left(w_i^{(1)}, w_i^{(2)}\right) = -2\sqrt{w_i^{(1)}, w_i^{(2)}} - \frac{1}{2}\log\left(-w_i^{(1)}\right) \tag{3.27}$$

Clearly, the sufficient statistics is: $T(d) = \left[\frac{1}{d}, d\right]$. The IG is a continuous distribution, thus when applying it into modeling discrete state duration we have to introduce an additional normalization term:

$$\mathrm{N}(\mathbf{w}_i) = \sum_d \exp\left(\mathbf{w}^{\mathsf{T}} T(d) - A(\mathbf{w}_i)\right) \tag{3.28}$$

Thus, the state duration i modeled by an IG distribution is defined by:

$$D_i(d) = \frac{\mathrm{IG}\left(\mathbf{w}_i\right)}{\mathrm{N}\left(\mathbf{w}_i\right)} = \frac{\exp\left(\mathbf{w}^{\mathsf{T}} T\left(d\right) - A\left(\mathbf{w}_i\right)\right)}{\mathrm{N}\left(\mathbf{w}_i\right)} \tag{3.29}$$

with the log partition function $A\left(\mathbf{w}_i\right)$ and the normalization factor $\mathrm{N}\left(\mathbf{w}_i\right)$ are defined by Eqs. (3.27) and (3.28), respectively. Substituting $D_i(d)$ in Eq. (3.29) into Eq. (3.16) results in:

$$\left\langle \mathcal{L}_D \right\rangle = \sum_{i,d} \left\langle T\left(D_i(d)\right)\right\rangle \mathbf{w}_i^{\mathsf{T}} T\left(d\right) - \sum_{i,d} \left\langle T\left(D_i(d)\right)\right\rangle A\left(\mathbf{w}_i\right) - \sum_{i,d} \left\langle T\left(D_i(d)\right)\right\rangle \log\left\{\mathrm{N}\left(\mathbf{w}_i\right)\right\} \tag{3.30}$$

Thus, compared to the standard result in Eq. (3.21), we have an extra (and problematic!) term $\sum_{i,d} \left\langle T\left(D_i(d)\right)\right\rangle \log\left\{\mathrm{N}\left(\mathbf{w}_i\right)\right\}$ to optimize that will require approximation. Theoretically we can choose any black-box optimization method (e.g., steepest descent, conjugate gradient, etc.). In this work, to avoid the costly computation of Jacobian or Hessian, we choose to optimize it by the Nelder-Mead method [Mathews and Fink, 1999], which requires neither the first nor the second derivative, but only function evaluations. The Nelder-Mead search is used to find the local maximum value for the expected log in Eq. (3.30) with respect to the two natural parameters $w_i^{(1)}$ and $w_i^{(2)}$. The search starts with an initial triangle (called the simplex) in the $w_i^{(1)} - w_i^{(2)}$ plane. At each vertex of the triangle, a value for $\left\langle \mathcal{L}_D \right\rangle$ is evaluated. The worst vertex, where $\left\langle \mathcal{L}_D \right\rangle$ is smallest, will be discarded. A new point in the $w_i^{(1)} - w_i^{(2)}$ plane is chosen to form a new triangle and the search continues until a convergence with respect to a predefined tolerance.

3.2.3 State Duration Models and Computational Issues

Modeling state duration by the Multinomial and other distributions from the Exponential Family suffers from a common *major* drawback. That is, the substantial increase in computational load and storage as inference depends linearly on the maximum possible duration M, i.e. $O(|Q|^2 MT)$. This is because the only way to represent state durations at each time slice is to explicitly count them (Fig. (3.3)). In addition, we have to face the problem of determining M in advance, and often in many cases, M can be as large as T. That may require us to use domain knowledge and perhaps intuition to pick a good value for M and then truncate the duration domain to M in inference and learning. The Multinomial also has another shortcom-

ing, which is the large number (i.e. $M-1$) of additional parameters required for each state. This drawback could be a serious problem resulting in overfitting, when only limited data is available for training. In addition, whereas the discrete Exponential Family (e.g. the Poisson) can be estimated in a closed-form via the Baum-Welch reestimation process, the continuous Exponential Family (e.g. the Inverse Gaussian) suffers from another disadvantage in that it requires numerical solutions when applied to the discrete domain. Therefore, it can be concluded that the problem of effective modeling of duration is still unsolved.

3.3 Closing remarks

In this chapter we explain how duration can be integrated into the HMM to form the HSMM. We investigate existing state duration models including the non-parametric Multinomial and the Exponential Family distribution in a new approach with graphical models, facilitating learning and inference procedures. In addition, we analyze existing duration models to understand their weaknesses. The next chapter will address the limitations of these models.

Chapter 4

The Coxian Hidden semi-Markov Model and its Applications

As shown in chapter 3, the problem of effective duration modeling in the HSMM is left unresolved with existing duration models. To solve this problem, in this chapter we propose the discrete Coxian distribution [Cox, 1955], a special case of the Phase-Type distribution [Neuts, 1989], to model state duration in the HSMM and form the (discrete) Coxian Hidden Semi-Markov Model (CxSHSMM). Further, we argue that in the work of modeling and recognizing human activities of daily living (ADLs), temporal information plays a very important role and the Coxian is a suitable candidate to capture this temporal dependency. As to the significance of temporal information, it is natural to see that duration is such a dominant characteristic of ADLs, e.g. "cooking dinner" is a significantly longer task than "brushing teeth". Duration information becomes even more important when it comes to distinguishing activities of the same type such as "cooking dinner" and "preparing a snack".

We apply the CxHSMM to automatic learning and recognition of ADLs and compare its performances with the other existing HSMMs and the standard HMM. Our contributions in this chapter include:

- A review of Phase-Type and Coxian distributions with elaborations on simplified cdf/pdf forms for Coxian distributions with non-identical phases and equations for computing Coxian distribution's mean and variance.
- A *novel* stochastic model named the (discrete) Coxian Hidden Semi-Markov Model (CxSHSMM), which is a Hidden semi-Markov model whose state du-

ration is modeled by the (discrete) Coxian distribution. The use of Coxian distribution has several advantages over traditional parameterization (e.g. Multinomial or Exponential Family) including low numbers of parameters, the existence of closed-form solutions, computational efficiency and denseness in the field of non-negative distributions.

- A complete analysis of the novel CxHSMM includes its dynamic Bayesian network representation, inference and maximum likelihood estimation for fully, partially observed models and with missing observations.

- A full comparison between the proposed CxHSMM and existing models including the Multinomial HSMM, the Exponential Family HSMM and the HMM (without duration modeling) at recognizing ADLs. The significance of this experiment is twofold. First, it is a thorough investigation into activity recognition in a smart home surveillance scenario of all available HSMM variants. Second, it helps to demonstrate the outstanding performance of the CxHSMM in many aspects, most importantly, its high recognition accuracy and low computational cost, making it especially suitable in recognizing ADLs whose movement trajectories are typically very long in nature.

4.1 The Discrete Phase-Type distribution

The discrete Coxian distribution belongs to a more generic class of distributions known as discrete phase-type (PH) distributions. A discrete PH [Neuts, 1981, 1989] is associated with a finite-state (discrete-time) Markov chain (MC) of finite $\mathcal{M}$ *transient states,* numbered from 1 to $\mathcal{M}$, and a single *absorbing state.* The MC starts in any transient state with initial probabilities $\mu_m \in [\boldsymbol{\mu}]_{\mathcal{M}\times 1}$, and in absorbing state with probability $1 - \sum_{m=1}^{\mathcal{M}} \mu_m$. After entering a transient state (*phase*) m, the MC stays in it for a period of time defined by the state's self transition probability $a_{mm} \in [\boldsymbol{A}]_{\mathcal{M}\times\mathcal{M}}$, before moving to the next transient state $n \neq m$ with transition probability $a_{mn} \in [\boldsymbol{A}]_{\mathcal{M}\times\mathcal{M}}$ or reaching the absorbing state with an ending probability $e_m \in [\boldsymbol{e}]_{\mathcal{M}\times 1}$. Probabilistic constraints require: $\boldsymbol{e} = \mathbf{1} - \boldsymbol{A}\mathbf{1}$, where $\mathbf{1}$ is an $\mathcal{M} \times 1$ vectors of 1. The MC never leaves the absorbing state. The MC for which there is a probability of 1 of ending up in an absorbing state is called the *absorbing* MC, and the total time since the MC is initialized until it comes to the absorbing state is called the *time to absorption.* The distribution of this time to absorption (i.e. the

distribution of the total number of transitions including self transitions within transient states required before absorption) is called the *discrete Phase-Type distribution*.

Definition 4.1. A discrete Phase-Type (PH) distribution, denoted as $\mathrm{PH}(\boldsymbol{\mu}, \boldsymbol{A})$, is the distribution of the total time $\boldsymbol{\tau} \in \mathbf{N}^0$ from *initialization till absorption* of an absorbing, finite-state, discrete-time Markov chain with a single absorbing state. The parameter set $(\boldsymbol{\mu}, \boldsymbol{A})$ is called the *representation* of the discrete PH distribution: $\boldsymbol{\mu}$ is the initial probabilities of transient states and $\boldsymbol{A}$ is the transition matrix between transient states of the Markov chain.
The discrete PH cumulative distribution function:

$$F_{\mathrm{PH}(\boldsymbol{\mu},\boldsymbol{A})}(d) = \Pr(\boldsymbol{\tau} \leq d \mid \boldsymbol{\mu}, \boldsymbol{A}) = 1 - \boldsymbol{\mu}^{\mathsf{T}} \boldsymbol{A}^{d} \mathbf{1} \tag{4.1}$$

The discrete PH probability mass function:

$$f_{\mathrm{PH}(\boldsymbol{\mu},\boldsymbol{A})}(d) = \Pr(\boldsymbol{\tau} = d \mid \boldsymbol{\mu}, \boldsymbol{A}) = \boldsymbol{\mu}^{\mathsf{T}} \boldsymbol{A}^{d-1} \boldsymbol{e} \tag{4.2}$$

■

Eq. (4.2) is intuitive as it shows that the time to absorption d is counted once the MC is initialized with probability $\boldsymbol{\mu}$ in its transient state, then stays in and transits within transient states for a period of $d-1$ with probability $\boldsymbol{A}^{d-1}$, and eventually comes to an end in its absorbing state in a single step (one time unit) with probability $\boldsymbol{e}$.

If the MC is restricted to start only in its transient states, the factorial moments of the discrete PH distribution function is given by:

$$E[\boldsymbol{\tau}(\boldsymbol{\tau}-1)\ldots(\boldsymbol{\tau}-n+1)] = n!\boldsymbol{\mu}^{\mathsf{T}} (\mathbb{I} - \boldsymbol{A})^{-n} \boldsymbol{A}^{n-1} \mathbf{1} \tag{4.3}$$

where $\mathbb{I}$ is an $\mathcal{M} \times \mathcal{M}$ identity matrix, and it is worth noting that the condition of an *absorbing* MC ensures $\mathbb{I} - \boldsymbol{A}$ to be a non-singular matrix.

Closure Properties:

It is shown in [Neuts, 1989] that under finite mixture and finite convolution products, the discrete PH distribution is closed and the resulting representation can be derived from the component representations.

Examples of discrete PH distribution:

The discrete PH distribution is a generalization of geometric distribution [Johnson and Kotz, 1969], and we present some examples.

- **The geometric distribution:** The geometric distribution is the distribution of the number of Bernoulli trials required to get one success. It is the simplest case of discrete PH distribution having only 1 phase. Its underlying MC stays in this phase (transient state) as long as a success is not registered (Fig. (4.1)(a)). In addition, it is straightforward to recognize that the familiar cdf and pmf of the geometric distribution (Geom (a)) is a special case of the discrete PH distribution with only one phase having self transition a: $\boldsymbol{\mu} = [1]$, $\boldsymbol{A} = [a]$ and $\boldsymbol{e} = [1 - a]$,

$$F_{\mathrm{Geom}(a)}(d) = F_{\mathrm{PH}(1,a)}(d) = 1 - a^d \tag{4.4}$$

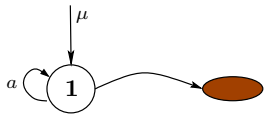
$$f_{\mathrm{Geom}(a)}(d) = f_{\mathrm{PH}(1,a)}(d) = a^{d-1}\,(1 - a) \tag{4.5}$$

- **The negative binomial distribution:** The negative binomial distribution is the distribution of the number of Bernoulli trials to produce N successes for a pre-determined number N. Clearly, when $N = 1$, it reduces to a geometric distribution. The negative binomial distribution, thus, requires N identical phases: its underlying MC always starts in the first phase and once a success is produced in phase $n < N$, the process moves to the next phase $n + 1$ and waits there till another success; finally it comes to an end in the absorbing state after the N^{th} success drawn in the last phase (Fig. (4.1)(b)).

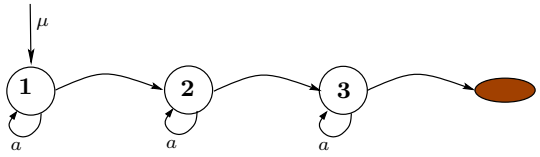
Figs. (4.1) - (4.3) show some examples of phase diagrams and pmfs from the discrete PH distribution. The pmfs of distribution from the PH family tend to lean to the left of their means, and have long tails on the right.

4.2 The Discrete Coxian distribution

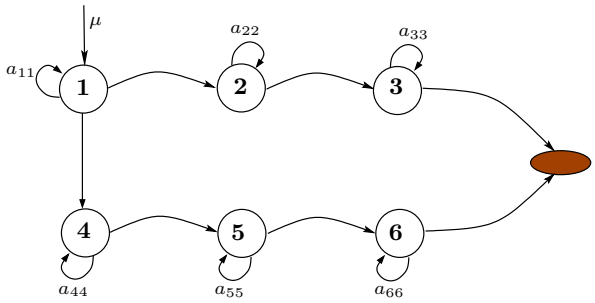
The Coxian distributions are introduced in 1955 by David Cox in [Cox, 1955]. It is one of two important subfamilies of the PH distribution, which are dense in the field of non-negative discrete distributions [Johnson and Taaffe, 1988] (the other subfamily is the finite mixture of discrete Erlang distributions). The Coxian distribution



(a) The geometric distribution.



(b) The negative binomial distribution: $N = 3$ phases (3 successes).



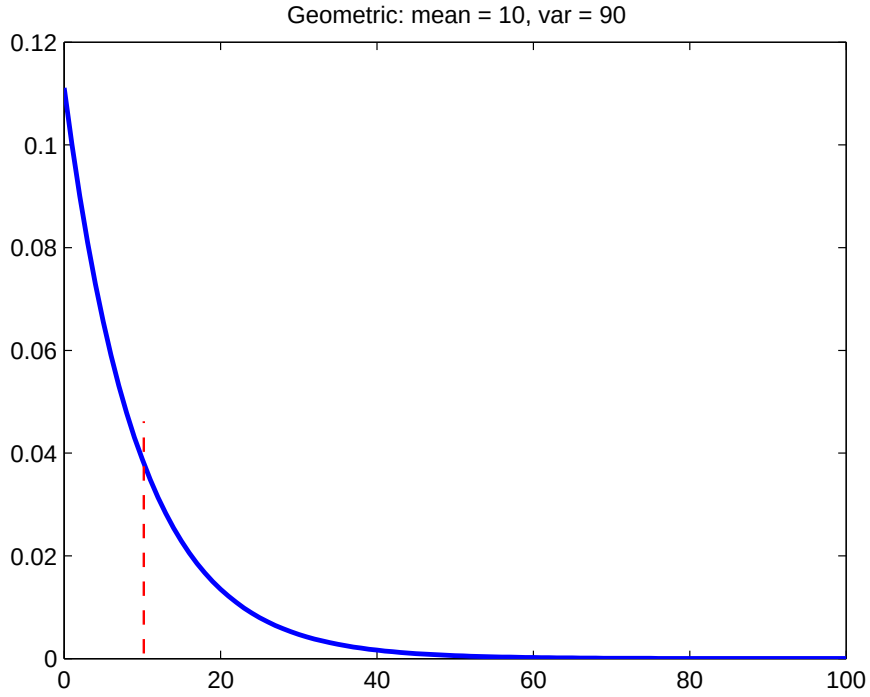
(c) A example of general discrete PH distribution.

Figure 4.1: Examples of discrete PH distribution: The Phase Diagrams.

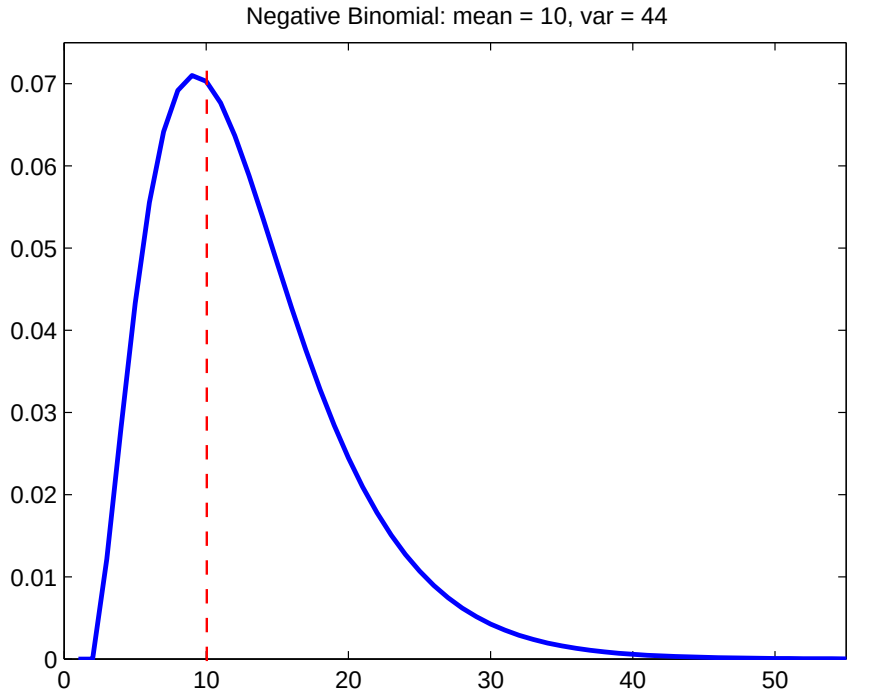
is appealing to us due to its simple underlying MC and its phase decomposition, making it useful to model ADLs as we usually need to conduct several "tasks" in sequence in order to complete activities, and each "task" is analogous to a phase in the Coxian distribution.

The Markov chain underlying the discrete Coxian distribution is a strictly left-to-right Markov chain. Fig. (4.4) shows a left-to-right Markov chain with $\mathcal{M}+1$ states numbered from 1 to $\mathcal{M}$, with the self transition parameter λ_i and an absorbing state. The first $\mathcal{M}$ transient states represent the $\mathcal{M}$ phases, while the last state is absorbing and acts like an end state. For every transient state $m \in [1, \mathcal{M}]$, its duration X_m is a random variable whose distribution is geometric with parameter λ_m: $X_i \sim \text{Geom}(\lambda_i)$. If we start from state m, $S_m = X_m + \ldots + X_{\mathcal{M}}$ is the duration of the Markov chain before the end state is reached. Thus, the Coxian distribution $\text{Cox}(\boldsymbol{\mu}, \boldsymbol{\lambda})$ is in fact the distribution of the duration of this constructed Markov chain when $\boldsymbol{\mu}$ is the initial state distribution.

Definition 4.2. A discrete $\mathcal{M}$-phase Coxian distribution with parameters $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_{\mathcal{M}}]^\mathsf{T}$ and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_{\mathcal{M}}]^\mathsf{T}$, denoted by $\text{Cox}(\boldsymbol{\mu}, \boldsymbol{\lambda})$, where $0 \leq \mu_i \leq 1$,

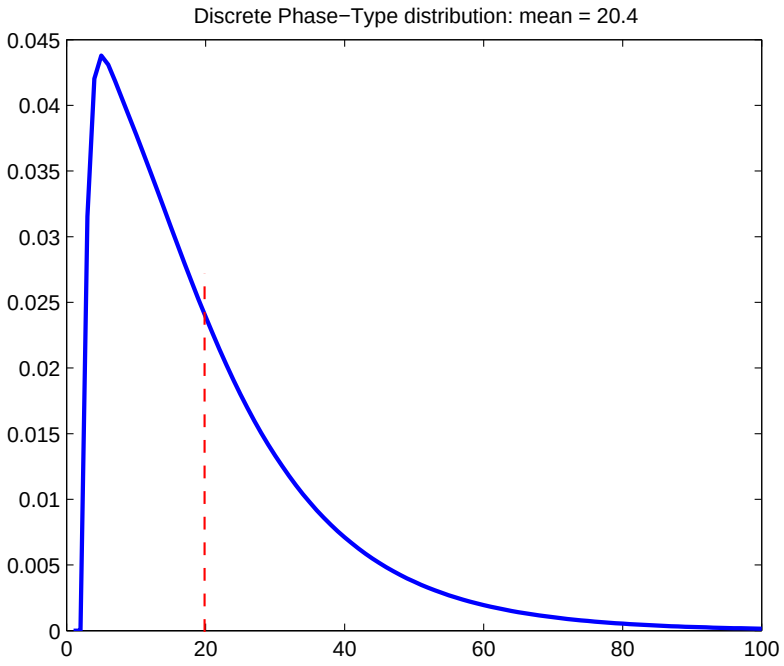


(a) The Geometric distribution: a=0.9.

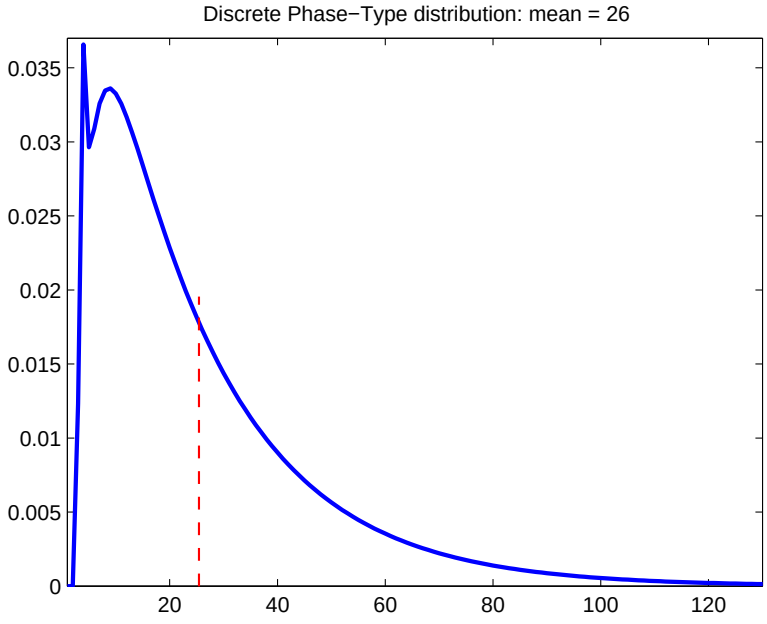


(b) The negative distribution: a =0.77, N=3.

Figure 4.2: Examples of pmfs of discrete PH distributions.



(a)



(b)

Figure 4.3: Examples of pmfs randomly generated from the PH distribution's phase diagram in Fig. (4.1)(c).

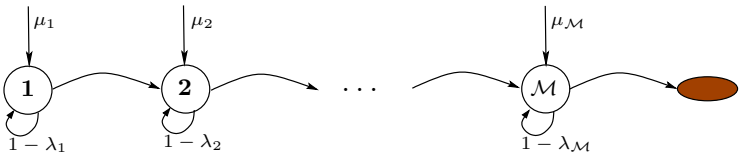


Figure 4.4: The phase diagram of an $\mathcal{M}$-phase Coxian.

$\sum \mu_i = 1$, $0 < \lambda_i \leq 1$, is defined as the mixture:

$$\text{Cox}\left(\boldsymbol{\mu}, \boldsymbol{\lambda}\right) = \text{Mix}(\mu_1, S_1; \ldots; \mu_{\mathcal{M}}, S_{\mathcal{M}}) = \sum_{m=1}^{\mathcal{M}} \mu_m S_m \tag{4.6}$$

where $S_m = X_m + X_{m+1} + \ldots + X_{\mathcal{M}}$; $X_{n\in[1,\mathcal{M}]}$ are independent variables having geometric[1] distributions $X_i \sim \text{Geom}(1-\lambda_i)$.

■

It is straightforward to see that the discrete Coxian is a special case of the PH distribution with the transition matrix and absorbing vector given as:

$$\boldsymbol{A} = \begin{bmatrix} 1-\lambda_1 & \lambda_1 & 0 & 0 & 0 \\ 0 & 1-\lambda_2 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ldots & \ldots & 0 \\ 0 & 0 & 0 & 1-\lambda_{\mathcal{M}-1} & \lambda_{\mathcal{M}-1} \\ 0 & 0 & 0 & 0 & 1-\lambda_{\mathcal{M}} \end{bmatrix}, \quad \boldsymbol{e} = \begin{bmatrix} 0 \\ 0 \\ \ldots \\ 0 \\ \lambda_{\mathcal{M}} \end{bmatrix} \tag{4.7}$$

Thus, given the above conversion, the Coxian cdf and pmf follow the same forms as those of the discrete PH distribution in Eqs. (4.1) and (4.2):

$$F_{\text{Cox}(\boldsymbol{\mu},\boldsymbol{A})}(d) = 1 - \boldsymbol{\mu}\boldsymbol{A}^d\boldsymbol{I} \tag{4.8}$$

$$f_{\text{Cox}(\boldsymbol{\mu},\boldsymbol{A})}(d) = \boldsymbol{\mu}\boldsymbol{A}^{d-1}\boldsymbol{e} \tag{4.9}$$

[1]When considering the continuous Coxian, the geometric distribution is replaced by its continuous counterpart, the exponential distribution.

However, if the Coxian distribution has $\mathcal{M}$ non-identical phases, i.e. $(1-\lambda_n) \neq (1-\lambda_m)$ for any $n \neq m$, we can efficiently compute Eqs. (4.8) - (4.9) by using matrix diagonalization techniques on $\boldsymbol{A}$.

Cdf/pdf for Coxian with non-identical phases:

As the determinant of triangular matrix is equal to the product of its diagonal entries, we have for any geometric phase $\lambda_n \in \boldsymbol{\lambda}$:

$$\det\left(\boldsymbol{A} - (1-\lambda_n)\,\mathbb{I}\right) = (\lambda_n - \lambda_1)\,(\lambda_n - \lambda_2)\ldots(\lambda_n - \lambda_{n-1})\,0\,(\lambda_n - \lambda_{n+1})\ldots(\lambda_n - \lambda_{\mathcal{M}}) = 0$$

which means $[1-\lambda_1,\ \ldots, 1-\lambda_{\mathcal{M}}]^{\mathsf{T}}$ is the set of eigenvalues of $\boldsymbol{A}$. The eigenvalues are organized into an eigenvalue diagonal matrix $\boldsymbol{\Lambda}$:

$$\boldsymbol{\Lambda} = \begin{bmatrix} 1-\lambda_1 & & \\ & \ddots & \\ & & 1-\lambda_{\mathcal{M}} \end{bmatrix}$$

Let $\mathbf{u}_n$, for $n \in [1, \mathcal{M}]$, be the unit eigenvector associated with eigenvalue λ_n, which is the normalized solution of $(\boldsymbol{A} - (1-\lambda_n)\,\mathbb{I})\,\mathbf{u}_n = \mathbf{0}$, and $\mathbf{U} = [\mathbf{u}_1;\ \ldots;\mathbf{u}_{\mathcal{M}}]$ be the eigenvector matrix. The assumption of Coxian with non-identical phases guarantees there are $\mathcal{M}$ distinguished eigenvalues, which means $\mathbf{u}_{n\in[1,\mathcal{M}]}$ are linearly independent and $\boldsymbol{A}$ is diagonalizable [Strang, 2003]:

$$\boldsymbol{A} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{-1} \tag{4.10}$$

It then follows:

$$\boldsymbol{A}^d = \left(\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{-1}\right)\left(\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{-1}\right)\ldots\left(\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{-1}\right) = \mathbf{U}\boldsymbol{\Lambda}^d\mathbf{U}^{-1} \tag{4.11}$$

where the d^{th}multiplication of $\boldsymbol{\Lambda}$ is simply as:

$$\boldsymbol{\Lambda}^d = \begin{bmatrix} \lambda_1^d & & \\ & \ddots & \\ & & \lambda_{\mathcal{M}}^d \end{bmatrix}$$

Thus, the cdf and pdf for Coxian with non-identical phases can be simplified as:

$$F_{\mathrm{Cox}(\boldsymbol{\mu},\boldsymbol{A})}(d) = 1 - \boldsymbol{\mu}\mathbf{U}\boldsymbol{\Lambda}^d\mathbf{U}^{-1}\mathbb{I} \tag{4.12}$$

$$f_{\mathrm{Cox}(\boldsymbol{\mu},\boldsymbol{A})}(d) = \boldsymbol{\mu}\mathbf{U}\boldsymbol{\Lambda}^{d-1}\mathbf{U}^{-1}\boldsymbol{e} \tag{4.13}$$

with the eigenvector matrix $\mathbf{U}$ and eigenvalue matrix $\boldsymbol{\Lambda}$.

Mean and Variance:

It is more convenient to derive the Coxian mean and variance from its definition in Eq. (4.6), based on a cumulative mixture of independent geometrics rather than from its pmf. The mean and variance of the component geometric distribution are computed as follows:
The first moment (mean):

$$\begin{aligned} E[X_n] &= \sum_{d=1}^{+\infty} d\,(1-\lambda_n)^{d-1}\,\lambda_n = \frac{\lambda_n}{1-\lambda_n}\sum_{d=1}^{+\infty}\frac{(1-\lambda_n)^d}{d^{-1}} \\ &= \frac{\lambda_n}{1-\lambda_n}\mathbf{Li}_{-1}(1-\lambda_n) = \frac{\lambda_n}{1-\lambda_n}\frac{1-\lambda_n}{\lambda_n^2} \\ &= \frac{1}{\lambda_n} \end{aligned} \tag{4.14}$$

The second moment:

$$\begin{aligned} E\left[X_n^2\right] &= \sum_{d=1}^{+\infty} d^2(1-\lambda_n)^{d-1}\lambda_n = \frac{\lambda_n}{1-\lambda_n}\mathbf{Li}_{-2}(1-\lambda_n) \\ &= \frac{\lambda_n}{1-\lambda_n}\frac{(1-\lambda_n)(2-\lambda_n)}{\lambda_n^3} = \frac{2-\lambda_n}{\lambda_n^2} \end{aligned} \tag{4.15}$$

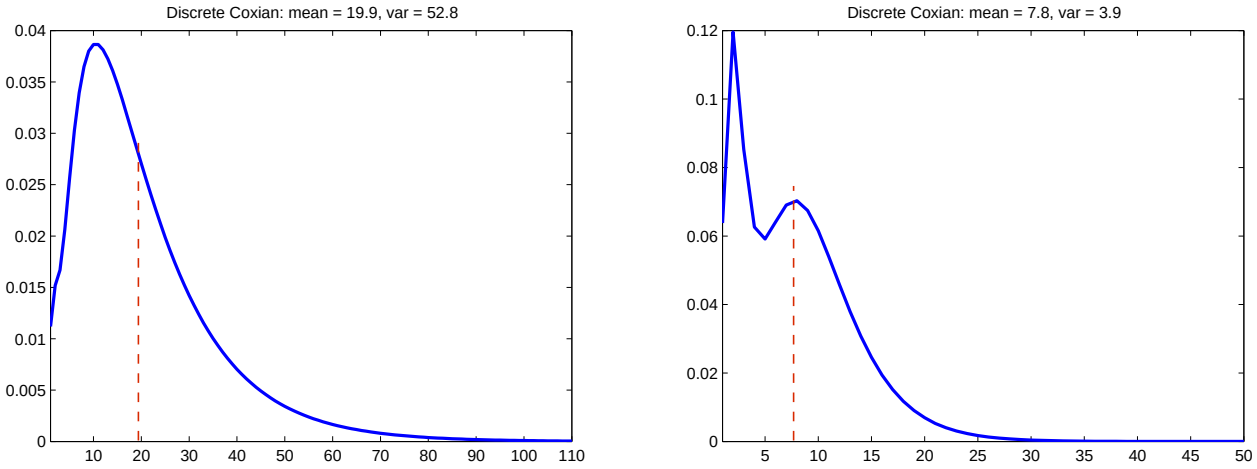
The variance:

$$\sigma_{X_n}^2 = E\left[X_n^2\right] - \bar{X}_n^2 = \frac{1-\lambda_n}{\lambda_n^2} \tag{4.16}$$

where $\mathbf{Li}_s(x) = \sum_{k=1}^{+\infty}\frac{x^k}{k^s}$ is the polylogarithm function valid for all s and $|x|<1$, and $\mathbf{Li}_{-1}(x) = \frac{x}{(1-x)^2}$, and $\mathbf{Li}_{-2}(x) = \frac{x(1+x)}{(1-x)^3}$.
Thus, the Coxian mean is given by:

$$\begin{aligned} \overline{\text{Cox}} &= E\left[\sum_{m=1}^{\mathcal{M}}\mu_m S_m\right] = \sum_{m=1}^{\mathcal{M}}\mu_m\bar{S}_m \\ &= \sum_{m=1}^{\mathcal{M}}\mu_m\sum_{n=m}^{\mathcal{M}}\bar{X}_n = \sum_{m=1}^{\mathcal{M}}\mu_m\sum_{n=m}^{\mathcal{M}}\frac{1}{\lambda_n} \end{aligned} \tag{4.17}$$

Since S_m are independent, their covariance is zero, thus the variance of the mixture is the sum of variances weighted by the squares of the original coefficients:



(a) (b)

Figure 4.5: Examples of Coxian distributions.
(a) $\boldsymbol{\mu} = [0.36\ 0.32\ 0.04\ 0.11\ 0.16]^\mathsf{T}, \boldsymbol{\lambda} = [0.18\ 0.64\ 0.43\ 0.62\ 0.07]^\mathsf{T}$
$\boldsymbol{\mu} = [0.32\ 0.31\ 0.01\ 0.25\ 0.11]^\mathsf{T}, \boldsymbol{\lambda} = [0.41\ 0.25\ 0.46\ 0.64\ 0.58]^\mathsf{T}$

$$\sigma^2_{\text{Cox}} = \sum_{m=1}^{\mathcal{M}} \text{var}\left(\mu_m S_m\right) = \sum_{m=1}^{\mathcal{M}} \mu_m^2 \sigma^2_{S_m}$$

The random variable S_m in turn consists of m independent random variable X_n (i.e. their covariances are zero) and thus:

$$\sigma^2_{\text{Cox}} = \sum_{m=1}^{\mathcal{M}} \mu_m^2 \sum_{n=m}^{\mathcal{M}} \sigma^2_{X_n} = \sum_{m=1}^{\mathcal{M}} \mu_m^2 \sum_{n=m}^{\mathcal{M}} \frac{1-\lambda_n}{\lambda_n^2} \tag{4.18}$$

Eqs. (4.17) and (4.18) show that the mean and variance are somewhat "proportional": a large mean would lead to a large variance (i.e. large μ means large μ^2; and large $\frac{1}{\lambda}$ results in small λ^2 and large $(1-\lambda)$, or equivalently large $\frac{1-\lambda}{\lambda^2}$) and otherwise. Fig. (4.5) shows examples of 5-phase Coxian density functions.

4.3 The Coxian Hidden semi-Markov Model

The discrete Coxian distribution is at first appealing to us as a suitable candidate for duration model in the hidden semi-Markov model due to: its denseness, making it a useful practical tool for approximating generic discrete distribution, and its simple

underlying MC, resulting in simple parameterization and possibly less computational load. In this section we formally present a novel stochastic model termed the discrete Coxian hidden semi-Markov model (CxHSMM). We start with a generic (discrete-time) HSMM (chapter 3) and describe how the Coxian distribution can be used to model state durations. We also present the methods for inference and parameter estimation by viewing the CxHSMM as a dynamic Bayesian network. To make our model applicable to real-life problems, we provide modified versions of inference and learning algorithms to deal with missing observations or labeled data. As our time domain is always discrete, henceforth, we omit the term discrete/discrete-time for simplicity.

4.3.1 Model definition

To recap from chapter 3, the HSMM is a generative model defined over a finite state space Q and an observation alphabet set V. The parameters of the HSMM include: the state initial probability $\pi : \ Q \mapsto [0, 1]$ specifying the starting state of the Markov chain defined over the states in Q; the transition matrix $A : \ Q^2 \mapsto [0, 1]$ governing the transitions within states ($A_{ii} = 0, \forall i \in Q$); the state duration distribution $D_i : \ \mathbf{N}^+ \mapsto [0, 1]$ defining the duration of state i in Q; and the observation model $B : \ Q \times V \mapsto [0, 1]$ determining the probability of generating an alphabet given a current state. We use a compact notation $\theta_{\text{HSMM}} \triangleq \{\pi, \ A, \ D, \ B\}$ to denote the set of parameters for the (flat) HSMM.

4.3.2 The Coxian Duration Model

For each state i in the state space Q, we define a discrete $\mathcal{M}$-phases Coxian distribution $D_i = \text{Cox}\left(\boldsymbol{\mu}^i, \boldsymbol{\lambda}^i\right)$ with the initial probabilities $\boldsymbol{\mu}^i = [\mu_1^i, \ldots, \mu_{\mathcal{M}}^i]^\mathsf{T}$ and transition probabilities $\boldsymbol{\lambda}^i = [\lambda_1^i, \ldots, \lambda_{\mathcal{M}}^i]^\mathsf{T}$ to model its duration. Note that the Coxian distributions associated with different states are independent but set to have the same number of phases. Once the Coxian of the current state goes to absorbing, the Markov chain transits to a new state and a new Coxian is initialized. We then term this HSMM as the *Coxian duration Hidden semi-Markov Model*, denoted as CxHSMM or $\mathcal{M}$-ph.CxHSMM when there is a specific phase number $\mathcal{M}$. Tab. (4.1) summarizes the $\mathcal{M}$-ph.CxHSMM parameter set $\theta_{\mathcal{M}\text{-ph.CxHSMM}}$. We note that when $\mathcal{M} = 1$ the model is equivalent to a HMM. Further, as a special case, if for all i, $\lambda_i = 1$ thus $\Pr(\text{duration of phase } i = d) = 1$ for $d = 1$, and $= 0$ for $d > 1$, we recover

Parameter	Dimension	Constraint	Meaning
π_i	$1 \times \lvert Q \rvert$	$\sum_{i \in Q} \pi_i = 1$	Initial probability of state i.
A_{ij}	$\lvert Q \rvert \times \lvert Q \rvert$	$\sum_{j \in Q} A_{ij} = 1$	Transition probability from state i to state j.
$D_i = \text{Cox}\left(\boldsymbol{\mu}^i, \boldsymbol{\lambda}^i\right)$			Coxian duration distribution for state i.
μ_n^i	$1 \times \mathcal{M}$	$\sum_{n=1}^{\mathcal{M}} \mu_n^i = 1$	Initial probability of phase n.
λ_n^i	$1 \times \mathcal{M}$	$0 < \lambda_n^i \leq 1$	Terminating probability of phase n.
$B_{v\lvert i}$	$\lvert Q \rvert \times \lvert V \rvert$	$\sum_{v \in V} B_{v\lvert i} = 1$	Probability of observing v given the current state i.

Table 4.1: Parameter sets $\theta_{\mathcal{M}\text{-ph.CxHSMM}}$.

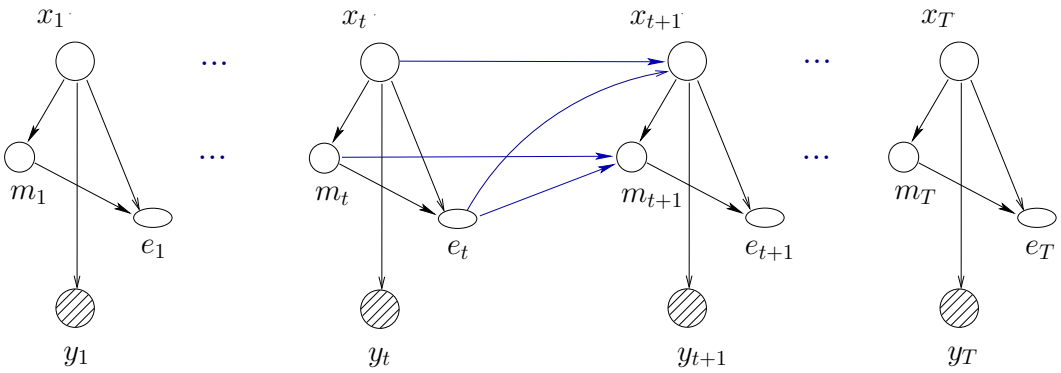


Figure 4.6: DBN representation of the CxHSMM.

the Multinomial distribution $\text{Mult}(\mu_1, \mu_2, \ldots, \mu_{\mathcal{M}})$. Finally, it is important to note that the number of free parameters for the Coxian duration model is $|Q|(2\mathcal{M} - 1)$ and is usually much smaller than $|Q|(M - 1)$ for the explicit duration model, where M can be potentially as large as T.

4.3.3 Dynamic Bayesian Network representation

This section constructs the Dynamic Bayesian Network (DBN) representation for the CxHSMM, states the network assumptions and details the mapping between DBN parameters and the model parameters. The CxHSMM's DBN representation is constructed in a similar manner to that of the Multinomial and exponential HSMM in chapter 3. However, there is a fundamental difference in the mechanism

controlling the duration of states which is no longer a simple "count" variable. Fig. (4.6) shows a DBN representation of the CxHSMM, in which shaded nodes are the observed variables while clear nodes are the hidden. At each time slice t, a set of variables $\mathcal{V}_t = \{x_t, m_t, e_t, y_t\}$ is maintained:

- x_t is the current state variable.
- m_t is an $\mathcal{M}$-valued variable representing the current phase of x_t.
- e_t is a Boolean-valued variable representing the ending status of x_t: $e_t = 1$ when m_t leaves the last phase (i.e. going to absorption), forcing x_t to terminate; otherwise $e_t = 0$.
- y_t is the observation returned by the system at time t.

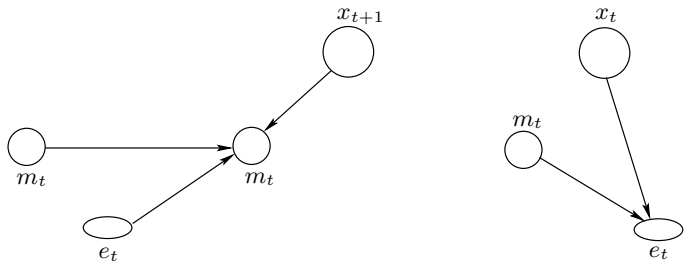
In general, $\{x_t, m_t, e_t\}$ are hidden and y_t is observed. In the setting of missing observation, y_t is replaced by empty set $\{\varnothing\}$; whereas in the presence of labeled data, e.g. x_t is observed, the observation set then includes the labels, e.g. the instantiations of x_t.

In this DBN representation, the *first* slice at time $t = 1$ is constructed as follows. Firstly, the state variable x_1 is initialized to an arbitrary state $i \in Q$ with a probability π_i. The variable m_1 is activated to a number $n \in [1, \mathcal{M}]$ drawn from the initial phase probability μ_n^i. If $m_1 < \mathcal{M}$, the ending variable e_1 is always set to 0; otherwise the link from x_1 to e_1 becomes active and e_1 is set to 0 with probability $1 - \lambda_{\mathcal{M}}^i$, or 1 with probability $\lambda_{\mathcal{M}}^i$. The observation y_1 is drawn from the emission matrix B.

After the first slice, the ending variable e_t for $t \geq 1$ specifies how the next time slice $t+1$ can be derived from the current time slice t given the model θ_{CxHSMM}. When $e_t = 0$, the same state x_t carries on to the next time slice, whereas when $e_t = 1$ (only when $m_t = \mathcal{M}$), the next state x_{t+1} is drawn from the transition matrix A:

$$\Pr\left(x_{t+1}^i \mid x_t^i, e_t^0\right) = 1$$
$$\Pr\left(x_{t+1}^j \mid x_t^i, e_t^1\right) = A_{ij}$$

In addition, the transitions of the phase variables m_t follow the parameters of the Coxian duration model as follows. When $e_t = 0$, we have $m_{t+1} \in \{m_t, m_t + 1\}$ and



(a) (b)

Figure 4.7: The two DBN cliques associated with the Coxian duration model in the CxHSMM.

the probability of staying in the same phase is :

$$\Pr\left(m_{t+1} = n \mid m_t = n, x_{t+1} = i, e_t = 0\right) = 1 - \lambda_n^i \quad \text{for } n < \mathcal{M} \tag{4.19}$$

$$\Pr\left(m_{t+1} = \mathcal{M} \mid m_t = \mathcal{M}, x_{t+1} = i, e_t = 0\right) = 1 \tag{4.20}$$

$$\Pr\left(e_t^0 \mid m_t^{\mathcal{M}}, x_t^i\right) = 1 - \lambda_{\mathcal{M}}^i \tag{4.21}$$

When $e_t = 1$, the starting phase of a new state is initialized:

$$\Pr\left(m_{t+1} = n \mid x_{t+1} = j, e_t = 1\right) = \mu_n^i \tag{4.22}$$

Eqs. (4.19)- (4.22) show that the Coxian parameters $\lambda_{1:\mathcal{M}-1}$ and $\mu_{1:\mathcal{M}}$ are associated with the conditional probability over the clique $\left\{m_{t+1} \mid m_t, e_t, x_{t+1}\right\}$ while the terminating probability of the last phase $\lambda_{\mathcal{M}}$ is attached to clique $\{e_t \mid m_t, x_t\}$. Thus, the Coxian duration model is fully defined over these two cliques (Fig. (4.7)) in the DBN representation.

The above analysis shows that the construction of this DBN imposes the following two restrictions:

$$\begin{aligned} m_t < \mathcal{M} \quad &\Longrightarrow \quad e_t = 0, \ \forall t \\ e_t = 0 \quad &\Longrightarrow \quad x_t = x_{t+1}, \ \forall t \end{aligned}$$

Finally, Tab. (4.2) shows the full set of the CxHSMM parameters θ_{CxHSMM} in section 4.3.1 mapped into their equivalent conditional probabilities in the DBN structure. Here again we adopt the notation s_t^i for the event $\{s_t = i\}$.

π_i	$=$	$\Pr\left(x_1^i\right)$
A_{ij}	$=$	$\Pr\left(x_{t+1}^j \mid x_t^i, e_t^1\right)$
D_i	$=$	$\mathrm{Cox}\left(\boldsymbol{\mu}^i, \boldsymbol{\lambda}^i\right)$
μ_n^i	$=$	$\Pr\left(m_{t+1}^n \mid x_{t+1}^i, e_t^1\right)$
$\lambda_{n<\mathcal{M}}^i$	$=$	$\Pr\left(m_{t+1}^{n+1} \mid m_t^n, x_{t+1}^i, e_t^0\right)$
$\lambda_{\mathcal{M}}^i$	$=$	$\Pr\left(e_t^1 \mid m_t^{\mathcal{M}}, x_t^i\right)$
$B_{v\mid i}$	$=$	$\Pr\left(y_t^v \mid x_t^i\right)$

Table 4.2: Mappings between the CxHSMM parameters and its the local conditional probabilities of its DBN representation.

4.3.4 Inference

Since the CxHSMM can be represented as a DBN, existing inference methods for DBNs can be readily applied. At time t, let $S_t \triangleq \{x_t, m_t, e_t\}$ be the amalgamated hidden state and its realization written in short as $\mathbf{s} \triangleq \{i, n, k\}$, then the CxHSMM can be viewed as a HMM with amalgamated hidden states $\{S_t\}$ and observations $\{y_t\}$, and inference task can be done similarly to that of the HMM (section 2.4.3). In particular, the familiar forward and backward procedures of the HMM can be used to compute the forward and backward variables of the CxHSMM:

$$\begin{aligned} &\text{forward variable:} && \alpha_t(i,n,k) = \Pr\left(x_t^i, m_t^n, e_t^k, y_{1:t}\right) \\ &\text{backward variable:} && \beta_t(i,n,k) = \Pr\left(y_{t+1:T} \mid x_t^i, m_t^n, e_t^k\right) \end{aligned}$$

From α and β, we then compute one- and two-slice smoothing distributions:

$$\begin{aligned} &\text{one-slice:} && \gamma_t(i,n,k) = \Pr\left(x_t^i, m_t^n, e_t^k \mid y_{1:T}\right) \\ &\text{two-slice:} && \xi_t(i,i',n,n',k,k') = \Pr\left(x_t^i, x_{t+1}^{i'}, m_t^n, m_{t+1}^{n'}, e_t^k, e_{t+1}^{k'} \mid y_{1:T}\right) \end{aligned}$$

which are required during EM training to compute the expected sufficient statistics for θ_{CxHSMM}.

4.3.4.1 The (scaled) Forward and Backward Variables

Using the expression in Eq. (2.34) of the HMM forward variable (section 2.4.3) and replacing the single hidden variable x_t^i by the amalgamated hidden variable $S_t^{\mathbf{s}} = \left\{x_t^i, m_t^n, e_t^k\right\}$, and x_{t+1}^j by $S_{t+1}^{\mathbf{s}'} = \left\{x_{t+1}^{i'}, m_{t+1}^{n'}, e_{t+1}^{k'}\right\}$, we can write the forward

variable for the CxHSMM as:

$$\begin{aligned}\alpha_{t+1}(\mathbf{s}') &= \Pr\left(S_{t+1}^{\mathbf{s}'}, y_{1:t+1}\right) \\ &= \sum_{\mathbf{s}} \Pr\left(y_{t+1} \mid S_{t+1}^{\mathbf{s}'}\right) \Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right) \Pr\left(S_t^{\mathbf{s}}, y_{1:t}\right) \end{aligned} \tag{4.23}$$

in which $\Pr\left(y_{t+1} \mid S_{t+1}^{\mathbf{s}'}\right) = \Pr\left(y_{t+1} \mid x_{t+1}^{i'}, m_{t+1}^{n'}, e_{t+1}^{k'}\right) = \Pr\left(y_{t+1} \mid x_{t+1}^{i'}\right) = B_{y_{t+1}|i'}$ is the emission probability, $\Pr\left(S_t^{\mathbf{s}}, y_{1:t}\right) = \alpha_t(\mathbf{s})$ is the forward variable associated with the previous time slice and the transition probability $\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right)$ is the product of three local conditional probabilities:

$$\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right) = \Pr\left(e_{t+1}^{k'} \mid x_{t+1}^{i'}, m_{t+1}^{n'}\right) \Pr\left(m_{t+1}^{n'} \mid x_{t+1}^{i'}, m_t^n, e_t^k\right) \Pr\left(x_{t+1}^{i'} \mid x_t^i, e_t^k\right) \tag{4.24}$$

Using the conditional independencies in the CxHSMM cliques, the individual transition probabilities are in turn given by:

$$\Pr\left(e_{t+1}^{k'} \mid x_{t+1}^{i'}, m_{t+1}^{n'}\right) = \begin{cases} \delta_{k'}^{(0)}\left(1-\lambda_{i'}^{\mathcal{M}}\right) + \delta_{k'}^{(1)}\lambda_{i'}^{\mathcal{M}}, & n' = \mathcal{M} \\ \delta_{k'}^{(0)}, & n' < \mathcal{M} \end{cases} \tag{4.25}$$

$$\Pr\left(m_{t+1}^{n'} \mid x_{t+1}^{i'}, m_t^n, e_t^k\right) = \begin{cases} \mu_{i'}^{n'}, & k=1,\ n=\mathcal{M} \\ \delta_{n'}^{(\mathcal{M})}, & k=0,\ n=\mathcal{M} \\ \delta_{n'}^{(n)}\left(1-\lambda_{i'}^{n}\right) + \bar{\delta}_{n'}^{(n)}\lambda_{i'}^{n}, & k=0,\ n<\mathcal{M} \end{cases} \tag{4.26}$$

$$\Pr\left(x_{t+1}^{i'} \mid x_t^i, e_t^k\right) = \begin{cases} \delta_{i'}^{(i)}, & k=0 \\ A_{ii'}, & k=1 \end{cases} \tag{4.27}$$

where the usual notation $\delta_a^{(b)} = 1$ only if $a = b$, and $= 0$ otherwise; and the new notation $\bar{\delta}_a^{(b)}$ shows the opposite, i.e. $\bar{\delta}_a^{(b)} = 1$ only if $a \neq b$, and $= 0$ otherwise. From Eqs. (4.24) to (4.27), the transition probability from slice t to $t+1$ is:

$$\begin{aligned}\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right) = &\left[\delta_{k'}^{(0)}\delta_{n'}^{(\mathcal{M})}\left(1-\lambda_{i'}^{\mathcal{M}}\right) + \delta_{k'}^{(1)}\delta_{n'}^{(\mathcal{M})}\lambda_{i'}^{\mathcal{M}} + \delta_{k'}^{(0)}\bar{\delta}_{n'}^{(\mathcal{M})}\right] \\ &\times \left[\delta_k^{(0)}\delta_n^{(\mathcal{M})}\delta_{n'}^{(\mathcal{M})}\delta_{i'}^{(i)} + \delta_k^{(0)}\bar{\delta}_n^{(\mathcal{M})}\delta_{n'}^{(n)}\left(1-\lambda_{i'}^{n}\right)\delta_{i'}^{(i)}\right. \\ &\left. + \delta_k^{(0)}\bar{\delta}_n^{(\mathcal{M})}\bar{\delta}_{n'}^{(n)}\lambda_{i'}^{n}\delta_{i'}^{(i)} + \delta_k^{(1)}\delta_n^{(\mathcal{M})}\mu_{i'}^{n'}\ A_{ii'}\right] \end{aligned} \tag{4.28}$$

Now the recursive forward variable can be written as:

$$\alpha_{t+1}(\mathbf{s}' = \{i', n', k'\}) = B_{y_{t+1}|i'} \sum_{\mathbf{s}} \Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right) \alpha_t(\mathbf{s}) \tag{4.29}$$

with the transition $\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right)$ given in Eq. (4.28). The forward variable starts at $t = 1$ with:

$$\begin{aligned}\tilde{\alpha}_1(i,n,k) &= \Pr\left(x_1^i, m_1^n, e_1^k, y_1\right) \\ &= \Pr(y_1 \mid i)\Pr\left(e_1^k \mid x_1^i, m_1^n\right)\Pr\left(m_1^n \mid x_1^i\right)\Pr\left(x_1^i\right) \\ &= B_{y_1|i}\left[\delta_k^{(0)}\delta_n^{(\mathcal{M})}\left(1-\lambda_i^{\mathcal{M}}\right) + \delta_k^{(1)}\delta_n^{(\mathcal{M})}\lambda_i^{\mathcal{M}} + \delta_k^{(0)}\bar{\delta}_{\mathcal{M}}^{(n)}\right]\mu_i^n\pi_i \end{aligned} \tag{4.30}$$

Similarly, the backward calculation is given in the same form as that of the HMM in Eq. (2.38) (section 2.4.3) with the single hidden state $\{x\}$ again being replaced by the amalgamated hidden state $\{x, m, e\}$:

$$\begin{aligned}\beta_t(\mathbf{s}) &= \Pr\left(y_{t+1:T} \mid S_t^{\mathbf{s}}\right) \\ &= \sum_{\mathbf{s}'}\Pr\left(y_{t+1} \mid S_{t+1}^{\mathbf{s}'}\right)\Pr\left(y_{t+2:T} \mid S_{t+1}^{\mathbf{s}'}\right)\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right) \\ &= \sum_{\mathbf{s}'} B_{y_{t+1}|i'}\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right)\beta_{t+1}(\mathbf{s}') \end{aligned} \tag{4.31}$$

where the transition probability $\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right)$ is again defined by Eq. (4.28). The initialization at $t = T$ is given by:

$$\beta_T(i,n,k) = \Pr\left(y_{T+1:T} \mid x_T^i, m_T^n, e_T^k\right) = \Pr\left(\varnothing \mid x_T^i, m_T^n, e_T^k\right) = 1 \tag{4.32}$$

In practice, we usually have to deal with long observation sequences and thus the calculation of α_t will encounter the numerical underflow problem since it will be a joint probability of a large number of variables when t becomes very large. To avoid this problem we use a s*caling scheme* similar to the HMM (section 2.4.3). Instead of calculating $\alpha_t(s)$, we calculate a *scaled* version $\tilde{\alpha}_t(\mathbf{s})$:

$$\tilde{\alpha}_t(\mathbf{s}) \triangleq \frac{\alpha_t(\mathbf{s})}{\Pr(y_{1:t})} = \Pr\left(S_t^{\mathbf{s}} \mid y_{1:t}\right) \tag{4.33}$$

Calculation of $\tilde{\alpha}_t(s)$ can be performed efficiently via dynamic programming. To simplify the task, we write:

$$\tilde{\alpha}_t(s) = \ddot{\alpha}_t(s)/\psi_t \tag{4.34}$$

where $\ddot{\alpha}_t(s) = \Pr(S_t^s, y_t \mid y_{1:t-1})$ is a partially scaled version of $\alpha_t(s)$, and $\psi_t = \Pr(y_t \mid y_{1:t-1})$ is the scaling factor. Given $\tilde{\alpha}_t(s)$, the recursion at time $t+1$ is

computed as:

$$\begin{aligned}\ddot{\alpha}_{t+1}(\mathbf{s}') &= \sum_{\mathbf{s}} \Pr\left(S_t^{\mathbf{s}}, S_{t+1}^{\mathbf{s}'}, y_{t+1} \mid y_{1:t}\right) \\ &= B_{y_{t+1}|i'} \sum_{\mathbf{s}} \Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right) \tilde{\alpha}_t(\mathbf{s}) \\ \psi_{t+1} &= \Pr\left(y_{t+1} \mid y_{1:t}\right) = \sum_{\mathbf{s}'} \ddot{\alpha}_{t+1}(\mathbf{s}')\end{aligned} \tag{4.35}$$

where the transition probability $\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right)$ is again given in Eq. (4.28). At time $t = 1$,

$$\tilde{\alpha}_1(i,n,k) = \Pr\left(x_1^i, m_1^n, e_1^k \mid y_1\right) = \frac{\Pr\left(x_1^i, m_1^n, e_1^k, y_1\right)}{\Pr(y_1)} = \frac{\alpha_1(i,n,k)}{\sum_{i,n,k} \alpha_1(i,n,k)}$$

with $\alpha_1(i,n,k)$ specified in Eq. (4.30).

The backward variable is scaled by a factor $\phi_t \triangleq \Pr\left(y_{t+1:T} \mid y_{1:t}\right)$ as follows:

$$\tilde{\beta}_t(\mathbf{s}) = \frac{\beta_t(\mathbf{s})}{\phi_t} \tag{4.36}$$

with scaled factor ϕ_t computed recursively as:

$$\begin{aligned}\phi_t &= \Pr\left(y_{t+1:T} \mid y_{1:t}\right) \\ &= \Pr\left(y_{t+2:T} \mid y_{1:t+1}\right) \Pr\left(y_{t+1} \mid y_{1:t}\right) \\ &= \phi_{t+1}\psi_{t+1}\end{aligned} \tag{4.37}$$

and

$$\phi_T = \Pr\left(y_{T+1:T} \mid y_{1:T}\right) = \Pr(\varnothing \mid y_{1:T}) = 1$$

Thus, the scaled backward variable is initialized at $t = T$ with $\tilde{\beta}_T(i,n,k) = {}^{\beta_T(i,n,k)}/_{\phi_T} = 1$.

Next, following the derivations from Eq. (2.47) to Eq. (2.54) of the HMM (section 2.4.3), the one- and two-time slice smoothing distributions for the CxHSMM is obtained as:

$$\gamma_t(\mathbf{s}) \triangleq \Pr\left(S_t^{\mathbf{s}} \mid y_{1:T}\right) = \tilde{\alpha}_t(\mathbf{s})\,\tilde{\beta}_t(\mathbf{s}) \tag{4.38}$$

$$\xi_t(\mathbf{s}, \mathbf{s}') \triangleq \Pr\left(S_t^{\mathbf{s}}, S_{t+1}^{\mathbf{s}'} \mid y_{1:T}\right) = \frac{\tilde{\alpha}_t(\mathbf{s})\,\tilde{\beta}_{t+1}(\mathbf{s}') B_{y_{t+1}|i'} \Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right)}{\psi_{t+1}} \tag{4.39}$$

with the transition probability $\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right)$ provided in Eq. (4.28).

Finally, the above analysis shows that the CxHSMM requires an inference complexity of $O(|Q|^2\mathcal{M}^2T)$, or $O(|Q|^2\mathcal{M}^2)$ for each filtering step. However, since within a given state, the phase variables are constrained so that $m_{t+1} \in \{m_t, m_t + 1\}$, the full joint probability of m_t and m_{t+1} can be represented in just $O(\mathcal{M})$ space instead of $O(\mathcal{M}^2)$. This reduces the overall complexity to $O(|Q|^2\mathcal{M}T)$ (or $O(|Q|^2\mathcal{M})$ per filtering step). We note that if the duration is modeled as a Multinomial distribution or an Exponential Family distribution, the complexity is $O(|Q|^2MT)$ with M being the maximum duration length. For $\mathcal{M} \ll M$ we have clearly achieved a significant speedup and at the same time avoided the problem of determining M in advance.

4.3.4.2 Inference with missing observations or observed states

The CxHSMM normally consists of hidden states $\{x_{1:T}, m_{1:T}, e_{1:T}\}$ and observations $\{y_{1:T}\}$ as shown by the clear and shaded variables in its DBN representation (Fig. (4.6)). However, in practice we sometimes encounter missing data due to various reasons such as tracking errors, and thus the need to perform inference in the presence of missing observation. For convenience let $\{g_1, \ldots, g_T\}$ be the observation set, and $g_t = \{\varnothing\}$ if there is a missing observation at time t, otherwise g_t is set to the observation, e.g. $g_t = \{y_t\}$. Another common practical situation is that the labels of some normally hidden states could be supplied (semi-supervised learning), for example via the use of various sensors. Let $\bar{x}_t$, $\bar{m}_t$, and $\bar{e}_t$ be the instantiations of x_t, m_t, and e_t, respectively. For instance, if the state variable is observed at time t together with the emission symbol, then $g_t = \{\bar{x}_t, y_t\}$; or if the end-node e_t is also observed, then $g_t = \{\bar{x}_t, \bar{e}_t, y_t\}$. Our inference algorithm also needs to be modified to tackle the issues.

Similar to inference in the normal context (section 4.3.4.1), let $S_t = \{x_t, m_t, e_t\}$ be the amalgamated state and $\mathbf{s} = \{i, n, k\}$ be its realization, then the following

auxiliary variables are required to be computed at each time t:

$$\begin{array}{lll}
\text{Scaled forward variable:} & \tilde{\alpha}_t(\mathbf{s}) & = \Pr\left(S_t^{\mathbf{s}} \mid g_{1:t}\right) \\
\text{Scaling factor:} & \psi_t & = \Pr\left(g_t \mid g_{1:t-1}\right) \\
\text{Scaled backward variable:} & \tilde{\beta}_t(\mathbf{s}) & = \dfrac{\Pr\left(g_{t+1:T} \mid S_t^{\mathbf{s}}\right)}{\Pr\left(g_{t+1:T} \mid g_{1:t}\right)} = \dfrac{\Pr\left(g_{t+1:T} \mid S_t^{\mathbf{s}}\right)}{\prod_{\tau=t+1}^{T} \psi_\tau} \\
\text{1-time slice smoothing dist.:} & \gamma_t(\mathbf{s}) & = \Pr\left(S_t^{\mathbf{s}} \mid g_{1:T}\right) \\
\text{2-time slice smoothing dist.:} & \xi_t(\mathbf{s}, \mathbf{s}') & = \Pr\left(S_t^{\mathbf{s}}, S_{t+1}^{\mathbf{s}'} \mid g_{1:T}\right)
\end{array}$$

We start with the recursion of the partially labeled scaled forward variable $\ddot{\alpha}_t(\mathbf{s})$:

$$\begin{aligned}
\ddot{\alpha}_t(\mathbf{s}) &= \Pr\left(S_t^{\mathbf{s}}, g_t \mid g_{1:t-1}\right) \\
&= \sum_{\mathbf{s}'} \Pr\left(S_{t-1}^{\mathbf{s}'}, S_t^{\mathbf{s}}, g_t \mid g_{1:t-1}\right) \\
&= \sum_{\mathbf{s}'} \Pr\left(g_t \mid S_t^{\mathbf{s}}\right) \Pr\left(S_t^{\mathbf{s}} \mid S_{t-1}^{\mathbf{s}'}\right) \Pr\left(S_{t-1}^{\mathbf{s}'} \mid g_{1:t-1}\right) \\
&= \Pr\left(g_t \mid S_t^{\mathbf{s}}\right) \sum_{\mathbf{s}'} \Pr\left(S_t^{\mathbf{s}} \mid S_{t-1}^{\mathbf{s}'}\right) \tilde{\alpha}_{t-1}(\mathbf{s}')
\end{aligned} \tag{4.40}$$

The only term in Eq. (4.40) requiring special treatment is $\Pr\left(g_t \mid S_t^{\mathbf{s}}\right)$ since it is the only one containing an observation at time t. The observation g_t must be consistent with the amalgamated state $S_t^{\mathbf{s}} = \left\{x_t^i, m_t^n, e_t^k\right\}$, otherwise $\Pr\left(g_t \mid S_t^{\mathbf{s}}\right) = 0$. In particular,

$$\Pr\left(g_t \mid S_t^{\mathbf{s}} = \left\{x_t^i, m_t^n, e_t^k\right\}\right) = \begin{cases}
B_{y_t|i}, & g_t = \{y_t\} \quad \text{(observing } y_t\text{)} \\
\delta_{\bar{x}_t}^{(i)}, & g_t = \{\bar{x}_t\} \quad \text{(observing } x_t = \bar{x}_t\text{)} \\
\delta_{\bar{m}_t}^{(n)}, & g_t = \{\bar{m}_t\} \quad \text{(observing } m_t = \bar{m}_t\text{)} \\
\delta_{\bar{e}_t}^{(k)}, & g_t = \{\bar{e}_t\} \quad \text{(observing } e_t = \bar{e}_t\text{)} \\
1, & g_t = \{\varnothing\} \quad \text{(missing observation)}
\end{cases} \tag{4.41}$$

or in short,

$$\Pr\left(g_t \mid S_t^{\mathbf{s}}\right) = \left(B_{y_t|i}\right)^{h(y_t \subseteq g_t)} \left(\delta_{\bar{x}_t}^{(i)}\right)^{h(\bar{x}_t \subseteq g_t)} \left(\delta_{\bar{m}_t}^{(n)}\right)^{h(\bar{m}_t \subseteq g_t)} \left(\delta_{\bar{e}_t}^{(k)}\right)^{h(\bar{e}_t \subseteq g_t)} \tag{4.42}$$

where $h(z) = 1$ if statement z is true; otherwise $= 0$. Thus, the probability of observed given hidden $\Pr\left(g_t \mid S_t^{\mathbf{s}}\right)$ is set to the emission probability if *only* the emission symbol is observed; otherwise, it is multiplied by an identity function of the observed state for consistency, or simply set to 1 when the observation is missing.

Given the partially scaled forward variable, the scaling factor is then computed as normal:

$$\psi_t = \Pr\left(g_t \mid g_{1:t-1}\right) = \sum_{\mathbf{s}} \Pr\left(S_t^{\mathbf{s}}, g_t \mid g_{1:t-1}\right) = \sum_{\mathbf{s}} \ddot{\alpha}_t\left(\mathbf{s}\right) \tag{4.43}$$

The expression in Eq. (4.43) is always valid for any cases of g_t with $\ddot{\alpha}_t\left(\mathbf{s}\right)$ computed as in Eq. (4.40), and it is straightforward to see that in case nothing is observed $g_t = \{\varnothing\}$, the scaling factor is simply equal to 1: $\psi_t = \Pr\left(\varnothing \mid g_{1:t-1}\right) = 1$.
The scaled forward variable then follows as:

$$\tilde{\alpha}_t\left(\mathbf{s}\right) \;=\; \Pr\left(S_t^{\mathbf{s}} \mid g_{1:t}\right) = \frac{\ddot{\alpha}_t\left(\mathbf{s}\right)}{\psi_t}$$

which can be further expressed as:

$$\begin{aligned}\tilde{\alpha}_t\left(\mathbf{s}\right) &= \frac{\ddot{\alpha}_t\left(\mathbf{s}\right)}{\sum_{\mathbf{s}} \ddot{\alpha}_t\left(\mathbf{s}\right)} \\ &= \frac{\Pr\left(g_t \mid S_t^{\mathbf{s}}\right) \sum_{\mathbf{s}'} \Pr\left(S_t^{\mathbf{s}} \mid S_{t-1}^{\mathbf{s}'}\right) \tilde{\alpha}_{t-1}\left(\mathbf{s}'\right)}{\sum_{\mathbf{s}} \Pr\left(g_t \mid S_t^{\mathbf{s}}\right) \sum_{\mathbf{s}'} \Pr\left(S_t^{\mathbf{s}} \mid S_{t-1}^{\mathbf{s}'}\right) \tilde{\alpha}_{t-1}\left(\mathbf{s}'\right)}\end{aligned} \tag{4.44}$$

with $\Pr\left(g_t \mid S_t^{\mathbf{s}}\right)$ given in Eq. (4.42). Our next step is to compute the scaled backward variable:

$$\begin{aligned}\tilde{\beta}_{t-1}(\mathbf{s}') &= \frac{\Pr\left(g_{t:T} \mid S_{t-1}^{\mathbf{s}'}\right)}{\prod_{\tau=t}^{T} \psi_\tau} \\ &= \frac{1}{\prod_{\tau=t}^{T} \psi_\tau} \sum_{\mathbf{s}} \Pr\left(S_t^{\mathbf{s}}, g_{t:T} \mid S_{t-1}^{\mathbf{s}'}\right) \\ &= \frac{1}{\psi_t} \sum_{\mathbf{s}} \frac{\Pr\left(g_{t+1:T} \mid S_t^{\mathbf{s}}\right)}{\prod_{\tau=t+1}^{T} \psi_\tau} \Pr\left(g_t \mid S_t^{\mathbf{s}}\right) \Pr\left(S_t^{\mathbf{s}} \mid S_{t-1}^{\mathbf{s}'}\right) \\ &= \frac{1}{\psi_t} \sum_{\mathbf{s}} \Pr\left(g_t \mid S_t^{\mathbf{s}}\right) \Pr\left(S_t^{\mathbf{s}} \mid S_{t-1}^{\mathbf{s}'}\right) \tilde{\beta}_t(\mathbf{s})\end{aligned}$$

again with $\Pr\left(g_t \mid S_t^{\mathbf{s}}\right)$ defined as in Eq. (4.42).

Given the forward and backward variables, the one-time slice smoothing distributions is then computed as follows:

$$\begin{aligned}\gamma_t\left(\mathbf{s}\right) &\triangleq \Pr\left(S_t^{\mathbf{s}} \mid g_{1:T}\right) \\ &= \frac{\Pr\left(g_{t+1:T} \mid S_t^{\mathbf{s}}\right) \Pr\left(S_t^{\mathbf{s}} \mid g_{1:t}\right)}{\Pr\left(g_{t+1:T} \mid g_{1:t}\right)} \\ &= \tilde{\beta}_t(\mathbf{s}) \tilde{\alpha}_t\left(\mathbf{s}\right)\end{aligned} \tag{4.45}$$

where the need for consistency between g_t and $S_t^{\mathbf{s}}$ is required in the computation of $\tilde{\alpha}_t(\mathbf{s}) = \Pr(S_t^{\mathbf{s}} \mid g_{1:t})$ (Eq. (4.44)), which in turn is taken care of in the probability $\Pr(g_t \mid S_t^{\mathbf{s}})$ as shown in Eq. (4.42). Next, the two-time slice smoothing distribution is given by:

$$\begin{aligned}
\xi_{t-1}(\mathbf{s}, \mathbf{s}') &\triangleq \Pr\left(S_{t-1}^{\mathbf{s}}, S_t^{\mathbf{s}'} \mid g_{1:T}\right) \\
&= \frac{\Pr\left(g_{t+1:T} \mid S_t^{\mathbf{s}'}\right) \Pr\left(g_t \mid S_t^{\mathbf{s}'}\right) \Pr\left(S_t^{\mathbf{s}'} \mid S_{t-1}^{\mathbf{s}}\right) \Pr\left(S_{t-1}^{\mathbf{s}'} \mid g_{1:t-1}\right)}{\Pr(g_{t:T} \mid g_{1:t-1})} \\
&= \frac{\Pr\left(g_t \mid S_t^{\mathbf{s}'}\right) \Pr\left(S_t^{\mathbf{s}'} \mid S_{t-1}^{\mathbf{s}}\right) \Pr\left(S_{t-1}^{\mathbf{s}} \mid g_{1:t-1}\right) \Pr\left(g_{t+1:T} \mid S_t^{\mathbf{s}'}\right)}{\Pr(g_t \mid g_{1:t-1}) \Pr(g_{t+1:T} \mid g_{1:t})} \\
&= \frac{1}{\psi_t} \Pr\left(g_t \mid S_t^{\mathbf{s}'}\right) \Pr\left(S_t^{\mathbf{s}'} \mid S_{t-1}^{\mathbf{s}}\right) \tilde{\alpha}_{t-1}(\mathbf{s}) \tilde{\beta}_t(\mathbf{s}')
\end{aligned} \tag{4.46}$$

where the only term involving both g_t and S_t is $\Pr\left(g_t \mid S_t^{\mathbf{s}'}\right)$, and is already given in Eq. (4.42).

In short, the definitions of auxiliary variables and the forward/backward procedures remain almost the same as in normal context (section 4.3.4.1), except that the conditional probability $\Pr(g_t \mid S_t)$ has to maintain consistency between the observations and the states. This leads to the following results: (a) if there is missing observation the emission probability $B_{v|i}$ is removed during calculation, and (b) if any states are observed, identity functions are multiplied into the emission probability. These additional steps ensure the one- and two-time slice smoothing distributions are computed consistently.

4.3.5 Learning

This section presents parameter learning in the CxHSMM. Since the CxHSMM can be represented as a DBN, it can be written in the Exponential Family form, and consequently the maximum likelihood (ML) estimation derived for the Exponential Family distributions (sections 2.2.1 and 2.2.2.2) can be used here. We first show the ML estimation when the model is fully observed before investigating the Expectation-Maximization (EM) algorithm for the general case where the states $\{x_{1:T}, m_{1:T}, e_{1:T}\}$ are hidden and the emissions $\{y_{1:T}\}$ are observed. To make the model applicable in real world applications, we also present learning in the presence of missing observations and partially labeled data.

4.3.5.1 Maximum Likelihood for fully observed CxHSMM

When all the states are observed, following the derivations in section 2.2.1, the ML solutions are set to the normalized sufficient statistics, which are the count of configurations. Let $T\left(\theta^i_{k,v}\right)$ be the sufficient statistic of parameter $\theta^i_{k,v}$, which is mapped to the local conditional probability $\Pr\left(X_{i_t} = k \mid X_{\pi_{i_t}} = v\right)$ at time slice t of the respective DBN. Then,

$$T\left(\theta^i_{k,v}\right) = \sum_{t=1}^{T} \delta^{(k)}_{X_{i_t}} \delta^{(v)}_{X_{\pi_{i_t}}} \tag{4.47}$$

$T\left(\theta^i_{k,v}\right)$ is the total number of configurations $\left\{X_{i_t} = k, X_{\pi_{i_t}} = v\right\}$ present in the DBN. This equation allows us to obtain sufficient statistics for θ_{CxHSMM} directly from inspecting its DBN.

For example, the sufficient statistic $T\left(\mu^i_n\right)$ of the parameter μ^i_n, mapped to $\Pr\left(m^n_{t+1} \mid x^i_{t+1}, e^1_t\right)$, is the the total count of configurations $\left\{m^n_{t+1}, x^i_{t+1}, e^1_t\right\}$ over time, which is also the number of times the Coxian associated with state i is initiated in phase n, or equivalently the total count of configurations $\left\{m^n_{t+1}, x^i_{t+1}, e^1_t\right\}$:

$$T\left(\mu^i_n\right) = \sum_{t=1}^{T-1} \delta^{(n)}_{m_{t+1}} \delta^{(i)}_{x_{t+1}} \delta^{(1)}_{e_t}$$

Thus, estimation for the Coxian phase initial probabilities is given by:

$$\hat{\mu}^i_n = \frac{T\left(\mu^i_n\right)}{\sum_n T\left(\mu^i_n\right)} = \frac{\sum_{t=1}^{T-1} \delta^{(n)}_{m_{t+1}} \delta^{(i)}_{x_{t+1}} \delta^{(1)}_{e_t}}{\sum_{n=1}^{\mathcal{M}} \sum_{t=1}^{T-1} \delta^{(n)}_{m_{t+1}} \delta^{(i)}_{x_{t+1}} \delta^{(1)}_{e_t}} = \frac{\sum_{t=1}^{T-1} \delta^{(n)}_{m_{t+1}} \delta^{(i)}_{x_{t+1}} \delta^{(1)}_{e_t}}{\sum_{t=1}^{T-1} \delta^{(i)}_{x_{t+1}} \delta^{(1)}_{e_t}}$$

Another example is the ML solution for the phase transition parameter λ^i_n. For any phase $n < \mathcal{M}$, the sufficient statistic for λ^i_n is counted every time the Coxian leaves phase n for phase $n+1$, which is in the configuration $\left\{m^{n+1}_{t+1}, m^n_t, x^i_{t+1}, e^0_t\right\}$. Hence, for $n < \mathcal{M}$:

$$\begin{aligned}
\hat{\lambda}^i_n &= \frac{T\left(\lambda^i_n\right)}{\sum_{m_{1:T}} T\left(\lambda^i_n\right)} \\
&= \frac{\sum_{t=1}^{T-1} \delta^{(n+1)}_{m_{t+1}} \delta^{(n)}_{m_t} \delta^{(i)}_{x_{t+1}} \delta^{(0)}_{e_t}}{\sum_{n'=n:n+1} \sum_{t=1}^{T-1} \delta^{(n')}_{m_{t+1}} \delta^{(n)}_{m_t} \delta^{(i)}_{x_{t+1}} \delta^{(0)}_{e_t}} \\
&= \frac{\sum_{t=1}^{T-1} \delta^{(n+1)}_{m_{t+1}} \delta^{(n)}_{m_t} \delta^{(i)}_{x_{t+1}} \delta^{(0)}_{e_t}}{\sum_{t=1}^{T-1} \delta^{(n)}_{m_t} \delta^{(i)}_{x_{t+1}} \delta^{(0)}_{e_t}}, \qquad \text{for } n < \mathcal{M}
\end{aligned}$$

The last phase parameter $\lambda^i_{\mathcal{M}}$ is mapped into the DBN clique $\left\{e^1_t \mid m^{\mathcal{M}}_t, x^i_t\right\}$, therefore, its sufficient statistic is counted on the configuration $\left\{e^1_t, m^{\mathcal{M}}_t, x^i_t\right\}$:

$$\begin{aligned}\hat{\lambda}^i_{\mathcal{M}} &= \frac{\langle T\left(\lambda^i_{\mathcal{M}}\right)\rangle}{\sum_{e_{1:T}}\langle T\left(\lambda^i_{\mathcal{M}}\right)\rangle} \\ &= \frac{\sum_{t=1}^{T}\delta^{(1)}_{e_t}\delta^{(\mathcal{M})}_{m_t}\delta^{(i)}_{x_t}}{\sum_{k=0:1}\sum_{t=1}^{T}\delta^{(k)}_{e_t}\delta^{(\mathcal{M})}_{m_t}\delta^{(i)}_{x_t}} = \frac{\sum_{t=1}^{T}\delta^{(1)}_{e_t}\delta^{(\mathcal{M})}_{m_t}\delta^{(i)}_{x_t}}{\sum_{t=1}^{T}\delta^{(\mathcal{M})}_{m_t}\delta^{(i)}_{x_t}}\end{aligned}$$

Finally, ML solutions for the remaining parameters are:

$$\hat{\pi}_i = \frac{T\left(\pi_i\right)}{\sum_i T\left(\pi_i\right)} = \frac{\delta^{(i)}_{x_1}}{\sum_i \delta^{(i)}_{x_1}} = \delta^{(i)}_{x_1} \tag{4.48}$$

$$\hat{A}_{ij} = \frac{T\left(A_{ij}\right)}{\sum_j T\left(A_{ij}\right)} = \frac{\sum_{t=1}^{T-1}\delta^{(j)}_{x_{t+1}}\delta^{(i)}_{x_t}\delta^{(1)}_{e_t}}{\sum_j\sum_{t=1}^{T-1}\delta^{(j)}_{x_{t+1}}\delta^{(i)}_{x_t}\delta^{(1)}_{e_t}} = \frac{\sum_{t=1}^{T-1}\delta^{(j)}_{x_{t+1}}\delta^{(i)}_{x_t}\delta^{(1)}_{e_t}}{\sum_{t=1}^{T-1}\delta^{(i)}_{x_t}\delta^{(1)}_{e_t}} \tag{4.49}$$

$$\hat{B}_{v|i} = \frac{T\left(B_{v|i}\right)}{\sum_v T\left(B_{v|i}\right)} = \frac{\sum_{t=1}^{T}\delta^{(v)}_{y_t}\delta^{(i)}_{x_t}}{\sum_v\sum_{t=1}^{T}\delta^{(v)}_{y_t}\delta^{(i)}_{x_t}} = \frac{\sum_{t=1}^{T}\delta^{(v)}_{y_t}\delta^{(i)}_{x_t}}{\sum_{t=1}^{T}\delta^{(i)}_{x_t}}$$

4.3.5.2 Expectation-Maximization for the CxHSMM

In general, the states $\{x_{1:T}, m_{1:T}, e_{1:T}\}$ are hidden and the emissions $\{y_{1:T}\}$ are observed. Thus, the Expectation-Maximization (EM) algorithm is employed for the parameter estimation tasks. Following Eq. (2.29) (section 2.2.2.2), we compute the expected complete log likelihood for the CxHSMM over the distribution $\Pr\left(S_{1:T} \mid y_{1:T}, \theta_F\right)$ as:

$$\begin{aligned}\left\langle \mathcal{L}^C\left(S_{t:T}, y_{1:T} \mid \theta_F\right)\right\rangle = &\sum_i \langle T\left(\pi_i\right)\rangle \log\pi_i + \sum_{i,j}\left\langle T\left(A_{ij}\right)\right\rangle \log A_{ij} \\ &+ \sum_{i,n}\left\langle T\left(\mu^i_n\right)\right\rangle \log\mu^i_n + \sum_{i,n}\left\langle T\left(\lambda^i_{n<\mathcal{M}}\right)\right\rangle \log\lambda^i_{n<\mathcal{M}} \\ &+ \sum_i \left\langle T\left(\lambda^i_{\mathcal{M}}\right)\right\rangle \log\lambda^i_{\mathcal{M}} + \sum_{v,i}\left\langle T\left(B_{v|i}\right)\right\rangle \log B_{v|i}\end{aligned}$$

in which the expected sufficient statistics (ESS's) are computed by following the results in Eqs. (2.20) and (2.29) which states that the ESS for any parameter $\theta_{k,v} = \Pr\left(X = k \mid X_\pi = v\right)$ defined as:

$$\begin{aligned}\langle T\left(\theta_{k,v}\right)\rangle &= \sum_{\mathbf{H}} T\left(\theta_{k,v}\right)\Pr\left(\mathbf{H} \mid \mathbf{O}, \theta\right) \\ &= \sum_{\mathbf{H}}\sum_{k,v}\delta^{(k)}_X\delta^{(v)}_{X_\pi}\Pr\left(\mathbf{H} \mid \mathbf{O}, \theta\right)\end{aligned} \tag{4.50}$$

In particular, let us first look at the phase initial probability μ_n^i in detail. The sufficient statistic of μ_n^i is collected every time the system enters phase n right after a transition to state i, which means it is counted over the clique $\left\{m_{t+1}^n \mid x_{t+1}^i, e_t^1\right\}$, and thus:

$$T\left(\mu_n^i\right) = \sum_{t=0}^{T-1} \delta_{m_{t+1}}^{(n)} \delta_{x_{t+1}}^{(i)} \delta_{e_t}^{(1)}$$

in which we assume e_t at time $t = 0$ is 1 by default. Taking the expectation of $T\left(\mu_n^i\right)$ over the probability of hidden variables over observed ones $\Pr(x_{1:T}, m_{1:T}, e_{1:T} \mid y_{1:T})$ results in:

$$\begin{aligned}\left\langle T\left(\mu_n^i\right)\right\rangle &= \sum_{x_{1:T}, m_{1:T}, e_{1:T}} \sum_{t=0}^{T-1} \delta_{m_{t+1}}^{(n)} \delta_{x_{t+1}}^{(i)} \delta_{e_t}^{(1)} \Pr(x_{1:T}, m_{1:T}, e_{1:T} \mid y_{1:T}) \\ &= \sum_{t=0}^{T-1} \Pr\left(x_{t+1}^i, m_{t+1}^n, e_t^1 \mid y_{1:T}\right)\end{aligned}$$

which is easily obtained by marginalizing the one- and two-time slice smoothing distributions:

$$\begin{aligned}\left\langle T\left(\mu_n^i\right)\right\rangle &= \Pr\left(x_1^i, m_1^n \mid y_{1:T}\right) + \sum_{t=1}^{T-1} \Pr\left(x_{t+1}^i, m_{t+1}^n, e_t^1 \mid y_{1:T}\right) \\ &= \sum_k \gamma_1(i, n, k) + \sum_{i', n', k} \xi_t(i', i, n', n, 1, k)\end{aligned}$$

Following the results of theorem 2.1, the re-estimated formula for μ_n^i is then given by:

$$\begin{aligned}\hat{\mu}_n^i &= \frac{\left\langle T\left(\mu_n^i\right)\right\rangle}{\sum_n \left\langle T\left(\mu_n^i\right)\right\rangle} \\ &= \frac{\sum_k \gamma_1(i, n, k) + \sum_{i', n', k} \xi_t(i', i, n', n, 1, k)}{\sum_{n,k} \gamma_1(i, n, k) + \sum_{i', n', n, k} \xi_t(i', i, n', n, 1, k)}\end{aligned} \tag{4.51}$$

The phase terminating probabilities λ_n^i need to be treated with more care as they are defined differently, as shown in Tab. (4.2), for $n < \mathcal{M}$ and $n = \mathcal{M}$. For $n < \mathcal{M}$, the sufficient statistic $T(\lambda_n^i)$ is counted every time the phase n is terminated within the given state i, or the number of configurations $\left\{m_{t+1}^{n+1} \mid m_t^n, x_{t+1}^i, e_t^0\right\}$:

$$T(\lambda_n^i) = \sum_{t=1}^{T-1} \delta_{m_{t+1}}^{(n+1)} \delta_{m_t}^{(n)} \delta_{x_{t+1}}^{(i)} \delta_{e_t}^{(0)}, \qquad \text{for } n < \mathcal{M}$$

The ESS then follows as:

$$\begin{aligned}\left\langle T\left(\lambda_n^i\right)\right\rangle &= \sum_{x_{1:T},m_{1:T},e_{1:T}} \sum_{t=1}^{T-1} \delta_{m_{t+1}}^{(n+1)} \delta_{m_t}^{(n)} \delta_{x_{t+1}}^{(i)} \delta_{e_t}^{(0)} \Pr(x_{1:T}, m_{1:T}, e_{1:T} \mid y_{1:T}) \\ &= \sum_{t=1}^{T-1} \Pr(m_{t+1}^{n+1}, m_t^n, x_{t+1}^i, e_t^0 \mid y_{1:T}) \qquad (4.52) \\ &= \sum_{t=1}^{T-1} \sum_{i',k} \xi_t\left(i', i, n, n+1, 0, k\right), \qquad \text{for } n < \mathcal{M} \qquad (4.53)\end{aligned}$$

The normalization factor (nFactor) in the ML solution for $\lambda_{n<\mathcal{M}}^i$ is obtained by marginalizing out all possible values of the phase m_{t+1} from the ESS in Eq. (4.52):

$$\begin{aligned}\text{nFactor}_{\lambda_{n<\mathcal{M}}^i} &= \sum_{t=1}^{T-1} \sum_{m_{t+1}} \Pr(m_{t+1}, m_t^n, x_{t+1}^i, e_t^0 \mid y_{1:T}) \\ &= \sum_{t=1}^{T-1} \Pr(m_t^n, x_{t+1}^i, e_t^0 \mid y_{1:T}) \\ &= \sum_{t=1}^{T-1} \sum_{i',n',k} \xi_t\left(i', i, n, n', 0, k\right) \qquad (4.54)\end{aligned}$$

From Eqs. (4.53) and (4.54), the ML solution is then given by:

$$\begin{aligned}\hat{\lambda}_n^i &= \frac{\left\langle T\left(\lambda_n^i\right)\right\rangle}{\text{nFactor}_{\lambda_{n<\mathcal{M}}^i}} \\ &= \frac{\sum_{t=1}^{T-1} \sum_{i',k} \xi_t\left(i', i, n, n+1, 0, k\right)}{\sum_{t=1}^{T-1} \sum_{i',n',k} \xi_t\left(i', i, n, n', 0, k\right)}, \qquad \text{for } n < \mathcal{M} \qquad (4.55)\end{aligned}$$

For $n = \mathcal{M}$, the terminating probability of the last phase $\lambda_{\mathcal{M}}^i$ becomes the probability that the state i has finished its duration, which is associated with the configuration $\left\{e_t^1 \mid m_t^{\mathcal{M}}, x_t^i\right\}$. Therefore, by using the same counting and expectation procedures, we obtain:

$$\begin{aligned}\left\langle T\left(\lambda_{\mathcal{M}}^i\right)\right\rangle &= E\left[\delta_{e_t}^{(1)} \delta_{m_t}^{(\mathcal{M})} \delta_{x_t}^{(i)}\right]_{\Pr\left(x_{1:T}, m_{1:T}, e_{1:T} \mid y_{1:T}\right)} \\ &= \sum_{t=1}^{T} \Pr(e_t^1, m_t^{\mathcal{M}}, x_t^i \mid y_{1:T}) \qquad (4.56) \\ &= \sum_{t=1}^{T} \gamma_t\left(i, \mathcal{M}, 1\right)\end{aligned}$$

The normalization factor now is equivalent to the probability that the Coxian is at its last phase (and the state i has or has not finished its duration). Thus, from Eq. (4.56) it follows that:

$$\begin{aligned}\text{nFactor}_{\lambda^i_{\mathcal{M}}} &= \sum_{t=1}^{T}\left[\Pr(e_t^1, m_t^{\mathcal{M}}, x_t^i \mid y_{1:T}) + \Pr(e_t^0, m_t^{\mathcal{M}}, x_t^i \mid y_{1:T})\right] \\ &= \sum_{t=1}^{T}\Pr(m_t^{\mathcal{M}}, x_t^i \mid y_{1:T}) \\ &= \sum_{t=1}^{T}\sum_{k}\gamma_t\left(i, \mathcal{M}, k\right)\end{aligned}$$

Hence,

$$\hat{\lambda}^i_{\mathcal{M}} = \frac{\langle T\left(\lambda^i_{\mathcal{M}}\right)\rangle}{\text{nFactor}_{\lambda^i_{\mathcal{M}}}} = \frac{\sum_{t=1}^{T}\gamma_t\left(i, \mathcal{M}, 1\right)}{\sum_{t=1}^{T}\sum_{k}\gamma_t\left(i, \mathcal{M}, k\right)} \tag{4.57}$$

The ML solutions for the remaining parameters can easily be obtained by first computing the ESS (via Eq. (4.50)) and then using Lagrange multiplier (theorem 2.1), and thus briefly listed below:

$$\hat{\pi}_i = \frac{\langle T\left(\pi_i\right)\rangle}{\sum_i \langle T\left(\pi_i\right)\rangle} = \frac{\Pr\left(x_1^i \mid y_{1:T}\right)}{\sum_i \Pr\left(x_1^i \mid y_{1:T}\right)} = \sum_{n,k}\gamma_1\left(i, n, k\right) \tag{4.58}$$

$$\begin{aligned}\hat{A}_{ij} &= \frac{\left\langle T\left(A_{ij}\right)\right\rangle}{\sum_j \left\langle T\left(A_{ij}\right)\right\rangle} = \frac{\sum_{t=1}^{T-1}\Pr\left(x_{t+1}^j, x_t^i, e_t^1 \mid y_{1:T}\right)}{\sum_j \sum_{t=1}^{T-1}\Pr\left(x_{t+1}^j, x_t^i, e_t^1 \mid y_{1:T}\right)} \\ &= \frac{\sum_{t=1}^{T-1}\sum_{n,n',k'}\xi_t\left(i, j, n, n', 1, k\right)}{\sum_{t=1}^{T-1}\sum_{j,n,n',1,k}\xi_t\left(i, j, n, n', 1, k\right)}\end{aligned} \tag{4.59}$$

$$\begin{aligned}\hat{B}_{v|i} &= \frac{\left\langle T\left(B_{v|i}\right)\right\rangle}{\sum_v \left\langle T\left(B_{v|i}\right)\right\rangle} = \frac{\sum_{t=1}^{T}\Pr\left(x_t^i \mid y_{1:T}\right)\delta_{y_t}^{(v)}}{\sum_v \sum_{t=1}^{T}\Pr\left(x_t^i \mid y_{1:T}\right)\delta_{y_t}^{(v)}} \\ &= \frac{\sum_{t=1}^{T}\sum_{n,k}\gamma_t\left(i, n, k\right)\delta_{y_t}^{(v)}}{\sum_{t=1}^{T}\sum_{n,k}\gamma_t\left(i, n, k\right)}\end{aligned} \tag{4.60}$$

4.3.5.3 Learning with missing observations or observed states

In the presence of missing observations, the ML solutions for $\hat{\pi}_i$, $\hat{A}_{ij}$, $\hat{\mu}^i_n$, $\hat{\lambda}^i_n$ in Eqs. (4.58), (4.59), (4.51), (4.55), and (4.57) in the previous section are all valid provided that the two smoothing distributions $\gamma_t\left(\mathbf{s}\right)$ and $\xi_t\left(\mathbf{s}, \mathbf{s}'\right)$ used are computed with

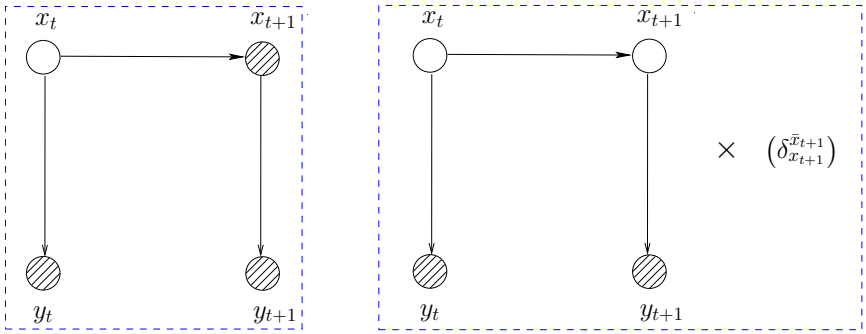


Figure 4.8: The two equivalent structures.

missing observations as in Eqs. (4.45) and (4.46) (section 4.3.4.2). However, there is a small change to the re-estimation formula for the emission probability as its sufficient statistic involves probability of y_t in the count of configuration $\{y_t \mid x_t\}$:

$$\begin{aligned}\left\langle T\left(B_{v|i}\right)\right\rangle &= \sum_{x_{1:T},m_{1:T},e_{1:T}} \sum_{t=1}^{T} \delta_{y_t}^{(v)} \delta_{x_t}^{(i)} \Pr\left(x_{1:T}, m_{1:T}, e_{1:T} \mid g_{1:T}\right) \\ &= \begin{cases} \sum_{t=1}^{T} \Pr\left(x_t^i \mid g_{1:T}\right) \delta_{y_t}^{(v)}, & g_t = y_t \\ \sum_{\tau=\{1:T\}\setminus t} \Pr\left(x_\tau^i \mid g_{1:T}\right), & g_t = \varnothing \quad \text{(missing obs.)} \end{cases}\end{aligned}$$

On the other hand, if some of the hidden states in the set $\{x_{1:T}, m_{1:T}, e_{1:T}\}$ are observed, then these observations have already been taken care of during the computation of the smoothing distributions $\gamma_t(\mathbf{s})$ and $\xi_t(\mathbf{s}, \mathbf{s}')$ (section 4.3.4.2) by multiplication with appropriate identity functions (Eq. (4.42)). For example, the structure in Fig. 4.8(a) is equivalent to the structure in Fig. 4.8(b) multiplied by a suitable identity function. In the E-step, the ESS's are computed directly from the smoothing distributions, thus, the effect of observed states is then embedded into the ESS's. For example, given x_τ is observed, e.g. $g_\tau = \{y_\tau, \bar{x}_\tau\}$, the ESS for the transition parameter A_{ij} would be calculated as follows:

$$\begin{aligned}\left\langle T\left(A_{ij}\right)\right\rangle &= \sum_{S_{1:T}} \sum_{t=1}^{T-1} \delta_{x_{t+1}}^{(j)} \delta_{x_t}^{(i)} e_t^1 \Pr\left(S_t \mid g_{1:T} = \{y_{1:T}, \bar{x}_\tau\}\right) \\ &= \sum_{t \in \mathcal{C}} \Pr\left(x_{t+1}^j, x_t^i, e_t^1 \mid y_{1:T}\right) + \Pr\left(x_\tau^j, x_{\tau-1}^i, e_{\tau-1}^1 \mid g_{1:T}\right) \\ &\quad + \Pr\left(x_{\tau+1}^j, x_\tau^i, e_\tau^1 \mid g_{1:T}\right) \end{aligned} \tag{4.61}$$

where the set $\mathcal{C} = [1 : \tau - 2, \tau + 1 : T]$. The first term in Eq. (4.61) is computed normally by marginalizing over the two-time slice smoothing distribution, and the effect of having x_τ observed will be shown in the probabilities $\Pr\left(g_\tau = \{y_\tau, \bar{x}_\tau\} \mid x_\tau^i, m_\tau, e_\tau^1\right)$ and $\Pr\left(g_\tau = \{y_\tau, \bar{x}_\tau\} \mid x_\tau^j, m_\tau^0, e_\tau\right)$ that will arise when we compute the last two terms. In particular, for $t \in [1 : \tau - 2, \tau + 1 : T]$, we have:

$$\sum_{t \in \mathcal{C}} \Pr\left(x_{t+1}^j, x_t^i, e_t^1 \mid y_{1:T}\right) = \sum_{t \in \mathcal{C}} \sum_{n,n',k} \xi_\tau\left(\{i, n, 1\}, \{j, n', k\}\right) \tag{4.62}$$

Substituting $\xi_\tau(.)$ in Eq. (4.39) with $\mathbf{s} = \{i, n, 1\}$ and $\mathbf{s}' = \{j, n', k\}$ into Eq. (4.62) results in:

$$\sum_{t \in \mathcal{C}} \Pr\left(x_{t+1}^j, x_t^i, e_t^1 \mid y_{1:T}\right) = \sum_{t \in \mathcal{C}} \sum_{n,n',k} \frac{B_{y_{t+1}|j} \Pr\left(S_{t+1}^{\{j,n',k\}} \mid S_t^{\{i,n,1\}}\right) \tilde{\alpha}_t(i, n, 1)\, \tilde{\beta}_{t+1}(j, n', k)}{\psi_{t+1}} \tag{4.63}$$

Next, the second term in Eq. (4.61) is given by:

$$\begin{aligned}
\Pr\left(x_\tau^j, x_{\tau-1}^i, e_{\tau-1}^1 \mid g_{1:T}\right) &= \sum_{n,n',k} \xi_{\tau-1}\left(\{i, n, 1\}, \{j, n', k\}\right) \\
&\overset{(a)}{=} \sum_{n,n',k} \Big[\Pr\left(g_\tau = \{y_\tau, \bar{x}_\tau\} \mid S_\tau^{\{j,n',k\}}\right) \Pr\left(S_\tau^{\{j,n',k\}} \mid S_{\tau-1}^{\{i,n,1\}}\right) \\
&\qquad \times \frac{\tilde{\alpha}_{\tau-1}(i, n, 1)\, \tilde{\beta}_\tau(j, n', k)}{\psi_\tau}\Big] \\
&\overset{(b)}{=} \sum_{n,n',k} \frac{B_{y_\tau|j} \delta_{\bar{x}_\tau}^{(j)} \Pr\left(S_\tau^{\{j,n',k\}} \mid S_{\tau-1}^{\{i,n,1\}}\right) \tilde{\alpha}_{\tau-1}(i, n, 1)\, \tilde{\beta}_\tau(j, n', k)}{\psi_\tau}
\end{aligned}$$

where steps (a) and (b) are obtained by following the result of Eqs. (4.46) and (4.42), respectively, and $\Pr\left(g_\tau = \{y_\tau, \bar{x}_\tau\} \mid S_\tau^{\{j,n',k\}}\right) = B_{y_\tau|j} \delta_{\bar{x}_\tau}^{(j)}$ is the only term required to be consistent with the observation of $\bar{x}_\tau$.
The third term in Eq. (4.61) is then computed as:

$$\begin{aligned}
\Pr\left(x_{\tau+1}^j, x_\tau^i, e_\tau^1 \mid g_{1:T}\right) &= \sum_{n,n',k} \xi_\tau\left(\{i, n, 1\}, \{j, n', k\}\right) \\
&= \sum_{n,n',k} \frac{B_{y_{\tau+1}|j} \Pr\left(S_{\tau+1}^{\{j,n',k\}} \mid S_\tau^{\{i,n,1\}}\right) \tilde{\alpha}_\tau(i, n, 1)\, \tilde{\beta}_{\tau+1}(j, n', k)}{\psi_{\tau+1}}
\end{aligned}$$

in which the consistency between the observation $g_\tau = \{y_\tau, \bar{x}_\tau\}$ and the state x_τ^i arises during the computation of $\tilde{\alpha}_\tau(i,n,1)$. Following Eq. (4.44), we have:

$$\begin{aligned}\tilde{\alpha}_\tau(i,n,1) &= \frac{\Pr\left(g_\tau \mid S_\tau^{\{i,n,1\}}\right)\sum_{\mathbf{s}'}\Pr\left(S_\tau^{\{i,n,1\}} \mid S_{\tau-1}^{\mathbf{s}'}\right)\tilde{\alpha}_{\tau-1}(\mathbf{s}')}{\sum_{\mathbf{s}}\Pr(g_\tau \mid S_\tau^{\mathbf{s}})\sum_{\mathbf{s}'}\Pr\left(S_\tau^{\mathbf{s}} \mid S_{\tau-1}^{\mathbf{s}'}\right)\tilde{\alpha}_{\tau-1}(\mathbf{s}')} \\ &= \frac{B_{y_\tau|i}\delta_{\bar{x}_\tau}^{(i)}\sum_{\mathbf{s}'}\Pr\left(S_\tau^{\{i,n,1\}} \mid S_{\tau-1}^{\mathbf{s}'}\right)\tilde{\alpha}_{\tau-1}(\mathbf{s}')}{\sum_{\mathbf{s}}B_{y_\tau|i}\delta_{\bar{x}_\tau}^{(i)}\sum_{\mathbf{s}'}\Pr\left(S_\tau^{\mathbf{s}} \mid S_{\tau-1}^{\mathbf{s}'}\right)\tilde{\alpha}_{\tau-1}(\mathbf{s}')}\end{aligned}$$

Next, given the ESS $\left\langle T\left(A_{ij}\right)\right\rangle$, the M-step follows as normal (theorem 2.1) and leads to the ML solution:

$$\hat{A}_{ij} = \frac{\left\langle T\left(A_{ij}\right)\right\rangle}{\sum_j \left\langle T\left(A_{ij}\right)\right\rangle}$$

In short, the inference step has ensured the consistency over observation of states. Hence, the set of ML solutions for θ_{CxHSMM} in Eqs. (4.58), (4.59), (4.60), (4.51), (4.55), and (4.57) remain valid provided that the smoothing distributions used in them are computed as shown by Eqs. (4.45) and (4.46).

4.4 Applications with the CxHSMM: recognition of activities of the same category

Apart from its other strengths, the Coxian is also appealing to activity recognition due to its structural advantage of having cascaded *phases*, which intuitively relate to sequences of *sub-activities* in an activity. In this section we apply the novel CxHSMM to the problem of recognizing ADLs and compare it with other existing semi-Markov models and the standard HMM. We argue that there are several common categories of ADLs in the house such as: "cooking meal", " washing dishes", " ironing clothes", " leisure reading", etc. Activities of the same category generally follow the same standard procedures. For example, "cooking meal" includes: "*taking food from fridge*" → "*washing veggies/cutting meat*" → "*seasoning food*" → "*cooking*"; or "ironing clothes" would consist of: "*bringing clothes to laundry*" → "*taking out the iron*" → "*setting up the iron board*" → "*ironing*" → "*tidying up*" →"*putting clothes away*". In other words, each activity category has its own sequential order of tasks needed to be fulfilled. However, the sub-activities within a given category may have different durations. For example, time spent at the stove for "cooking lunch" would be less than that for "cooking dinner", or time spent at

the laundry for "ironing a shirt" on weekday mornings would be much less than for "ironing the whole set of clothes" at weekends. The challenging problem is how to *learn* and *distinguish* ADLs of the same category mainly based on the differences in the *durations* of their sub-activities.

This section investigates how effectively different duration models (the Coxian, the popular Multinomial and the Exponential Family) at modeling and recognizing activities of the same category, in particular the three different routines (activities) of meal preparation and consumption (category). Thus, the experimental models include the CxHSMM with the number of phases of the Coxian ranging from 2 to 7, the non-parametric Multinomial duration HSMM (MuHSMM), representatives of the Exponential Family HSMMs: the Poisson duration HSMM (PsHSMM) and the Inverse Gaussian duration HSMM (IgHSMM), and the baseline HMM.

4.4.1 Data and environment descriptions

We collect a total of 48 sequences for the three following activities in the meal preparation and consumption category:

- (**a.1**) "tea – cake – newspaper breakfast"
- (**a.2**) "scrambled egg on toast lunch"
- (**a.3**) "lasagna – salad lunch"

We consider the extreme case in which the three activities have exactly the same sequential order of sub-activities but differ in the durations of these tasks. This is also the hardest scenario since the differences are in duration patterns and not in trajectories, making our task of activity classification more challenging. Fig. (4.9) shows the twelve fixed sequential steps: (1) "*take food from fridge*" → (2) "*bring food to stove*" → (3) "*wash veggies/fill water at sink*" → (*4*) "*come back to stove for cooking*" → (5) "*take plates/cup from cupboard*" → (*6*) "*get food from stove*" → (7) "*bring food to table*" → (8) "*take drink from fridge*" → (9) "*have meal at table*" → (10) "*clean stove*" → (11) "*wash dishes at sink*" → (12) "*leave the kitchen*". Tab. (4.3) shows the statistics of typical durations spent at special landmarks (fridge, stove, sink, cupboard, and table) for the three activities. For example, $15 - 17$(s) is the duration spent at the stove for cooking scrambled eggs on toast, which is generally

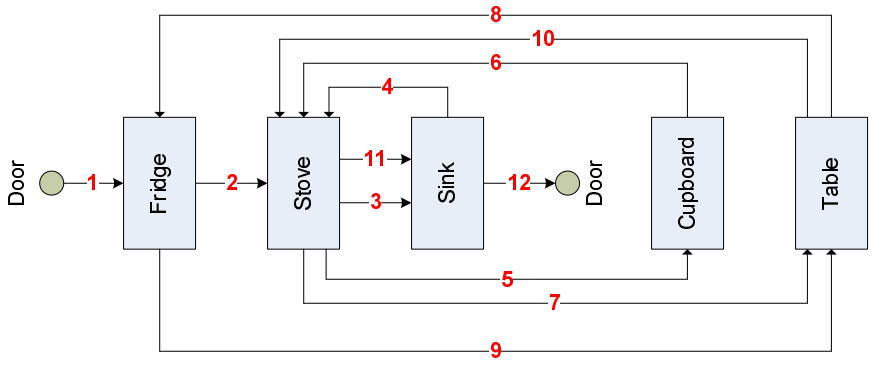


Figure 4.9: Sequential orders of visits.

	Fridge		Stove				Sink		Cupb	Table	
(a.1)	1–2	4–6	1–2	1–2	7–9	1–2	2–4	8–10	8–10	1–2	28–32
(a.2)	6–8	1–2	8–10	15–17	4–6	8–10	6–8	18–20	1–2	3–4	14–16
(a.3)	10–12	1–2	4–6	8–10	2–4	3–5	12–14	12–14	1–2	3–4	19–21

Table 4.3: Typical durations spent (in seconds) at the landmarks obtained from empirical data.

longer than for reheating the lasagna $(8-10(\text{s}))$, or making a cup of tea $(7-9(\text{s}))$; having breakfast while reading the morning newspaper, $28-32(\text{s})$, usually requires more time at the table than simply having lunch alone, $14-16(\text{s})$ or $19-21(\text{s})$. In addition, Tab. (4.3) shows that each landmark may have multiple durations: the first column shows the duration of the first visit, the second column is the duration of the second visit, etc. For example, for activity (**a.1**), the occupant first stops at the fridge for 1-2(s) to check out milk and cake, then later returns to the fridge for 4-6(s) to take out milk and cake; whereas in activity (**a.2**), the occupant stops at the fridge the first time for 6-8(s) to take out food and then re-visits the fridge afterwards for 1-2(s) to get a drink. In this experiment we have covered the possibility that an occupant may visit some landmarks several times within an activity, and different activities may sometimes share the same typical duration at the same place.

The environment is a kitchen set-up as shown in Fig. (4.10). The scene is captured by two cameras mounted at two opposite ceiling corners and a multiple-camera tracking module is used to detect movements and returns the list of positions in x-y coordinates visited by the occupant. For modeling convenience the kitchen is quantized into 28 square cells of 1m^2, and the x-y readings are then converted into

the cell numbers. The low-level vision tracking module employed in this work is the same as that of [Nguyen, 2004a] and detailed in appendix A. This tracking module however occasionally loses track of the occupant due to occlusions, or when the actor stays still for too long and is confused with the background. That leads to about a third of the captured sequences having approximately 7% entries being missed on the average. Therefore, we will experiment with two sets of data: the first set $\mathcal{A}$ consists of the originally captured sequences with *missing observations*, and to further test the robustness of our model, we construct a second set of data $\mathcal{B}$ in which a missing entry is interpolated by its neighbors.

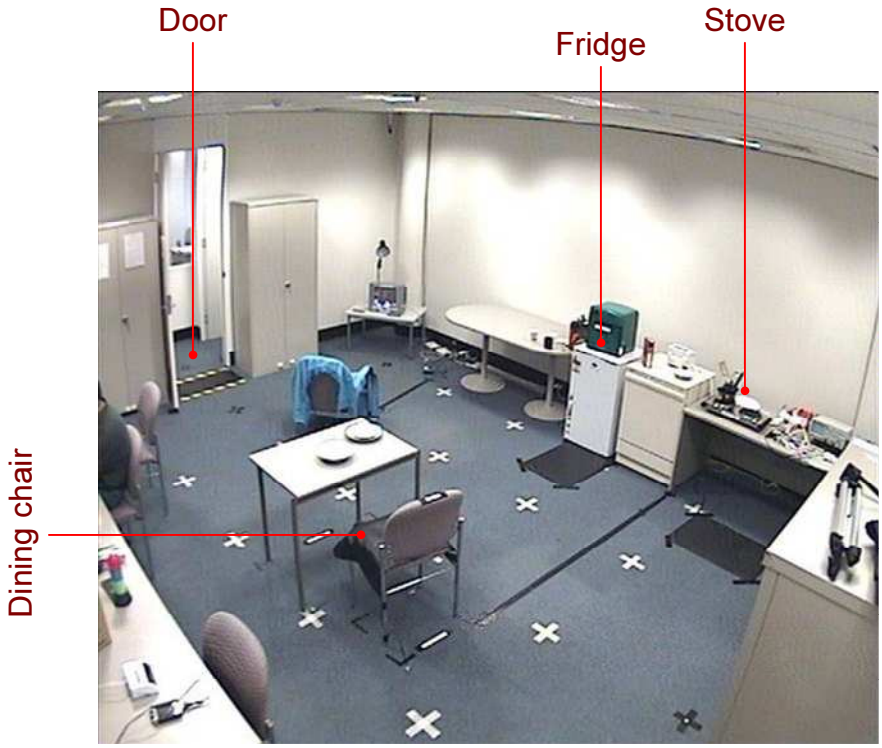
4.4.2 Training and testing strategy

To ensure an objective result, we employ a *leave-one-out* strategy for training and testing. We sequentially pick out one sequence Y from the dataset D for testing, and use the remainder $\{D \setminus Y\}$ for training. In the trained models (various $\mathcal{M}$-ph.CxHSMMs, a MuHSMM, a PsHSMM, an IgHSMM and a HMM), we let the number of states $|Q| = 28$, equal to the number of quantized cells in the kitchen environment (Fig. (4.10)). For the MuHSMM, the PsHSMM, and the IgHSMM, we equate the maximum duration M to the maximum activity length ($\sim 100 - 120$(s)). All these models have a fixed observation model B obtained offline from the characteristics of the tracking model, and all other parameters are randomly initialized.

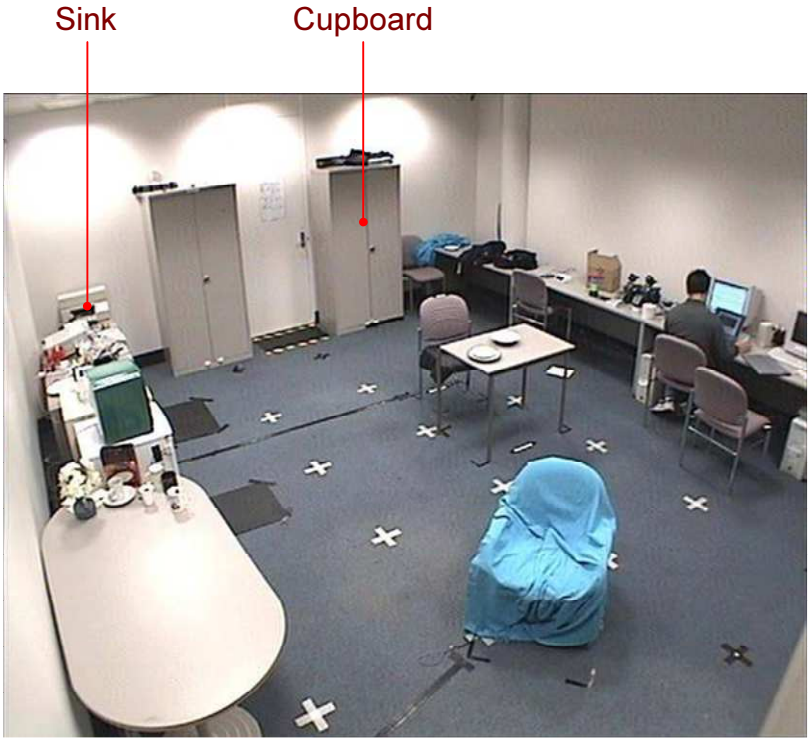
The model performance is tested on three criteria including *classification accuracy*, *early detection rate* and *running time* defined as follows.

Definition 4.3. Given N different stochastic models $\{\theta_1, \ldots, \theta_N\}$, each trained on a class of activities, and an observation sequence $y_{1:T}$, the likelihood $\Pr(y_{1:t} \mid \theta_n)$ is computed at each time $t \in [1, T]$ and used to label the most likely activity class. The *online recognition accuracy* at any time t is the ratio of activities correctly labeled at time t to the total activities tested. The ***classification accuracy*** is the online recognition accuracy at time $t = T$. The ***early detection rate*** *(EDR)* is the ratio t_0/T where t_0 is the earliest time from which the activity label remains accurate. The ***running time*** is simply the time required to run one EM iteration during training.

■



(a) camera1



(b) camera 2.

Figure 4.10: The kitchen environment viewed from two cameras.

Model selection on different $\mathcal{M}$-ph.CxHSMMs: When modeling the state duration by a Coxian distribution we have to choose the best number of phases. The key is to balance the complexity of the model and its degree of fitness to the data. For the $\mathcal{M}$-ph.CxHSMM, we train six different variants by varying $\mathcal{M}$ from 2 to 7 (note that for $\mathcal{M} = 1$, the CxHSMM reduces to a HMM). We measure the model performance in terms of *classification accuracy* and *early detection rate*, and *running time* on *unseen* data to select the most suitable $\mathcal{M}$.

4.4.3 Experiments with missing observation dataset $\mathcal{A}$

This section experiments with dataset $\mathcal{A}$, which has some sequences containing missing observations. In both the learning and recognition phases, missing entries in the observation vectors are treated as hidden (theory in sections 4.3.4.2 and 4.3.5.3).

First, we look at how the different models have learned the state durations. Fig. (4.11) shows the duration spent at the table in activity (**a.3**) learned by the PsHSMM, the IgHSMM, the MuHSMM, and the 5-ph.CxHSMM. While the Poisson fails to learn accurate duration information, the rest, especially the Coxian and the Multinomial, have captured, relatively well, the mixture of two typical durations: 3-4(s), and 19-21(s) (the statistics of empirical durations is shown in Tab. (4.3)). The Coxian has not fully separated the two peaks but successfully smoothed the spikes in the durations in comparison with the Multinomial.

In addition, Fig. (4.12) shows the log likelihood learned from the dataset of activity (**a.3**) by all the models. Due to their simplicity the HMM and the PsHSMM quickly converge after about 4 iterations; while the IgHSMM requires about 20 iterations, both the MuHSMM and 5-ph.CxHSMM take up to 25 iterations. However, the learning time is well worth it as they deliver much higher performance as discussed below.

Tab. (4.4)(c) shows that the HMM performs worst with only 68% classification accuracy due to its incapacity to model non-geometric durations. The PsHSMM performance is almost equally poor (69% accuracy), possibly because the Poisson is not flexible enough (i.e. having only one parameter) to model complicated state occupancies. The IgHSMM (76% accuracy) performs almost comparably to the 2-ph.CxHSMM (78% accuracy), but is completely surpassed by any $\mathcal{M} \geq$ 3-ph.CxHSMMs. The disadvantage of the Inverse Gaussian model is possibly be-

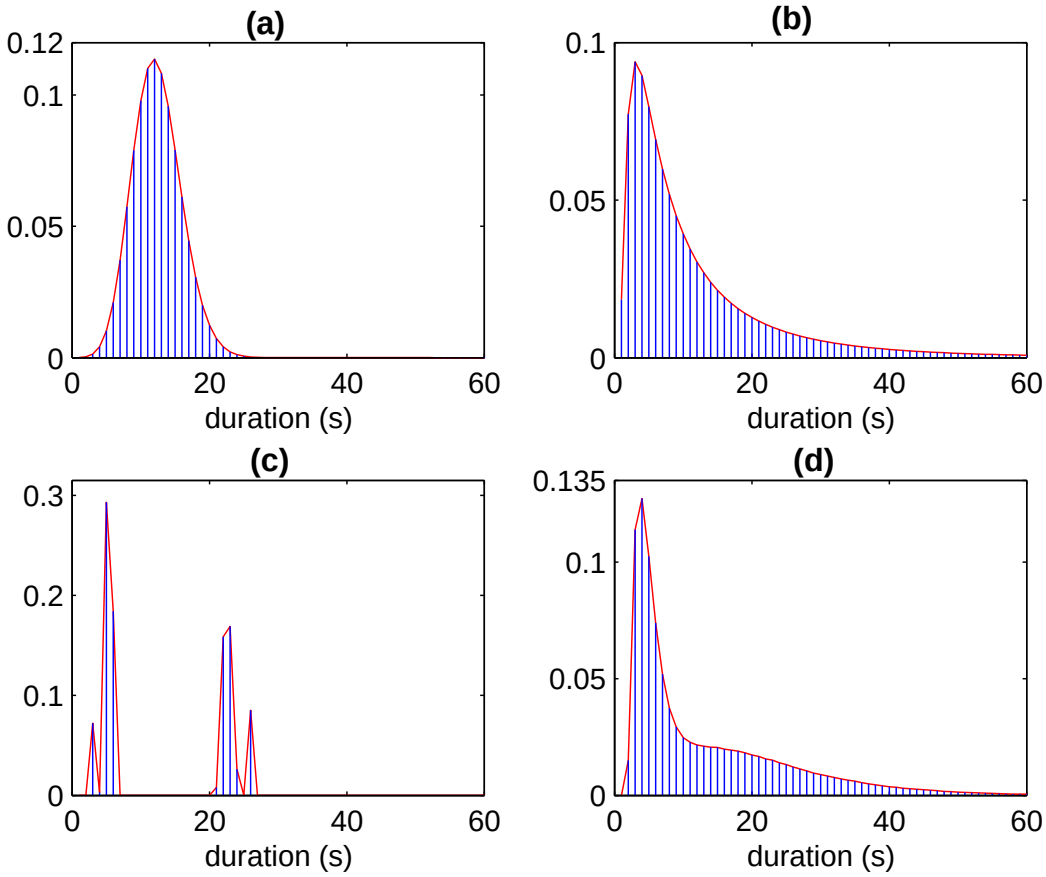
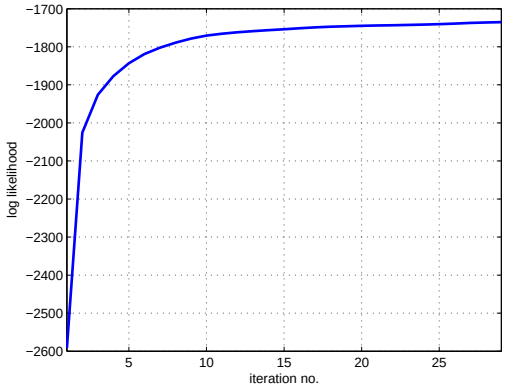
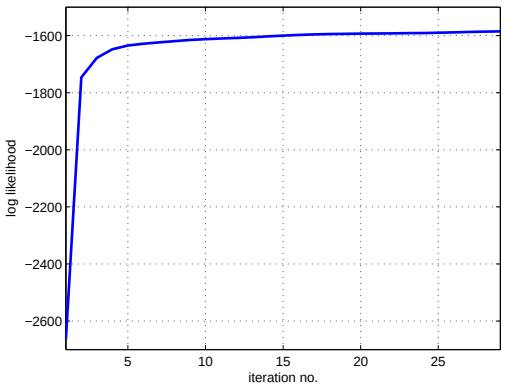


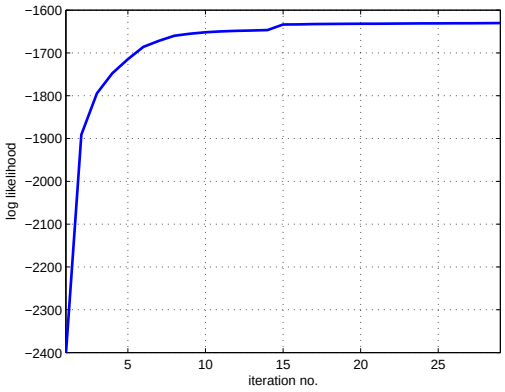
Figure 4.11: The duration distribution of state "at-table" in activity (a.3) learned by: **(a)** the PsHSMM, **(b)** the IgHSMM, **(c)** the MuHSMM, and **(d)** the 5-ph.CxHSMM.



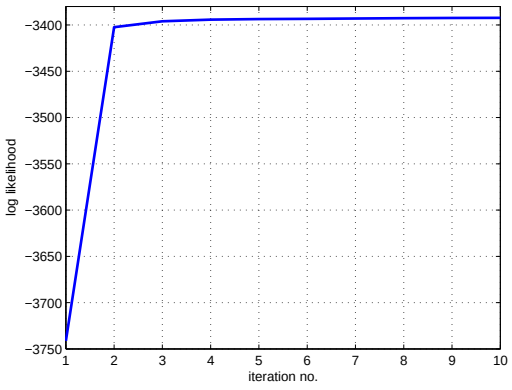
(a) The 5-ph.CxHSMM.



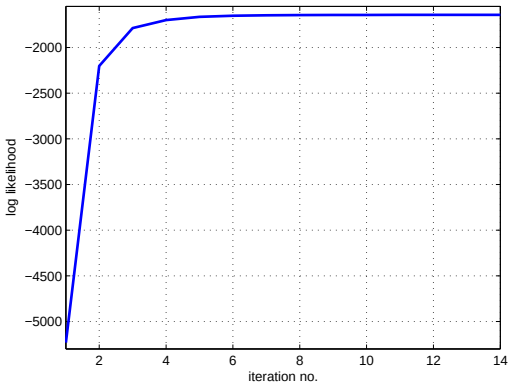
(b) The MuHSMM.



(c) The IgHSMM.



(d) The PsHSMM.



(e) The HMM.

Figure 4.12: The log likelihoods learned from activity (**a.3**).

cause of its approximation to the discrete domain and the additional approximation during EM learning. Nevertheless, the most significant result is that the best accuracy (91.39%) is achieved with $\mathcal{M} = 5$, a relatively *small* number, thus indicating the Coxian as a *promising* choice for modeling ADLs. In addition, the early detection rates (EDRs) in Tab. (4.5) show that on average the 5-ph.CxHSMM is capable of recognizing activities as early as 18.38% of their "lifetime", which is earlier than other $\mathcal{M}$ -phase models. Furthermore, across all activities the 5-ph.CxHSMM has an upper bound of less than 28% EDR, as compared to 35% (or above) for all other values of $\mathcal{M}$. In comparison with the 5-ph.CxHSMM, at the *heavy* expense of computational cost, the MuHSMM performs slightly better with an average of 95.56% for accuracy and 15.59% for EDR[2].

It is further observed that most models generally detect activity (a.1) accurately and early, while sometimes confusing the other two activities. This is consistent with the fact that activities (a.2) and (a.3) share more common durations as shown in Tab. (4.3). Fig. (4.13) illustrates an example of online recognition performed by the 5-ph.CxHSMM for a randomly chosen sequence of activity (a.2).

The comparison between the HMM and the CxHSMM shows us that *by simply adding one more geometric phase* (extending from HMM to 2-ph.CxHSMM) the Markov model can be improved significantly in applications, in particular the accuracy increases from 68.02% to 78.61% (Tab. (4.4)(a & c)), and *by adding a few more geometric phases* (the 5-ph.CxHSMM), we can achieve much better performance (91.39% - Tab. (4.4)(a)). In addition, the model performance slightly decreases for $\mathcal{M} \geq 6$, making $\mathcal{M} = 5$ the best number of phases for this data.

Regarding *running time*, theoretically the $\mathcal{M}$-ph.CxHSMM time scales up linearly by its phase number $\mathcal{M}$, while the MuHSMM and the Exponential Family HSMM (including PsHSMM and IgHSMM) scale up by the maximum duration length M. In our experiment the best performance is achieved with $\mathcal{M} = 5$, while M is in the range of $[100, 120]$, depending on each activity. Thus, the Coxian is far better than any other model in computational time by a theoretical factor of $20 - 24$ times. It is also worthy to noting that compared to the Multinomial duration model,

[2]We have smoothed the Multinomial duration to get rid of the spikes by a moving average, the effectiveness of smoothing is discussed in section 6.1.1.

	$\mathcal{M}=2$ (avg.78.61%)			$\mathcal{M}=3$ (avg.89.03%)			$\mathcal{M}=4$ (avg.85.00%)		
	(a.1)	(a.2)	(a.3)	(a.1)	(a.2)	(a.3)	(a.1)	(a.2)	(a.3)
(a.1)	100	0	0	100	0	0	94.12	5.88	0
(a.2)	0	62.50	37.50	0	93.75	6.25	0	75.00	25.00
(a.3)	13.33	13.33	73.34	0	26.67	73.33	0	20.00	80.00

(a) Tested on different $\mathcal{M}$-ph.CxHSMMs.

	$\mathcal{M}=$ **5 (avg.91.39%)**			$\mathcal{M}=6$ (avg.89.44%)			$\mathcal{M}=7$ (avg.89.17%)		
	(a.1)	(a.2)	(a.3)	(a.1)	(a.2)	(a.3)	(a.1)	(a.2)	(a.3)
(a.1)	**100**	0	0	100	0	0	100	0	0
(a.2)	0	**87.50**	12.50	0	75.00	25.00	0	87.50	12.50
(a.3)	0	13.33	**86.67**	0	6.67	93.33	0	20.00	80.00

(b) Tested on different $\mathcal{M}$-ph.CxHSMMs.

	HMM (avg.68.02%)			PsHSMM (avg.69.05%)			IgHSMM (avg.76.53%)			MuHSMM (avg.95.56%)		
	(a.1)	(a.2)	(a.3)	(a.1)	(a.2)	(a.3)	(a.1)	(a.2)	(a.3)	(a.1)	(a.2)	(a.3)
(a.1)	88.24	0	11.76	58.82	17.65	23.53	100	0	0	100	0	0
(a.2)	0	62.50	37.50	0	75.00	25.00	0	56.25	43.75	0	100	0
(a.3)	13.33	33.33	53.33	0	26.67	73.33	0	26.67	73.33	0	13.33	86.67

(c) Tested on other models.

Table 4.4: Classification accuracy with the data containing missing observations using the HSMM variants and the HMM.

	HMM	PsHSMM	IgHSMM	MuHSMM	$\mathcal{M}=2$	$\mathcal{M}=3$	$\mathcal{M}=4$	$\mathcal{M}=5$	$\mathcal{M}=6$	$\mathcal{M}=7$
(**a.1**)	9.12	31.54	7.99	8.97	7.12	6.47	8.35	**7.26**	7.70	7.84
(**a.2**)	37.28	13.89	47.72	11.77	31.28	11.41	31.39	**20.31**	25.99	17.72
(**a.3**)	42.57	43.96	31.96	26.03	41.76	39.93	56.23	**27.56**	34.47	52.29
Avg.	29.66	29.80	29.22	15.59	26.72	19.27	31.99	**18.38**	22.72	25.95

Table 4.5: Early Detection Rate with data containing missing observations using the HSMM variants and the HMM.

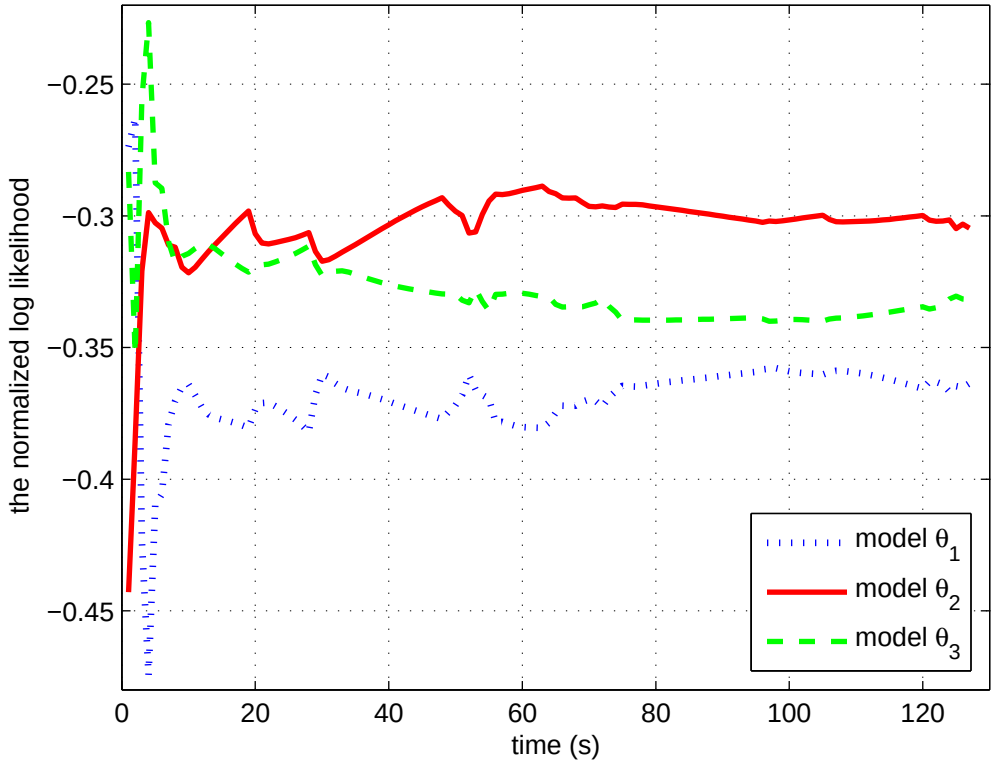


Figure 4.13: Example of online recognition for an unseen sequence of activity (**a**.2) obtained from the 5-ph.CxHSMM. Model θ_i is trained from the set of activities (**a.i**).

the Coxian duration model requires a significantly less number of free parameters: $2 \times \mathcal{M} - 1 = 9$ vs. $M - 1 \approx 114$. Fig. (4.14) shows our MATLAB computation time for one EM iteration run on ten sequences randomly chosen from activities (**a.1**) to (**a.3**). The empirical speed up factor goes from 7 times for the first four sequences, which are from activity (**a.1**) whose lengths are shortest among the three activity types, to 10 times for the next three sequences taken from activity (**a.2**), whose lengths are the generally longest. While the CxHSMM computation time does not increase noticeably with the activity length (i.e. (**a.1**) v.s (**a.2**)), the MuHSMM does suffer considerably more computational cost as it moves from activity (**a.1**) to (**a.2**). The difference between theoretical and empirical speed up factor is due to the actual implementation. In our MATLAB work we have used matrix tricks to significantly speed up the MuHSMM; however, it is impossible to bring it up to the same speed with the CxHSMM and the gap becomes wider as the activity length increases.

Therefore, in comparison with the PsHSMM and the IgHSMM, the CxHSMM is far better, not only in performance but also in running time; whereas in comparison with the MuHSMM the CxHSMM achieves a comparable performance level, but at a fraction of the computational time. We believe that the computational speedup achieved is extremely crucial for semi-Markov models to have their real-world applications, as activity lengths can be arbitrarily long.

4.4.4 Experiment with interpolated dataset $\mathcal{B}$

To further test the robustness of our models towards the problem of missing observation, we construct a dataset $\mathcal{B}$ which is essentially the same as the first dataset, except missing observations are interpolated (averaged in our case) by the values of the previous and next available entries. $\mathcal{B}$ thus does not have missing observation, but may contain approximate or wrong coordinate readings of the actor.

Again, we employ a *leave-one-out* strategy and Tabs. (4.6)-(4.7) present the classification and early detection results (we show here the 2-phase and 3-phase CxHSMMs as representative examples for the Coxian variants). We observe that even though the HMM has improved as compared with the first experiment (Tab. (4.4)(c) vs. Tab. (4.6)), while the 2-ph.CxHSMM and 3-ph.CxHSMM accuracy drops (Tab. (4.4)(a) vs. Tab. (4.6)), both the CxHSMMs still outperform the HMM. One pos-

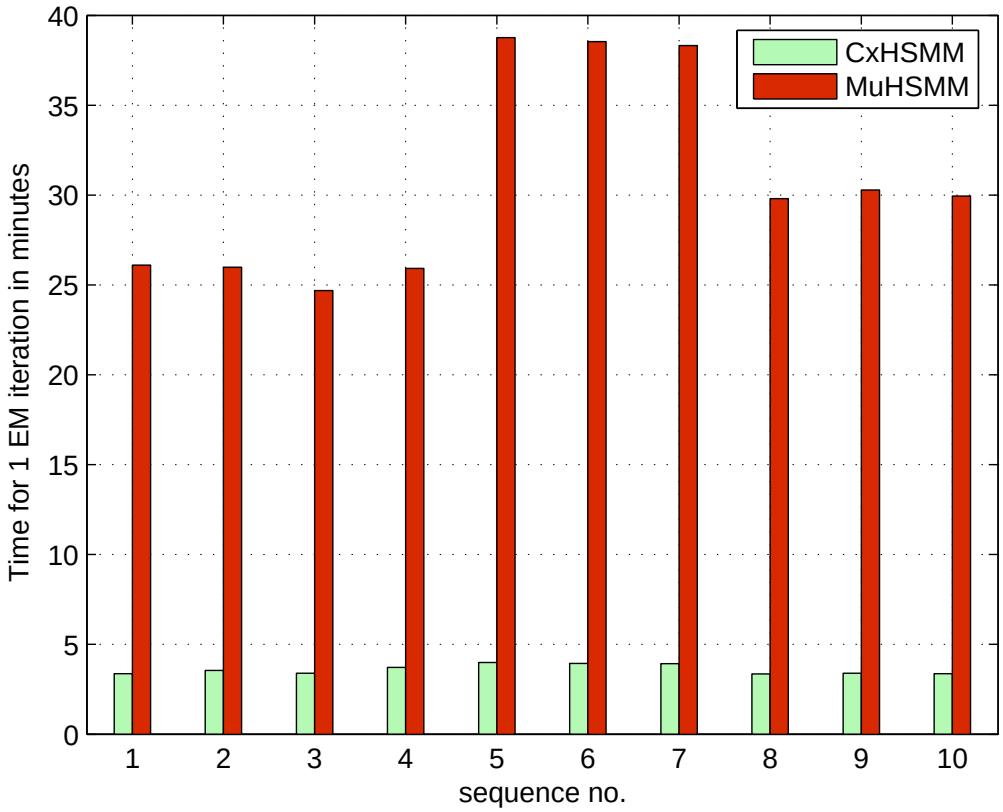


Figure 4.14: EM running time comparison between a 5-ph.CxHSMM and a MuHSMM.

	HMM (avg.72.21%)			$\mathcal{M} = \mathbf{2}$ (avg.74.17%)			$\mathcal{M} = \mathbf{3}$ (avg.82.64%)		
	(a.1)	(a.2)	(a.3)	(a.1)	(a.2)	(a.3)	(a.1)	(a.2)	(a.3)
(**a.1**)	94.12	0	5.88	100	0	0	100	0	0
(**a.2**)	0	62.50	37.50	0	62.50	37.50	0	81.25	18.75
(**a.3**)	20.00	20.00	60.00	0	40.00	60.00	6.67	26.67	66.67

Table 4.6: Classification accuracy (%) with interpolated data using the HMM and the $\mathcal{M}$-ph.CxHSMMs.

	HMM	$\mathcal{M} = \mathbf{2}$	$\mathcal{M} = \mathbf{3}$
(**a.1**)	9.35	8.03	4.63
(**a.2**)	40.52	29.89	9.27
(**a.3**)	43.52	42.96	31.68
Avg.	31.13	26.96	15.19

Table 4.7: Early detection rate (%) with interpolated data using the $\mathcal{M}$-ph.CxHSMMs and the HMM.

sible explanation for the change in performance is that our interpolation method is only appropriate for the case where the track is lost due to a person staying still. If it is due to occlusion, then filling in by averaging neighboring entries are more likely to be inaccurate and states with short durations may now wrongly turn out to be long. By modeling durations, the CxHSMM is thus more sensitive to any wrong interpolations than the HMM. The better performances achieved in the first experiment (i.e. with dataset $\mathcal{A}$) clearly suggest that the type of uncertainty caused by a small percentage of missing observations can be handled robustly and directly by the CxHSMM[3].

4.5 Closing remarks

This chapter started with a review of the discrete Phase-Type (PH) distribution and its special case: the discrete Coxian distribution. Next, we presented the *main* contribution of this chapter. That is the proposal of a *novel* stochastic model: the discrete Coxian duration hidden semi-Markov model (CxHSMM) with a *complete*

[3]When a significant portion of observations are missed, as shown in chapter 6 - section 6.1.3, a small labels may be required to supply in training.

analysis on the CxHSMM consisting of its definition, parameterization, DBN presentation, inference including scaling and learning with/without latent variables. To make the CxHSMM applicable to real-world problems, we addressed the issue of inference and learning with missing observations or observed states. The introduction of the Coxian duration model has several advantages over classic duration parameterization, including its denseness, its low number of free parameters (and thus easy to control), and its computational attractiveness (i.e. computation cost linearly scales up by the number of phases instead of by the maximum duration length as in the Multinomial and Exponential Family case). The *second* main contribution lies in the application. Our experiments with the CxHSMM vs. other existing HSMM models including the MuHSMM, the PsHSMM, the IgHSMM, and the baseline HMM at learning and recognizing ADLs show that the CxHSMM performs comparably with the MuHSMM but at a *fraction* of computation time, and outperforms the rest. In addition, best performance can be achieved when the number of phases of the Coxian is as small as 5, again making it a very attractive model for the ADL domain. The results also point out that the CxHSMM can robustly handle the case when there is a small amount of missing observations (7%) in the data. Finally, our experiments can also be considered as a full investigation into a rich set of duration modeling methods for the HSMM in ADLs.

Given the encouraging results for the duration modeling problem in the HSMM, our next effort is to incorporate hierarchical knowledge into the HSMM to form a new kind of stochastic model that is capable of exploiting both temporal and hierarchical relations in a computationally efficient fashion.

Chapter 5

The Coxian Switching Hidden Semi-Markov Model

Chapters 3 and 4 present both the theory and application of the Hidden Semi-Markov Model (HSMM), an extension in terms of duration modeling into the Hidden Markov Model (HMM). As mentioned in chapter 2, another important extension to the HMM is the introduction of hierarchical knowledge to form the Hierarchical Hidden Markov Model (HHMM), first proposed in [Fine et al., 1998]. However, no attempt has been made to combine both the durational and hierarchical extensions to form a unified model which is capable of fully exploiting both the temporal and structural properties that are inherent in many physical signals. Further, there is also the need for sub-structure sharing with flexible sub-temporal characteristics in modeling hierarchy as this feature is particularly useful in many applications, such as in modeling ADLs. For example, both activities "cooking dinner" and "making coffee" may share the use of the *stove* but with different durations.

Our *main contribution* in this chapter is the introduction of a novel stochastic model named the *Coxian Switching Hidden Semi-Markov Model* (CxSHSMM). The CxSHSMM is a two-level structure. The bottom level is a set of concatenated CxHSMMs and each is initiated by different top-level states that follow a Markov chain. Further, different top-level states can share common states in the CxHSMMs at the bottom level but can also assign them with different duration parameters. The semantics of our proposed CxSHSMM is somewhat similar to the switching linear dynamic system (SLDS), used in several applications such as [Pavlovic et al., 2000] for learning, tracking, synthesizing and classifying human motion and [Oh

et al., 2005] for interpretation of honey-bee dances. While both SHSMM and SLDS have two layers and their top layers switch in a similar manner, they are different in two fundamental ways: our state spaces are discrete while the SLDS is continuous at the lower level and hence cannot model duration information; inference in ours can be done exactly, while that in the SLDS is intractable and thus needs approximate inference. The CxSHSMM offers the following advantages over existing models:

- A capability of automatically modeling both the structural and temporal variations.
- An effective way of modeling the temporal dependencies via the Coxian distribution.
- An ability to share sub-structural units that may carry different sub-temporal characteristics once conditioned on parents.

We present the formal definition for the proposed CxSHSMM and the complete tools for learning and inference with its DBN representation, which include the case of missing observations and labeled states. Furthermore, when no specific distribution is referred to for modeling the state duration at the bottom level, the new model is simply called a Switching Hidden Semi-Markov Model (SHSMM). For the purpose of comparison as well as completeness, we include, in parallel, a study including definitions, inference and parameter estimation of a SHSMM whose state duration at the bottom level is modeled by distributions other than the Coxian.

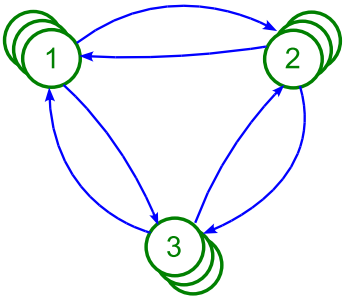
Other contributions in this chapter are extensions to the CxSHSMM to develop the Hierarchical Hidden Semi-Markov Models of any depth, in which duration information can be incorporated at any layer. We also discuss the inference complexity with different inference methods in these models.

The rest of the chapter is organized as follows. We start with the generic HSMM and point out how to integrate hierarchy into this model to form the SHSMM in section 5.1. The formal definition for the CxSHSMM then follows in section 5.2. Next, in section 5.3, we present the DBN representation of the CxSHSMM which clearly shows the conditional (in)dependency between different layers in the model. The problems of inference and learning in the CxSHSMM are then fully discussed in section 5.4 and 5.5, respectively. From sections 5.2 to 5.5, we present a brief

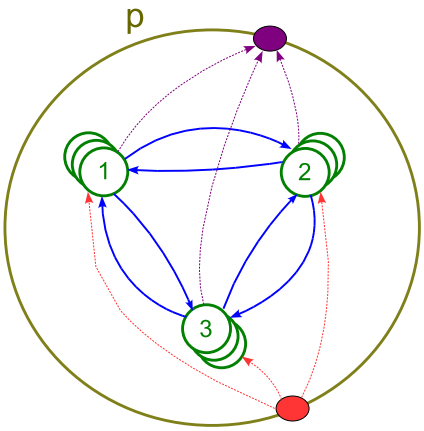
analysis on the Multinomial and Exponential Family duration models in addition to the Coxian. We then present how the SHSMM can be extended into deep hierarchy in section 5.6. Section 5.7 discusses some of the work on extending the SHSMM and CxSHSMM by other authors. Finally, our closing remarks are presented in section 5.8.

5.1 From HSMM to SHSMM: intuition

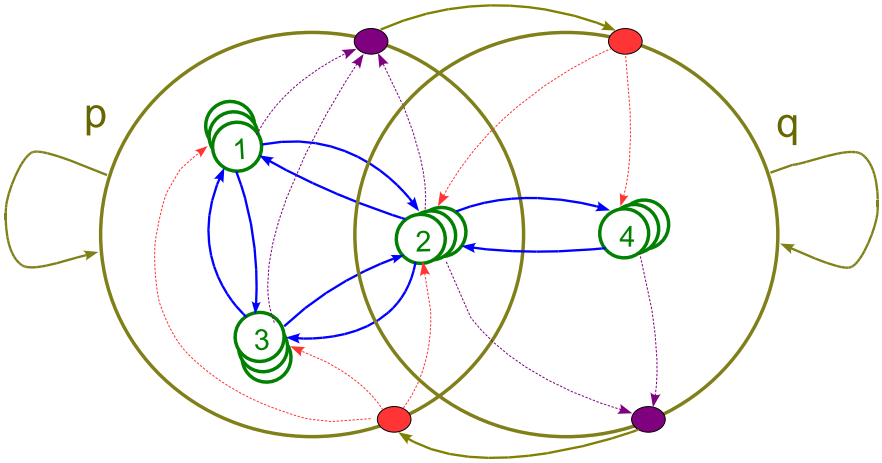
Fig. (5.1)(a) shows an example of a HSMM with three states: $Q = \{1, 2, 3\}$. In this diagram, the duplicated circles associated with each state indicate that the state can last an arbitrary time before making a transition to a new state. We can introduce hierarchy into this HSMM to form the Switching Hidden semi-Markov Model (SHSMM) by viewing all states as children of a higher state p, as shown in Fig. (5.1)(b). When the parent state p is initialized, it can start in any child from 1 to 3 (demonstrated by the red arrows) and a semi-Markov chain within p, also called the p-initiated semi-Markov chain, begins. This semi-Markov chain can end in any child 1, 2 or 3 as shown by the purple arrows. However, it is important to note that the parent state p can be restricted to start or end with only certain child states, if required. When the p-initiated semi-Markov chain ends, it triggers a transition between states at the higher level (Fig. (5.1)(c)). Different from the lower level, which is semi-Markovian, the higher level is strictly Markovian, and thus a transition at this level means the state p repeats itself or transits to a totally new state $q \neq p$. In either case a new semi-Markov chain is initialized at the lower level. The name Switching HSMM comes as this two-layer structure can be viewed as the concatenation of many HSMMs, each initialized by a different "switching" state p. Thus, the dynamics and parameters of the HSMMs at the bottom level are not time invariant, but "switched" from time to time, similar to the way linear Gaussian dynamics are "switched" in a Switching Kalman Filter [Murphy, 1998]. Furthermore, it is important to point out that in this SHSMM high-level states are allowed to share common children. For example, Fig. 5.1(c) show the case where a common child (state 2) is shared between two parents p and q. In addition, child 2 may possess different temporal properties when called by different parents.



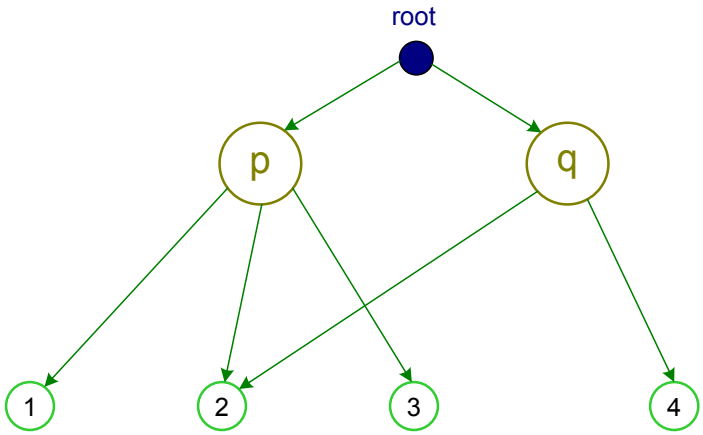
(a) A HSMM with three states.



(b) The HSMM in (a) can be considered as children of a higher state p.



(c) A SHSMM with two parents p and q.



(d) Topology of the SHSMM in (c)

Figure 5.1: From HSMM to SHSMM.

5.2 The CxSHSMM: definition

When we use the Coxian distributions to model durations of child states to exploit its advantage over other existing duration models (chapter 4), the SHSMM is then referred to as a Coxian Switching Hidden semi-Markov Model (CxSHSMM). It is formally defined by a 3-tuple $\langle \zeta, \theta_{\text{CxSHSMM}}, V \rangle$: a topology ζ, a set of parameters θ_{CxSHSMM} and a set of emission alphabets V, as detailed below.

5.2.1 The topology ζ

The CxSHSMM is a two-layer structure, thus, the topology ζ divides the state space into two sets: a set of states at the top level $Q^* = \{1, \ldots, |Q^*|\}$, and a set of states at the bottom level $Q = \{1, \ldots, |Q|\}$. Our convention is to use the letters $p,\ q$ to refer to elements of Q^* and $i,\ j$ to refer to elements of Q. For each parent p in Q^*, the topology defines three associated sets: the children set $\text{ch}(p) \in Q$, the starting set $\text{chS}(p) \subseteq \text{ch}(p)$, and the ending set $\text{chE}(p) \subseteq \text{ch}(p)$. The p-initiated semi-Markov chain is allowed to start with only states in $\text{chE}(p)$, transit among $\text{ch}(p)$, and exit from $\text{chE}(p)$. Note that due to sharing of sub-structures, the sets $\{\text{ch}(p) \cap \text{ch}(q)\}$, $\{\text{chS}(p) \cap \text{chS}(q)\}$, and $\{\text{chE}(p) \cap \text{chE}(q)\}$ may not be empty for $p \neq q$. Finally, the bottom level is also referred to as the production level as it is the only level that emits observation.

Example 5.1. Fig. (5.1)(d) presents the topology of a simple SHSMM whose state transition diagram is shown in Fig. (5.1)(c). This topology may present a simple in-home scenario. For example, the two parent states are: (p)."having lunch" and (q)."reading newspaper & having coffee", and child states $\{1, 2, 3, 4\}$ are the designated spaces in the kitchen: $(1) - Table$, $(2) - Stove$, $(3) - Sink$, and $(4) - Sofa$. Activity (p)."having lunch" includes three atomic activities, $\text{ch}(p) = \{1, 2, 3\}$, conducted sequentially as: (2) "*cooking lunch at Stove*" $\rightarrow$ (1) "*eating lunch at Table*" $\rightarrow$ (3) "*washing dishes at Sink*", while activity (q)."reading newspaper & having coffee" has $\text{ch}(q) = \{2, 4\}$ in the following order: (2) "*making coffee at Stove*" $\rightarrow$ (4) "*reading newspaper - enjoying coffee at Table*".

5.2.2 The parameter set θ_{CxSHSMM}

The Markov chain at the top level of the CxSHSMM is defined by the initial and transition probabilities $\left\{\pi^*_p, A^*_{pq}\right\}$. A transition to p at the top level will initiate a

semi-Markov chain at the bottom level over the states in ch(p). The parameters of this p-initiated chain are given by $\left\{\pi_i^p, A_{ij}^p, A_{i,\text{end}}^p, D_i^p\right\}$ where π_i^p, A_{ij}^p and $A_{i,\text{end}}^p$ are the initial, transition, and terminating probabilities, respectively, while D_i^p is the duration parameter. At each time, an alphabet v from the (discrete) *observation space* V is generated with a probability of $B_{v|i}$, where i is the current state of the p-initiated semi-Markov chain. Tab. (5.1) provides the full definitions of the parameter set θ_{CxSHSMM}.

5.2.2.1 The duration model

Given the disadvantages of existing duration models (i.e. Multinomial and Exponential Family distributions), as described in chapter 3, we propose the use of the Coxian distribution to model state durations at the bottom level in the CxSHSMM.

For each p-initiated semi-Markov sequence, the duration distribution of a child state i is specified by parameter $D_i^p \triangleq \text{Cox}(\boldsymbol{\mu}^{p,i}, \boldsymbol{\lambda}^{p,i})$. As in the CxHSMM (chapter 4), both the phase initial probabilities $\boldsymbol{\mu}^{p,i} = \left[\mu_1^{p,i}, \ldots, \mu_{\mathcal{M}}^{p,i}\right]^\mathsf{T}$ and the phase transition probabilities $\boldsymbol{\lambda}^{p,i} = \left[\lambda_1^{p,i}, \ldots, \lambda_{\mathcal{M}}^{p,i}\right]^\mathsf{T}$ are $\mathcal{M}$-dimensional vectors where $\mathcal{M}$ is a fixed constant representing the number of geometric phases in the discrete Coxian (Fig. (5.2)), and $\sum_{n=1}^{\mathcal{M}} \mu_n^{p,i} = 1$, $0 < \lambda_n^{p,i} \leq 1$. The Coxian duration starts a transient phase $n \in [1, \mathcal{M}]$ with probability $\mu_n^{p,i}$, and then makes a transition to the next phase $n+1 \leq \mathcal{M}$ with the probability $\lambda_n^{p,i}$. The process continues until the last phase $n = \mathcal{M}$ is reached, and the Coxian then moves to its absorbing state with a terminating probability of $\lambda_{\mathcal{M}}^{p,i}$. Finally, note that for $\mathcal{M} = 1$, the CxSHSMM reduces to a HHMM.

In addition, if distributions other than the Coxian are used to model the state durations, the prefix of the name SHSMM would change accordingly, e.g. a Multinomial Switching Hidden semi-Markov model (MuSHSMM) would mean its bottom-level state durations are modeled by Multinomial distributions. Further, if Multinomial or Exponential Family distributions are used, all parameters in Tab. (5.1) remain unchanged except for the duration model. We then have to define a new duration parameter $D_i^p = [D_i^p(1), \ldots, D_i^p(M)]^\mathsf{T}$ with M being the maximum allowed duration, $D_i^p(\tau)$ is the probability of child state i in the p-initiated chain having duration τ, and stochastic constraints require $\sum_{n=1}^{M} D_i^p(n) = 1$. In a special case of the CxSHSMM, $\lambda_n^{p,i} = 1$, $\forall p, i, n$, the CxSHSMM becomes a MuSHSMM with

At the top level

Parameters	Meanings, dimension, constraints
π^*_p	is the initial probability of parent p, dim:$1 \times \lvert Q^*\rvert$, and $\sum_{p\in Q^*} \pi^*_p = 1$.
A^*_{pq}	is the transition probability from parent p to parent q, dim: $\lvert Q^*\rvert \times \lvert Q^*\rvert$, and $\sum_{q\in Q^*} A^*_{pq} = 1$.

At the bottom level

π^p_i	is the initial probability of child i in the p-initiated semi-Markov chain, dim: $\lvert \mathrm{ch}(p)\rvert \times \lvert \mathrm{ch}(p)\rvert$, and $\sum_{i\in \mathrm{ch}(p)} \pi^p_i = 1$.
A^p_{ij}	is the transition probability from child i to child j in the p-initiated semi-Markov chain, dim: $\lvert \mathrm{ch}(p)\rvert \times \lvert \mathrm{ch}(p)\rvert$, and $A^p_{ii} = 0$.
$A^p_{i,\mathrm{end}}$	is the transition probability of child i going to end in the p-initiated semi-Markov chain, dim: $1 \times \lvert \mathrm{ch}(p)\rvert$, and $\sum_{j\in \mathrm{ch}(p)} A^p_{ij} + A^p_{i,\mathrm{end}} = 1$.
$D^p_i = \mathrm{Cox}\left(\boldsymbol{\mu}^{p,i}, \boldsymbol{\lambda}^{p,i}\right)$	is the Coxian duration model of child i in the p-initiated semi-Markov chain.
$\mu^{p,i}_n$	is the initial probability of phase n, dim: $1 \times \mathcal{M}$, and $\sum_{n=1}^{\mathcal{M}} \mu^{p,i}_n = 1$.
$\lambda^{p,i}_n$	is the terminating probability of phase n, dim: $1 \times \mathcal{M}$, and $0 < \lambda^{p,i}_n \leq 1$.

At the emission level

$B_{v\lvert i}$	is the emission probability of symbol v from the current child state i, dim: $\lvert Q\rvert \times \lvert V\rvert$, and $\sum_{v\in V} B_{v\lvert i} = 1$.

Table 5.1: Parameter definitions for the CxSHSMM, where $\lvert \mathrm{ch}(p)\rvert$ denotes the number of elements in $\mathrm{ch}(p)$, and $\mathcal{M}$ is the number of phases of the Coxian distributions.

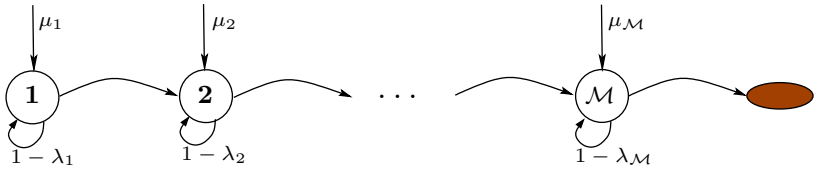


Figure 5.2: The phase diagram of an $\mathcal{M}-$phase Coxian. The clear circles are the transient phases and the shaded ellipse is the absorbing phase.

$D_i^p = \left[\mu_1^{p,i}, \ldots, \mu_{\mathcal{M}}^{p,i}\right]^{\mathsf{T}}$.

5.3 Dynamic Bayesian Network Representation

Fig. (5.3) shows the DBN representation of the CxSHSMM over two time-slices, which in turn can be viewed as a hierarchical extension[1] of the CxHSMM's DBN in Fig. (4.6). A set of variables $\mathcal{V}_t = \{z_t, \epsilon_t, x_t, e_t, m_t, y_t\}$ is maintained at any given time slice t:

- At the top level:
  - z_t is the current top-level state acting as a switching variable. Every time z_t is switched to a new state, it initializes a semi-Markov chain at the bottom level.
  - ϵ_t is a Boolean-valued variable set to 1 when the z_t-initiated semi-Markov sequence ends at the current time-slice.
- At the bottom level:
  - x_t is the current child state in the z_t-initiated semi-Markov sequence.
  - e_t is a Boolean-valued variable set to 1 when x_t reaches the end of its duration[2].

[1] For a full study on hierarchical decomposition of stochastic dynamic process modeled in DBN, readers are referred to work of [Bui et al., 2000, Murphy and Paskin, 2001, Phung, 2005b].

[2] In an HSMM, t is the end of duration of the state x_t iff $x_t \neq x_{t+1}$. However, in an SHSMM, it is possible that x_{t+1} is actually part of a newly initiated HSMM. Thus $x_{t+1} \neq x_t$ if $e_t = 1$ and $\epsilon_t = 0$, but we can have $x_{t+1} = x_t$ if $e_t = \epsilon_t = 1$.

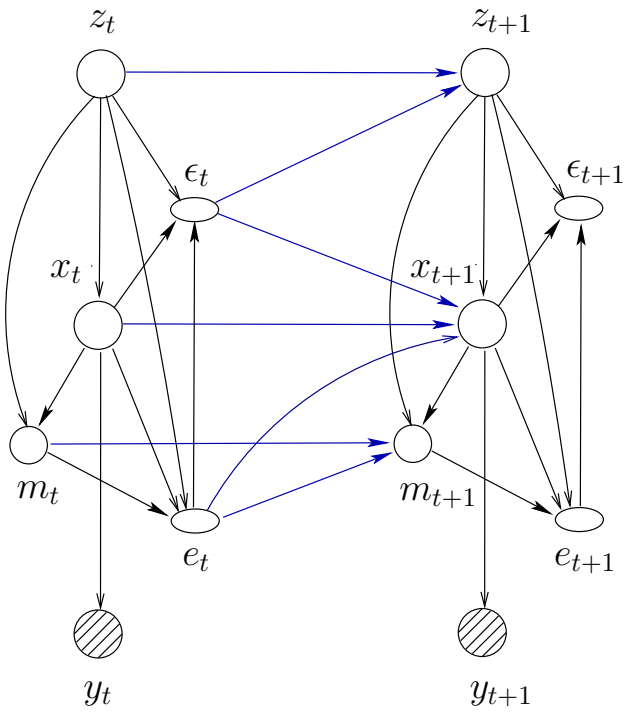


Figure 5.3: DBN representation of the CxSHSMM for two time-slices.

- m_t is an $\mathcal{M}$-valued variable representing the current phase of x_t and only transient phases are shown in the DBN[3].
- y_t is the observed symbol emitted by the system at time t.

Generally, $\{y_{1:T}\}$ are observed as indicated by shaded nodes, while $\mathcal{V}_{1:T} \setminus \{y_{1:T}\}$ are hidden as shown by clear nodes. However, in practice, we sometimes encounter missing observations or are supplied with labeled data at some point in time. At any time t, if the observation y_t is missing, then the node y_t is simply discarded from the DBN, and thus the observation at t is set to $\{\varnothing\}$. On the other hand, if some states are observed at time t (labeled data), their corresponding nodes are shaded and labeled with the observed values. For example, let $\bar{z}_t$ denote the observed value of z_t, then node z_t is shaded and labeled with $\bar{z}_t$. As shown later, the inference and learning algorithms developed based on this DBN representation (Fig. (5.3)) can be easily modified to handle these scenarios.

[3]Once the current Coxian duration leaves its last transient phase for the absorbing phase, the DBN continues with representing a newly initiated Coxian at the next time slice.

5.3.1 Network construction

The DBN for the CxSHSMM is constructed with two assumptions: (i.) the parent at the top level cannot end unless its child states at the bottom level do so first; and (ii.) a child state, in turn, cannot end if its duration has not yet expired, which happens when its Coxian duration model leaves the last transient phase for the absorbing phase. Equivalently, if the Coxian has not reached its last transient phase, the child state has to carry on the the next time slice, and so does its parent. Therefore, from a bottom-up view:

$$\begin{aligned} & \forall t, \quad m_t < \mathcal{M} \implies e_t = 0 \\ \text{and } & \forall t, \quad e_t = 0 \implies \epsilon_t = 0 \end{aligned}$$

Alternatively, from a top-down view:

$$\begin{aligned} & \forall t, \quad \epsilon_t = 1 \implies e_t = 1 \\ \text{and } & \forall t, \quad e_t = 1 \implies m_t = \mathcal{M} \end{aligned}$$

Now we start with the network construction at time $t = 1$. Firstly, at the top level the variable z_1 is initialized to a parent state $p \in Q^*$ with probability π^*_p, which then triggers a semi-Markov chain at the bottom level defined over its child states $\mathrm{ch}(p)$. This semi-Markov chain starts in its initial state $x_1 = i,$ for $i \in \mathrm{ch}(p)$, with a probability π^p_i. Next, the Coxian distribution associated with state $i \in \mathrm{ch}(p)$, termed the (p, i)-Coxian, is then activated in its transient phase $m_1 = n$ with probability $\mu^{p,i}_n$. If $m_1 = n < \mathcal{M}$, i.e. the (p, i)-Coxian has not reached its last phase, the bottom-state ending variable e_1 must be set to 0 regardless of z_1 and x_1, as shown by the broken dependencies in Fig. (5.4)(a). This is a form of context specific independence in the BN where the dependencies in the network change with certain observed context values. In contrast, if $m_1 = \mathcal{M}$, the causal relationships between e_1 and $\{z_1, x_1\}$ remain active (Fig. (5.4)(b)), and e_1 is assigned to 1 with probability $\lambda^{p,i}_{\mathcal{M}}$, or to 0 with probability $1 - \lambda^{p,i}_{\mathcal{M}}$. These relations apply to any time $t \in [1, T]$, therefore:

$$\forall t \in [1, T], \ \Pr\left(e_t = 1 \mid z^p_t, x^i_t, m^n_t\right) = \begin{cases} 0, & n < \mathcal{M} \\ \lambda^{p,i}_{\mathcal{M}}, & n = \mathcal{M} \end{cases}$$

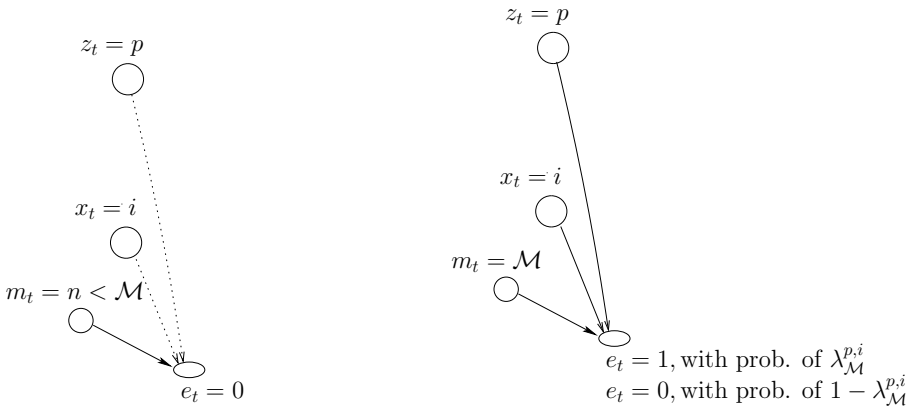


Figure 5.4: The conditional dependencies over node e_t and its parents. Dotted lines show broken dependencies.

Also, when $e_1 = 0$, since the top state cannot switch if its current child has not ended, ϵ_1 must be fixed to 0; otherwise when $e_1 = 1$, ϵ_1 is assigned to 1 with probability $A^p_{i,\text{end}}$ signaling the end of the p-initiated semi-Markov chain, and to 0 with probability $1 - A^p_{i,\text{end}}$. Further, as illustrated in Fig. (5.5), these dependencies hold for all $t \in [1, T]$, thus:

$$\forall t \in [1, T]\,,\ \Pr\left(\epsilon_t = 1 \mid z^p_t, x^i_t, e^k_t\right) = \begin{cases} 0, & k = 0 \\ A^p_{i,\text{end}}, & k = 1 \end{cases}$$

Lastly, the observation y_1 is generated from $x_1 = i$ with an emission probability $B_{y_1|i}$.

After the network initialization at $t = 1$, the ending variables ϵ_t and e_t act like context in term of defining how the next time-slice $t+1$ can be derived from the current time-slice t (shown by the blue connections in Fig. (5.3)). When $\{\epsilon_t = 0,\ e_t = 0\}$, the same states at the top and bottom levels carry on to the next time-slice while the phase variable m_t may have the choice of moving to the next phase or staying in the same phase (Fig. (5.6)(a-b)):

$$\Pr\left(m^{n+1}_{t+1} \mid m^n_t, z^p_{t+1}, x^i_{t+1}, e^0_t\right) = \lambda^{p,i}_n, \quad n < \mathcal{M}$$

$$\Pr\left(m^n_{t+1} \mid m^n_t, z^p_{t+1}, x^i_{t+1}, e^0_t\right) = \begin{cases} 1 - \lambda^{p,i}_n, & n < \mathcal{M} \\ 1, & n = \mathcal{M} \end{cases}$$

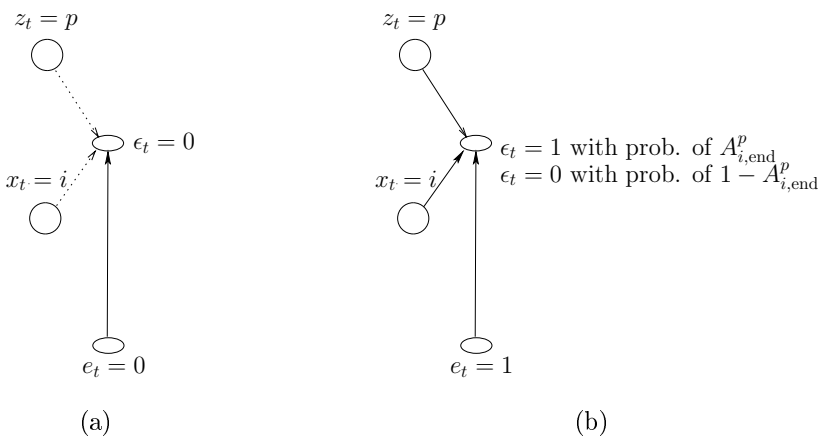


Figure 5.5: The conditional dependencies over ϵ_t and its parents. Dotted lines show broken dependencies.

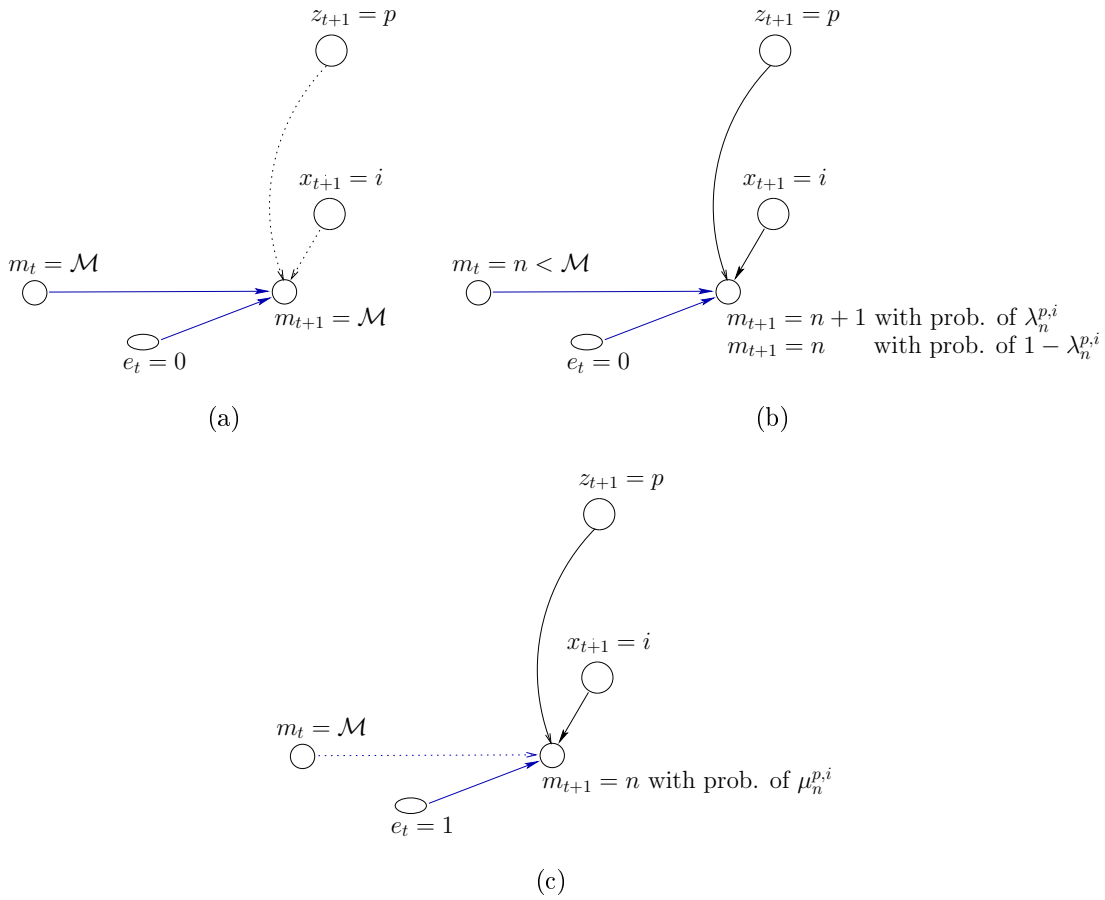


Figure 5.6: The conditional dependencies over node m_t and its parents. Dotted lines show broken dependencies.

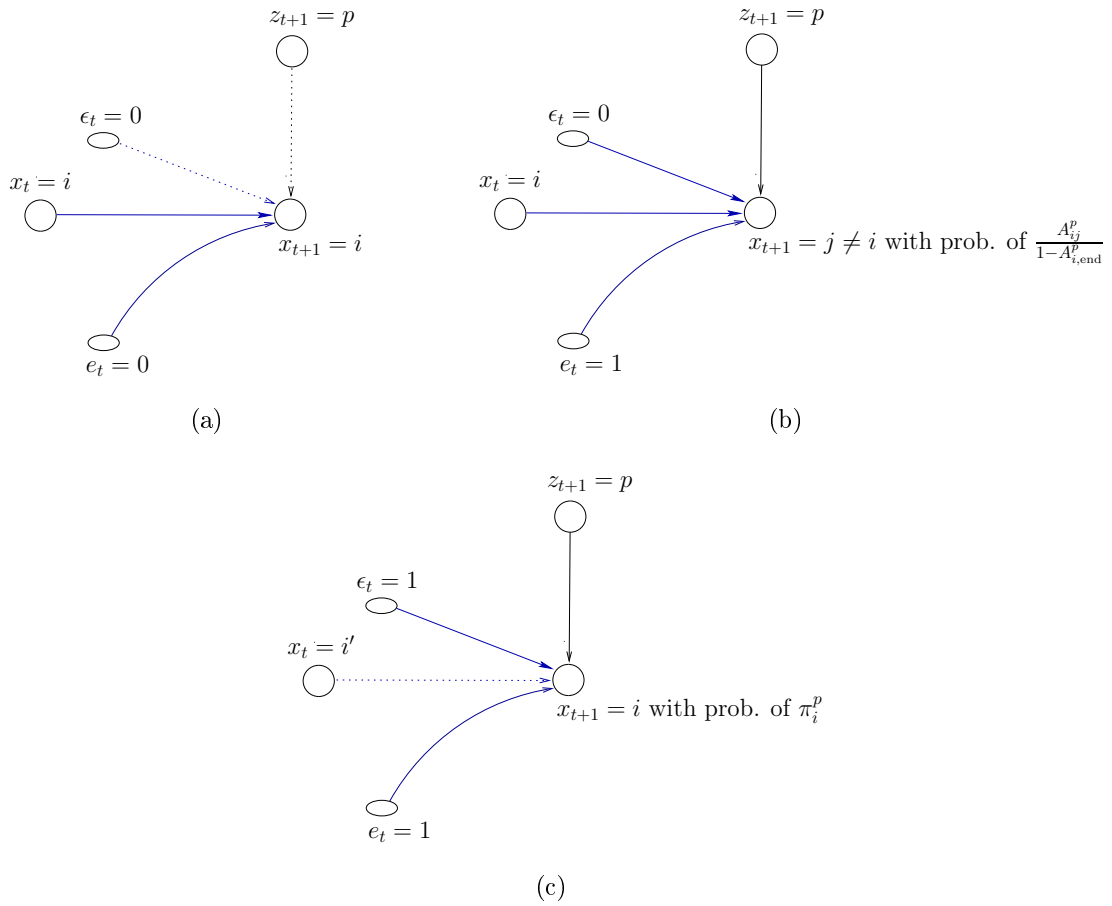


Figure 5.7: The conditional dependencies over node x_{t+1} and its parents. Dotted lines show broken dependencies.

When $\{\epsilon_t = 0,\ e_t = 1\}$, the same top-level state p carries on to the next time-slice, but the semi-Markov sequence at the bottom level transits from its current state i to a new state $j \neq i$ (Fig. (5.7)(b)) with probability:

$$\Pr\left(x^j_{t+1} \mid x^i_t, z^p_{t+1}, \epsilon^0_t, e^1_t\right) = \frac{A^p_{ij}}{1 - A^p_{i,\text{end}}}$$

and a new Coxian associated with state j is activated in its transient phase (Fig. (5.6)(c)) with probability:

$$\Pr\left(m^n_{t+1} \mid z^p_{t+1}, x^j_{t+1}, e^1_t\right) = \mu^{p,j}_n$$

At the top level:		
π^*_p	$=$	$\Pr\left(z^p_1\right)$
A^*_{pq}	$=$	$\Pr\left(z^q_{t+1} \mid x^p_t, \epsilon^1_t\right), \forall t \in [1, T-1]$
At the bottom level:		
π^p_i	$=$	$\Pr\left(x^i_{t+1} \mid z^p_{t+1}, \epsilon^1_t, e^1_t\right), \forall t \in [0, T-1]$
A^p_{ij}	$=$	$\Pr\left(x^j_{t+1}, \epsilon^0_t \mid x^i_t, z^p_{t+1}, e^1_t\right), \forall t \in [1, T-1]$
$A^p_{i,\text{end}}$	$=$	$\Pr\left(\epsilon^1_t \mid z^p_t, x^i_t, e^1_t\right), \forall t \in [1, T]$
D^p_i	$=$	$\text{Cox}\left(\boldsymbol{\mu}^{p,i}, \boldsymbol{\lambda}^{p,i}\right)$
$\mu^{p,i}_n$	$=$	$\Pr\left(m^n_{t+1} \mid z^p_{t+1}, x^i_{t+1}, e^1_t\right), \forall t \in [0, T-1]$
$\lambda^{p,i}_{n<\mathcal{M}}$	$=$	$\Pr\left(m^{n+1}_{t+1} \mid m^n_t, z^p_{t+1}, x^i_{t+1}, e^0_t\right), \forall t \in [1, T-1]$
$\lambda^{p,i}_{\mathcal{M}}$	$=$	$\Pr\left(e^1_t \mid z^p_t, x^i_t, m^{\mathcal{M}}_t\right), \forall t \in [1, T]$
$B_{v\|i}$	$=$	$\Pr\left(y^v_t \mid x^i_t\right), \forall t \in [1, T]$

Table 5.2: Mapping from the model parameters θ_{CxSHSMM} to the DBN parameters. Nodes ϵ_0 and e_0 are set to 0 by default.

When $\{\epsilon_t = 1, e_t = 1\}$, the top-level state p "switches" to the next state q, where q could be the same as p, with probability:

$$\Pr\left(z^q_{t+1} \mid x^p_t, \epsilon^1_t\right) = A^*_{pq}$$

and a new semi-Markov sequence is initiated at the bottom level (Fig. (5.7)(c)) with probability:

$$\Pr\left(x^i_{t+1} \mid z^p_{t+1}, \epsilon^1_t, e^1_t\right) = \pi^p_i$$

Also, at the same time, a new Coxian defining the duration of state i is initialized to in its transient phase n with probability $\mu^{q,i}_n$.

Finally, for convenience a full summary of parameters in θ_{CxSHSMM} when mapped into the DBN is presented in Tab. (5.2).

5.3.2 DBN representation with duration models other than the Coxian

We present a DBN representation of the SHSMM in Fig. (5.8) where state durations at its bottom level are modeled by the Multinomial or the Exponential Family distributions. The set of phase and ending variables $\{m_t, e_t\}$ at the bottom level is now

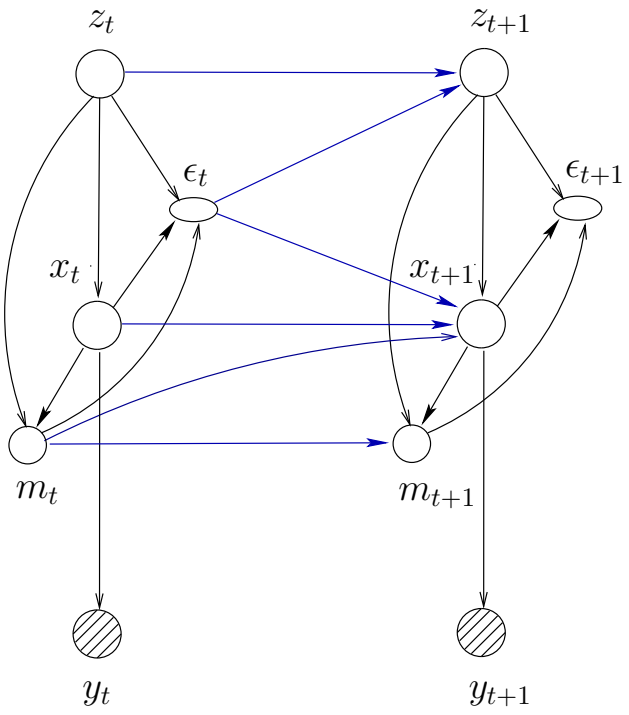


Figure 5.8: DBN representation of the SHSMM whose state durations at the bottom level are modeled by Multinomial or Exponential Family distributions.

reduced to a "count" variable m_t. Every time the semi-Markov chain at the bottom level enters a new state, the count variable is initiated to that new state's duration, which is a positive number less than or equal to the predefined maximum duration M. The count variable then counts down one unit every time slice until it reaches 1, indicating the end of this new state. Hence, together with the top-level ending variable ϵ_t, the count variable m_t (replaced e_t in the previous case) now acts as a context, defining how the next time slice $t+1$ is related to the current time slice t. When $m_t > 1$, the bottom-level state has not finished its duration; hence, the top-level ending variable ϵ_t must be fixed to 0, and the states at both levels continue to the next time slice. When $m_t = 1$, the bottom-level state i comes to end, and there are two possibilities: if $\epsilon_t = 0$, the same top-level state proceeds to the next slice, while the semi-Markov chain at the bottom level transits to a new child state; if $\epsilon_t = 1$, the top-level state switches to a new state and a new semi-Markov chain starts at the bottom level.

Now we look at the duration variable m_t in more detail. When $m_t > 1$, the duration

of the current state has not expired, thus, it continues to count down at the next time slice:

$$\forall \tau > 1,\ \Pr\left(m_{t+1} = \tau - 1 \mid m_t^\tau, z_{t+1}^p, x_{t+1}^i\right) = 1$$

When $m_t = 1$, a duration τ of a new state (within the same p-initiated semi-Markov sequence if $\epsilon_t = 0$, or of a newly p-initiated semi-Markov sequence if $\epsilon_t = 1$) is assigned with probability:

$$\Pr\left(m_{t+1} = \tau \mid m_t^1, z_{t+1}^p, x_{t+1}^i\right) = D_i^p(\tau)$$

If the duration is modeled by a Multinomial, then $D_i^p \sim Mult\left(D_i^p(1), \ldots, D_i^p(M)\right)$. On other hand, if distribution from the Exponential Family is used, then:

$$D_i^p(\tau) = h\left(\tau\right) \exp\left\{\boldsymbol{w}^\mathsf{T} T\left(\tau\right) - A\left(\boldsymbol{w}\right)\right\}$$

where $h\left(\tau\right)$ is the base function, $\boldsymbol{w}$ are the natural parameters, $T\left(\tau\right)$ is the sufficient statistic and $A\left(\boldsymbol{w}\right)$ is the log-partition function. For example, if the distribution is a Poisson with mean λ_i^p, $D_i^p(\tau)$ has the following form:

$$D_i^p(\tau) = \frac{1}{\tau!} \exp\left\{\tau \log \lambda_i^p - \lambda_i^p\right\}$$

Tab. (5.3) presents the complete list of parameters in θ_{SHSMM}, in which durations of states at the bottom level are modeled by Multinomial or Exponential Family, when mapped into its DBN conditional probabilities depicted Fig. (5.8).

5.4 Inference

In the inference task, let $S_t \triangleq \{z_t, \epsilon_t, x_t, m_t, e_t\}$ be the amalgamated *hidden* state, we are interested in computing the filtering distribution $\Pr(S_t \,|\, y_{1:t})$ and the smoothing distributions $\gamma_t\left(\mathbf{s}\right) = \Pr(S_t \,|\, y_{1:T})$ and $\xi_t\left(\mathbf{s}, \mathbf{s}'\right) = \Pr\left(S_t^{\mathbf{s}}, S_{t+1}^{\mathbf{s}'} \mid y_{1:T}\right)$. A range of queries regarding the current top-level state z_t, the current bottom-level state x_t and the remaining duration of the current bottom-level state can be answered from the marginals of these distributions. In the normal setting or in the presence of missing observations or labeled states, the inference, including scaling, is done in a similar fashion to that of the CxHSMM (section 4.3.4). However, the amalgamated hidden state S_t is now extended to include two more variables: the parent state z_t and the switching state ϵ_t. The state space of S_t is now $O(|Q^*||Q|\mathcal{M})$, therefore, the recursive complexities of the smoothing distribution is $O(|Q^*|^2|Q|^2\mathcal{M}T)$ as the full joint probability of m_t and m_{t+1} is just $O(\mathcal{M})$ instead of $O(\mathcal{M}^2)$ (section 4.3.4.1).

At the top level:		
π^*_p	$=$	$\Pr\left(z^p_1\right)$
A^*_{pq}	$=$	$\Pr\left(z^q_{t+1} \mid x^p_t, \epsilon^1_t\right), \forall t \in [1, T-1]$
At the bottom level:		
π^p_i	$=$	$\Pr\left(x^i_{t+1} \mid z^p_{t+1}, \epsilon^1_t, m^1_t\right), \forall t \in [0, T-1]$
A^p_{ij}	$=$	$\Pr\left(x^j_{t+1}, \epsilon^0_t \mid x^i_t, z^p_{t+1}, m^1_t\right), \forall t \in [1, T-1]$
$A^p_{i,\text{end}}$	$=$	$\Pr\left(\epsilon^1_t \mid z^p_t, x^i_t, m^1_t\right), \forall t \in [1, T]$
$D^p_i(\tau)$	$=$	$\Pr\left(m^\tau_{t+1} \mid m^1_t, z^p_{t+1}, x^i_{t+1}\right), \forall t \in [0, T-1]$
$B_{v\|i}$	$=$	$\Pr\left(y^v_t \mid x^i_t\right), \forall t \in [1, T]$

Table 5.3: Mapping from the model parameters to the DBN parameters for the Multinomial/Exponential Family duration SHSMM. Nodes ϵ_0 and m_0 are set to 0 by default.

5.4.1 Inference with scaling

Similar to inference in the HMM and CxHSMM (sections 2.4.3 and 4.3.4), the filtering distribution $\Pr\left(S_t \mid y_{1:t}\right)$, also called the scaled forward variable[4], is computed recursively by using the Markov properties to decompose its local conditional independencies. For convenience, we use the compact notation $S^{\mathbf{s}}_t \triangleq \left\{z^p_t, \epsilon^l_t, x^i_t, m^n_t, e^k_t\right\}$ to denote a realization of the amalgamated hidden state. The scaled forward variable is defined by:

$$\tilde{\alpha}_t\left(\mathbf{s}\right) \triangleq \Pr\left(S^{\mathbf{s}}_t \mid y_{1:t}\right) = \frac{\ddot{\alpha}_t\left(\mathbf{s}\right)}{\psi_t} \tag{5.1}$$

in which $\ddot{\alpha}_t\left(\mathbf{s}\right) = \Pr\left(S^{\mathbf{s}}_t, y_t \mid y_{1:t-1}\right)$ is the partially forward variable, and $\psi_t = \Pr\left(y_t \mid y_{1:t-1}\right)$ is the familiar scaling factor. The recursive computation at time

[4]We can begin with the forward variable $\Pr\left(S_t, y_{1:t}\right)$, then move to the scaled forward variable $\Pr\left(S_t \mid y_{1:t}\right)$. However, as this step has been presented in detail for the CxHSMM in the previous chapter and in practice we normally work with the scaled version to prevent numerical underflow, we thus go directly to inference with scaling here.

$t+1$ for the partially forward variable is given as:

$$\begin{aligned}
\ddot{\alpha}_{t+1}\left(\mathbf{s}' \triangleq \{p', l', i', n', k'\}\right) &= \Pr\left(S_{t+1}^{\mathbf{s}'}, y_{t+1} \mid y_{1:t}\right) \\
&= \sum_{\mathbf{s}} \Pr\left(S_t^{\mathbf{s}}, S_{t+1}^{\mathbf{s}'}, y_{t+1} \mid y_{1:t}\right) \\
&= \sum_{\mathbf{s}} \Pr\left(y_{t+1} \mid S_{t+1}^{\mathbf{s}'}\right) \Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right) \Pr\left(S_t^{\mathbf{s}} \mid y_{1:t}\right) \\
&= B_{y_{t+1}|i'} \sum_{\mathbf{s}} \Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right) \tilde{\alpha}_t\left(\mathbf{s}\right) \qquad (5.2)
\end{aligned}$$

The only difference in the recursive expression of the partially forward variable of the CxSHSMM in Eq. (5.2) and that of the CxHSMM in Eq. (4.35) is how the transition probability $\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right)$ is computed. As the CxSHSMM has hierarchy, its transition probability is more complex. We first need to break $\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right)$ into its embedded conditional probabilities:

$$\begin{aligned}
\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right) &\triangleq \Pr\left(z_{t+1}^{p'}, \epsilon_{t+1}^{l'}, x_{t+1}^{i'}, m_{t+1}^{n'}, e_{t+1}^{k'} \mid z_t^p, \epsilon_t^l, x_t^i, m_t^n, e_t^k\right) \\
&= \Pr\left(\epsilon_{t+1}^{l'} \mid z_{t+1}^{p'}, x_{t+1}^{i'}, e_{t+1}^{k'}\right) \Pr\left(e_{t+1}^{k'} \mid z_{t+1}^{p'}, x_{t+1}^{i'}, m_{t+1}^{n'}\right) \\
&\quad \times \Pr\left(m_{t+1}^{n'} \mid z_{t+1}^{p'}, x_{t+1}^{i'}, m_t^n, e_t^k\right) \Pr\left(x_{t+1}^{i'} \mid z_{t+1}^{p'}, \epsilon_t^l, x_t^i, e_t^k\right) \\
&\quad \times \Pr\left(z_{t+1}^{p'} \mid z_t^p, \epsilon_t^l\right)
\end{aligned}$$

and except for the simple probability $\Pr\left(z_{t+1}^{p'} \mid z_t^p, \epsilon_t^l\right)$, all other local probabilities defined over a node and its parents are illustrated in Figs. (5.4) to (5.7). In particular, depending on the the context defined by the ending variables and the duration

variable, these local probabilities take on different values as summarized below:

$$\Pr\left(\epsilon_{t+1}^{l'} \mid z_{t+1}^{p'}, x_{t+1}^{i'}, e_{t+1}^{k'}\right) = \begin{cases} \delta_{l'}^{(0)}, & k' = 0 \\ \left(1 - A_{i',\text{end}}^{p'}\right)\delta_{l'}^{(0)} + A_{i',\text{end}}^{p'}\delta_{l'}^{(1)}, & k' = 1 \end{cases}$$

$$\Pr\left(e_{t+1}^{k'} \mid z_{t+1}^{p'}, x_{t+1}^{i'}, m_{t+1}^{n'}\right) = \begin{cases} \delta_{k'}^{(0)}, & n' < \mathcal{M} \\ \left(1 - \lambda_{\mathcal{M}}^{p',i'}\right)\delta_{k'}^{(0)} + \lambda_{\mathcal{M}}^{p',i'}\delta_{k'}^{(1)}, & n' = \mathcal{M} \end{cases}$$

$$\Pr\left(m_{t+1}^{n'} \mid z_{t+1}^{p'}, x_{t+1}^{i'}, m_t^{n}, e_t^{k}\right) = \begin{cases} \delta_{n'}^{(\mathcal{M})}, & k = 0,\ n < \mathcal{M} \\ \left(1 - \lambda_{n}^{p',i'}\right)\delta_{n'}^{(n)} + \lambda_{n}^{p',i'}\delta_{n'}^{(n+1)}, & k = 0,\ n = \mathcal{M} \\ \mu_{n'}^{p',i'}, & k = 1,\ n = \mathcal{M} \end{cases}$$

$$\Pr\left(x_{t+1}^{i'} \mid z_{t+1}^{p'}, \epsilon_t^{l}, x_t^{i}, e_t^{k}\right) = \begin{cases} \delta_{i'}^{(i)}, & k = 0,\ l = 0 \\ \frac{A_{ii'}^{p'}}{1 - A_{i,\text{end}}^{p'}}, & k = 1,\ l = 0 \\ \pi_{i'}^{p'}, & k = 1,\ l = 1 \end{cases}$$

and lastly,

$$\Pr\left(z_{t+1}^{p'} \mid z_t^{p}, \epsilon_t^{l}\right) = \begin{cases} \delta_{p'}^{(p)}, & l = 0 \\ A_{pp'}^{*}, & l = 1 \end{cases}$$

Next, the scaling factor at time $t+1$ is computed as:

$$\psi_t = \Pr\left(y_{t+1} \mid y_{1:t}\right) = \sum_{\mathbf{s}} \Pr\left(S_{t+1}^{\mathbf{s}}, y_{t+1} \mid y_{1:t}\right) = \sum_{\mathbf{s}} \ddot{\alpha}_{t+1}\left(\mathbf{s}\right) \tag{5.3}$$

The forward procedure is initialized at time $t = 1$ with $\ddot{\alpha}_1\left(\mathbf{s}\right)$ given by:

$$\begin{aligned} \ddot{\alpha}_1\left(\mathbf{s}\right) &= \Pr\left(S_1^{\mathbf{s}} = \left\{z_1^p, \epsilon_1^l, x_1^i, m_1^n, e_1^k\right\}, y_1\right) \\ &= \Pr\left(\epsilon_1^l \mid z_1^p, x_1^i, e_1^k\right)\Pr\left(e_1^k \mid z_1^p, x_1^i, m_1^n\right)\Pr\left(m_1^n \mid z_1^p, x_1^i\right) \\ &\quad \times \Pr\left(y_1 \mid x_1^i\right)\Pr\left(x_1^i \mid z_1^p\right)\Pr\left(z_1^p\right) \end{aligned}$$

where

$$\begin{aligned} \Pr\left(\epsilon_1^l \mid z_1^p, x_1^i, e_1^k\right) &= \delta_l^{(0)}\delta_k^{(0)} + \left(1 - A_{i,\text{end}}^{p}\right)\delta_l^{(0)}\delta_k^{(1)} + A_{i,\text{end}}^{p}\delta_l^{(1)}\delta_k^{(1)} \\ \Pr\left(e_1^k \mid z_1^p, x_1^i, m_1^n\right) &= \delta_k^{(0)}\delta_n^{(<\mathcal{M})} + \left(1 - \lambda_{\mathcal{M}}^{p,i}\right)\delta_k^{(0)}\delta_n^{(\mathcal{M})} + \lambda_{\mathcal{M}}^{p,i}\delta_k^{(1)}\delta_n^{(\mathcal{M})} \\ \Pr\left(m_1^n \mid z_1^p, x_1^i\right) &= \mu_n^{p,i} \\ \Pr\left(y_1 \mid x_1^i\right) &= B_{y_1|i} \\ \Pr\left(x_1^i \mid z_1^p\right) &= \pi_i^p \\ \Pr\left(z_1^p\right) &= \pi_p^*. \end{aligned}$$

To compute the smoothing distribution we also need the (scaled) backward variable:

$$\tilde{\beta}_t(\mathbf{s}) \triangleq \frac{\Pr\left(y_{t+1:T} \mid S_t^{\mathbf{s}}\right)}{\prod_{\tau=t+1}^{T} \psi_\tau}$$

which is computed backward as:

$$\begin{aligned}
\tilde{\beta}_t(\mathbf{s}) &= \frac{1}{\prod_{\tau=t+1}^{T} \psi_\tau} \sum_{\mathbf{s}'} \Pr\left(S_{t+1}^{\mathbf{s}'}, y_{t+1:T} \mid S_t^{\mathbf{s}}\right) \\
&= \frac{1}{\prod_{\tau=t+1}^{T} \psi_\tau} \sum_{\mathbf{s}'} \Pr\left(y_{t+2:T} \mid S_{t+1}^{\mathbf{s}'}\right) \Pr\left(y_{t+1} \mid S_{t+1}^{\mathbf{s}'}\right) \Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right) \\
&= \frac{1}{\psi_{t+1}} \sum_{\mathbf{s}'} B_{y_{t+1}|i'} \Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right) \tilde{\beta}_{t+1}(\mathbf{s}') \qquad (5.4)
\end{aligned}$$

and the backward recursion starts at time $t = T$ with $\tilde{\beta}_T(\mathbf{s}) = 1$.

Finally, similar to the CxHSMM (section 4.3.4.1), the smoothing distributions follow as:

$$\gamma_t(\mathbf{s}) = \tilde{\alpha}_t(\mathbf{s})\, \tilde{\beta}_t(\mathbf{s}) \qquad (5.5)$$

$$\xi_t(\mathbf{s}, \mathbf{s}') = \frac{\tilde{\alpha}_t(\mathbf{s})\, \tilde{\beta}_{t+1}(\mathbf{s}') B_{y_{t+1}|i'} \Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right)}{\psi_{t+1}} \qquad (5.6)$$

5.4.2 Inference in the presence of missing observations or labeled states

As shown by the clear and shaded nodes in Fig. (5.3), all state variables of the CxSHSMM are hidden, and only the emission symbols are observed. However, to broaden the CxSHSMM's applicability in real-world problems, we consider the two following scenarios: the observations are missing (for instance, there are errors in the captured data), and the exact status of some state variables are given (for example, there are labels returned from additional sensors). We use the same approach as for the CxHSMM in section 4.3.4.2 to tackle these cases.

In the DBN representation, let $\{S_1, \ldots, S_T\}$ be the set of amalgamated states. Each $S_t \triangleq \{z_t, \epsilon_t, x_t, m_t, e_t\}$ consists of all nodes (hidden or observed), except the emission, at a given time. Also, let $\{g_1, \ldots, g_T\}$ be the observation set. In a normal context, $g_t = \{y_t\}$; however if any states, for example z_t is also observed with value

$\bar{z}_t$, then $g_t = \{y_t, \bar{z}_t\}$. Alternatively if nothing is observed at time t, we set $g_t = \{\varnothing\}$.

The inference processes, including recursive computation of the scaled forward, backward variables and derivations of the two smoothing distributions, are carried out analogously to section 5.4.1, with extra care taken to ensure consistency over the observations. Without loss of generality, suppose there are missing observations or labeled states at some time t, consistency over the observations is done by replacing the simple emission probability $\Pr(y_t \mid i)$, whenever it arises, with the conditional probability $\Pr(g_t \mid S_t^{\mathbf{s}})$ during computations. The conditional probability $\Pr(g_t \mid S_t)$ is computed such that the observation g_t must be consistent with the set of states in $S_t^{\mathbf{s}}$:

$$\Pr\left(g_t \mid S_t^{\mathbf{s}} = \left\{z_t^p, \epsilon_t^l, x_t^i, m_t^n, e_t^k\right\}\right) = \begin{cases} B_{y_t|i}, & g_t = \{y_t\} \quad \text{(observing } y_t\text{)} \\ \delta_{\bar{z}_t}^{(p)}, & g_t = \{\bar{z}_t\} \quad \text{(observing } z_t = \bar{z}_t\text{)} \\ \delta_{\bar{\epsilon}_t}^{(l)}, & g_t = \{\bar{\epsilon}_t\} \quad \text{(observing } \epsilon_t = \bar{\epsilon}_t\text{)} \\ \delta_{\bar{x}_t}^{(i)}, & g_t = \{\bar{x}_t\} \quad \text{(observing } x_t = \bar{x}_t\text{)} \\ \delta_{\bar{m}_t}^{(n)}, & g_t = \{\bar{m}_t\} \quad \text{(observing } m_t = \bar{m}_t\text{)} \\ \delta_{\bar{e}_t}^{(k)}, & g_t = \{\bar{e}_t\} \quad \text{(observing } e_t = \bar{e}_t\text{)} \\ 1, & g_t = \{\varnothing\} \quad \text{(missing observation)} \end{cases}$$

Let $h(z)$ be a function such that $h(z) = 1$ if statement z is true, otherwise $h(z) = 0$, the conditional probability $\Pr(g_t \mid S_t^{\mathbf{s}})$ can be expressed in a more compact form as:

$$\Pr(g_t \mid S_t^{\mathbf{s}}) = \left(B_{y_t|i}\right)^{h(y_t \subseteq g_t)} \left(\delta_{\bar{z}_t}^{(p)}\right)^{h(\bar{z}_t \subseteq g_t)} \left(\delta_{\bar{\epsilon}_t}^{(l)}\right)^{h(\bar{\epsilon}_t \subseteq g_t)} \left(\delta_{\bar{x}_t}^{(i)}\right)^{h(\bar{x}_t \subseteq g_t)} \left(\delta_{\bar{m}_t}^{(n)}\right)^{h(\bar{m}_t \subseteq g_t)} \left(\delta_{\bar{e}_t}^{(k)}\right)^{h(\bar{e}_t \subseteq g_t)} \tag{5.7}$$

Eq. (5.7) shows that during inference at time t the emission probability $B_{y_t|i}$ is simply replaced by 1 if the observation is missing, or multiplied by a set of relevant identity functions if the labels of some states are supplied.

5.4.3 Inference with duration models other than Coxian

When the state durations at the bottom level of the SHSMM are modeled by a Multinomial or more generally by a distribution from the Exponential Family, the inference is carried out similarly by forward and backward recursions. Consistent with the absence of $e_{1:T}$ in the DBN representation (Fig. (5.8)), the amalgamated

hidden state now consists of one less variable : $S_t \triangleq \{z_t, \epsilon_t, x_t, m_t\}$, and the set of auxiliary variables are: $\tilde{\alpha}_t(\mathbf{s}) = \Pr(S_t^{\mathbf{s}} \mid y_{1:t})$, $\tilde{\beta}_t(\mathbf{s}) = {}^{\Pr(y_{t+1:T} \mid S_t^{\mathbf{s}})}/_{\prod_{\tau=t+1}^{T} \psi_\tau}$, $\gamma_t(\mathbf{s}) = \Pr(S_t^{\mathbf{s}} \mid y_{1:T})$, and $\xi_t(\mathbf{s}, \mathbf{s}') = \Pr\left(S_t^{\mathbf{s}}, S_{t+1}^{\mathbf{s}'} \mid y_{1:T}\right)$ with scaling factor $\psi_t = \Pr\left(y_t \mid y_{1:t-1}\right)$. The set of Eqs. (5.1), (5.2), (5.3), (5.4), (5.5), and (5.6) of the CxSHSMM inference in section 5.4.1 are all applicable to these variables, except the definition of the transition probability $\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right)$ and the initial value of $\tilde{\alpha}_t(\mathbf{s})$ at time $t = 1$. These differences come with the changes in its DBN representation (Fig. (5.8) compared to Fig. (5.3)). Also, note that for any duration settings the backward recursion is always initialized at time $t = T$ with $\tilde{\beta}_T(\mathbf{s}) = \Pr(\varnothing \mid S_T^{\mathbf{s}}) = 1$ since we can interpret that the probability of observing nothing given something is 1.

The transition probability $\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right)$ is decomposed as follows:

$$\begin{aligned}
\Pr\left(S_{t+1}^{\mathbf{s}'} \mid S_t^{\mathbf{s}}\right) &= \Pr\left(z_{t+1}^{p'}, \epsilon_{t+1}^{l'}, x_{t+1}^{i'}, m_{t+1}^{\tau'} \mid z_t^{p}, \epsilon_t^{l}, x_t^{i}, m_t^{\tau}\right) \\
&= \Pr\left(\epsilon_{t+1}^{l'} \mid z_{t+1}^{p'}, x_{t+1}^{i'}, m_{t+1}^{\tau'}\right) \Pr\left(m_{t+1}^{\tau'} \mid z_{t+1}^{p'}, x_{t+1}^{i'}, m_t^{\tau}\right) \\
&\quad \times \Pr\left(x_{t+1}^{i'} \mid z_{t+1}^{p'}, \epsilon_t^{l}, x_t^{i}, m_t^{\tau}\right) \Pr\left(z_{t+1}^{p'} \mid z_t^{p}, \epsilon_t^{l}\right)
\end{aligned}$$

in which each local probability is defined based on the status of the ending variable ϵ and the duration variable m as detailed below:

$$\Pr\left(\epsilon_{t+1}^{l'} \mid z_{t+1}^{p'}, x_{t+1}^{i'}, m_{t+1}^{\tau'}\right) = \begin{cases} \delta_{l'}^{(0)}, & \tau' > 1 \\ \left(1 - A_{i',\text{end}}^{p'}\right) \delta_{l'}^{(0)} + A_{i',\text{end}}^{p'} \, \delta_{l'}^{(1)}, & \tau' = 1 \end{cases}$$

$$\Pr\left(m_{t+1}^{\tau'} \mid z_{t+1}^{p'}, x_{t+1}^{i'}, m_t^{\tau}\right) = \begin{cases} \delta_{\tau'}^{(\tau-1)}, & \tau > 1 \\ D_{i'}^{p'}(\tau'), & \tau = 1 \end{cases}$$

$$\Pr\left(x_{t+1}^{i'} \mid z_{t+1}^{p'}, \epsilon_t^{l}, x_t^{i}, m_t^{\tau}\right) = \begin{cases} \delta_{i'}^{(i)}, & \tau > 1, l = 0 \\ \frac{A_{ii'}^{p'}}{1 - A_{i,\text{end}}^{p'}}, & \tau = 1, l = 0 \\ \pi_{i'}^{p'}, & \tau = 1, l = 1 \end{cases}$$

$$\Pr\left(z_{t+1}^{p'} \mid z_t^{p}, \epsilon_t^{l}\right) = \begin{cases} \delta_{p'}^{(p)}, & l = 0 \\ A_{pp'}^{*}, & l = 1 \end{cases}$$

Next, we compute the initialization for the forward procedure at time $t = 1$:

$$\begin{aligned}\ddot{\alpha}_1(\mathbf{s}) &= \Pr\left(S_1^{\mathbf{s}} = \left\{z_1^p, \epsilon_1^l, x_1^i, m_1^l\right\}, y_1\right) \\ &= \Pr\left(\epsilon_1^l \mid z_1^p, x_1^i, m_1^\tau\right) \Pr\left(m_1^\tau \mid z_1^p, x_1^i\right) \Pr\left(y_1 \mid x_1^i\right) \Pr\left(x_1^i \mid z_1^p\right) \Pr\left(z_1^p\right) \\ &= \left[\delta_l^{(0)} h(\tau > 1) + \left(1 - A_{i,\text{end}}^p\right) \delta_l^{(0)} \delta_\tau^{(1)} + A_{i,\text{end}}^p \delta_l^{(1)} \delta_\tau^{(1)}\right] D_i^p(\tau) B_{y_1|i} \pi_i^p \pi_p^*\end{aligned}$$

Again, it is important to note that the amalgamated hidden state S_t now has a state space of $(|Q^*|\,|Q|\,M)$, which makes the inference complexity $\left(|Q^*|^2\,|Q|^2\,MT\right)$, significantly larger than $\left(|Q^*|^2\,|Q|^2\,\mathcal{M}T\right)$ of the CxSHSMM as typically $M >> \mathcal{M}$. Thus, when the model becomes more complex (i.e. hierarchical vs. flat HSMM), a greater computational factor is saved by using the Coxian duration model. Lastly, inference in the presence of missing observations and labeled states are handled analogous to the CxSHSMM case.

5.5 Learning

Like the HMM, the set of parameters θ_{CxSHSMM} for the CxSHSMM ties together different parameters of the DBN, and thus can be viewed as a member of the Exponential Family. Consequently, techniques developed for learning with Exponential Family in sections 2.2.1 and 2.2.2 are applicable. We consider four different settings: (i.) all states and emission symbols are fully observed, (ii.) states are hidden, but emission symbols are observed, (iii.) similar to case (ii.) except some emission symbols are missing, and (iv.) similar to case (ii.) except some state labels are supplied.

5.5.1 Maximum Likelihood with fully observed data

To recap, the sufficient statistic is the count of configurations (section 2.2.1.1). Further, since our model is in the DBN form, parameters are tied over time and thus we have to sum the sufficient statistic over time:

$$T\left(\theta_{k,v}^i\right) = \sum_{t=1}^{T} \delta_{X_{i_t}}^{(k)} \delta_{X_{\pi_{i_t}}}^{(v)}$$

in which $T\left(\theta_{k,v}^i\right)$ is the sufficient statistic of the parameter $\theta_{k,v}^i = \Pr\left(X_i = k \mid X_{\pi_i} = v\right)$. It is important to note that the above equation allows us to conveniently derive the

sufficient statistics by simply observing the DBN. Next, the ML solutions are solved locally by using Lagrange multipliers (theorem 2.1).

We first derive the sets of ML solutions for parameters at the top level, consisting of π^*_p and A^*_{pq}. As π^*_p shows the initial conditions of the network, its sufficient statistic arises only at the first time slice:

$$T\left(\pi^*_p\right) = \delta^{(p)}_{z_1} \tag{5.8}$$

thus,

$$\hat{\pi}^*_p = \frac{T\left(\pi^*_p\right)}{\sum_p T\left(\pi^*_p\right)} = \frac{\delta^{(p)}_{z_1}}{\sum_p \delta^{(p)}_{z_1}} = \delta^{(p)}_{z_1}$$

Next, the sufficient statistic for A^*_{pq} is counted every time the top-level state switches from p to q shown by the configuration $\left\{z^q_{t+1}, z^p_t, \epsilon^1_t\right\}$:

$$\begin{aligned} T\left(A^*_{pq}\right) &= \sum_{t=1}^{T-1} \delta^{(q)}_{z_{t+1}} \delta^{(p)}_{z_t} \delta^{(1)}_{\epsilon_t} \\ \Longrightarrow \quad \hat{A}^*_{pq} &= \frac{T\left(A^*_{pq}\right)}{\sum_q T\left(A^*_{pq}\right)} = \frac{\sum_{t=1}^{T-1} \delta^{(q)}_{z_{t+1}} \delta^{(p)}_{z_t} \delta^{(1)}_{\epsilon_t}}{\sum_{t=1}^{T-1} \delta^{(p)}_{z_t} \delta^{(1)}_{\epsilon_t}} \end{aligned} \tag{5.9}$$

Beside, we observe that within each p- initiated semi-Markov chain, the ML estimation process is equivalent to that of a CxHSMM, except that the explicit information about the current parent state is carried along. Therefore, the sets of sufficient statistics for the CxHSMM in section 4.3.5.1 can be reused by adding information on the current parent state, and the status of the ending variables when necessary[5]. For example, as compared with Eq. (4.49): $T\left(A_{ij}\right) = \sum_{t=1}^{T-1} \delta^{(j)}_{x_{t+1}} \delta^{(i)}_{x_t} \delta^{(1)}_{e_t}$ for the CxHSMM, the sufficient statistic $T\left(A^p_{ij}\right)$ of the CxSHSMM has to also carry the information showing that the transition is happening *within* the p-initiated semi-Markov chain by including the status of the parent state (set to p), and the top-level ending variable set to 0, hence,

$$T\left(A^p_{ij}\right) = \sum_{t=1}^{T-1} \delta^{(j)}_{x_{t+1}} \delta^{(i)}_{x_t} \delta^{(p)}_{z_{t+1}} \delta^{(0)}_{\epsilon_t} \delta^{(1)}_{e_t}$$

Sufficient statistics for the rest are listed in Tab. (5.4). The "sequence-ending" parameter $A^p_{i,\text{end}}$ is a new concept not present in the CxHSMM. $A^p_{i,\text{end}}$ is the probability

[5]Alternatively, we can derive the sufficient statistics directly by examining relevant cliques in Figs. (5.4), (5.5), (5.6), and (5.7).

of the p-initiated semi-Markov coming to end. As illustrated in Fig. (5.5)(b), it is mapped to the configurations $\{\epsilon_t^1 \mid z_t^p, x_t^i, e_t^1\}$:

$$T\left(A_{i,\text{end}}^p\right) = \sum_{t=1}^{T} \delta_{\epsilon_t}^{(1)} \delta_{z_t}^{(p)} \delta_{x_t}^{(i)} \delta_{e_t}^{(1)}$$

Given all the sufficient statistics, we can now use Lagrange multiplier to derive the ML solutions for the parameters in the bottom level. For example, the transition probability A_{ij}^p is re-estimated as:

$$\hat{A}_{ij}^p = \frac{T\left(A_{ij}^p\right)}{\sum_j T\left(A_{ij}^p\right) + A_{i,\text{end}}^p}$$

The full ML solution set is presented in Tab. (5.4).

5.5.2 Expectation Maximization with CxSHSMM

The maximum likelihood parameter $\theta^* = \text{argmax}_\theta \Pr(y_{1:T}|\,\theta)$ can be estimated iteratively using the EM algorithm. First, the expected sufficient statistics (ESS's) $\langle T(.)\rangle$ are computed in the E-step taking the expected value of the sufficient statistics $T(.)$ over the probability of hidden states given observed ones. Given the availability of all the sufficient statistics for the CxSHSMM in the previous section (section 5.5.1), and the probability of hidden given observed in our context is $\Pr\left(S_{1:T} \mid y_{1:T}, \theta_{\text{CxSHSMM}}\right)$, the ESS's can be easily computed as:

$$\left\langle T\left(\theta_{k,v}^i\right)\right\rangle = \sum_{S_{1:T}} \Pr\left(S_{1:T} \mid y_{1:T}, \theta_{\text{CxSHSMM}}\right) T\left(\theta_{k,v}^i\right) \tag{5.10}$$

The resulting ESS's then come as marginal probabilities and thus are obtained by marginalizing the one and two time-slices smoothing distributions:

$$\gamma_t(p,l,i,n,k) = \Pr\left(z_t^p, \epsilon_t^l, x_t^i, m_t^n, e_t^k \mid y_{1:T}\right)$$

$$\xi_t(p,l,i,n,k,p',l',i',n',k') = \Pr\left(z_t^p, \epsilon_t^l, x_t^i, m_t^n, \epsilon_t^k, z_{t+1}^{p'}, \epsilon_{t+1}^{l'}, x_{t+1}^{i'}, m_{t+1}^{n'}, e_{t+1}^{k'} \mid y_{1:T}\right)$$

For example, given the sufficient statistic of the Coxian initial phase parameter

Sufficient statistics	
At the top level	
$T\left(\pi^*_p\right)$	$= \delta^{(p)}_{z_1}$
$T\left(A^*_{pq}\right)$	$= \sum_{t=1}^{T-1} \delta^{(q)}_{z_{t+1}} \delta^{(p)}_{z_t} \delta^{(1)}_{\epsilon_t}$
At the bottom level	
$T\left(\pi^p_i\right)$	$= \delta^{(i)}_{x_1} \delta^{(p)}_{z_1} + \sum_{t=0}^{T-1} \delta^{(i)}_{x_{t+1}} \delta^{(p)}_{z_{t+1}} \delta^{(1)}_{\epsilon_t} \delta^{(1)}_{e_t}$
$T\left(A^p_{ij}\right)$	$= \sum_{t=1}^{T-1} \delta^{(j)}_{x_{t+1}} \delta^{(i)}_{x_t} \delta^{(p)}_{z_{t+1}} \delta^{(0)}_{\epsilon_t} \delta^{(1)}_{e_t}$
$T\left(A^p_{i,\text{end}}\right)$	$= \sum_{t=1}^{T} \delta^{(1)}_{\epsilon_t} \delta^{(p)}_{z_t} \delta^{(i)}_{x_t} \delta^{(1)}_{e_t}$
$T\left(\mu^{p,i}_m\right)$	$= \delta^{(n)}_{m_1} \delta^{(p)}_{z_1} \delta^{(i)}_{x_1} + \sum_{t=1}^{T-1} \delta^{(n)}_{m_{t+1}} \delta^{(p)}_{z_{t+1}} \delta^{(i)}_{x_{t+1}} \delta^{(1)}_{e_t}$
$T\left(\lambda^{p,i}_n\right)$	$= \begin{cases} \sum_{t=1}^{T-1} \delta^{(n+1)}_{m_{t+1}} \delta^{(n)}_{m_t} \delta^{(p)}_{z_{t+1}} \delta^{(0)}_{e_t} & n < \mathcal{M} \\ \sum_{t=1}^{T} \delta^{(1)}_{e_t} \delta^{(p)}_{z_t} \delta^{(i)}_{x_t} \delta^{(n)}_{m_t} & n = \mathcal{M} \end{cases}$
$T\left(B_{v\|i}\right)$	$= \sum_{t=1}^{T} \delta^{(v)}_{y_t} \delta^{(i)}_{x_t}$
Re-estimation formulae	
At the top level	
$\hat{\pi}^*_p$	$= T\left(\pi^*_p\right) \Big/ \sum_p T\left(\pi^*_p\right)$
$\hat{A}^*_{pq}$	$= T\left(A^*_{pq}\right) \Big/ \sum_q T\left(A^*_{pq}\right)$
At the bottom level	
$\hat{\pi}^p_i$	$= T\left(\pi^p_i\right) / \sum_i T\left(\pi^p_i\right)$
$\hat{A}^p_{ij}$	$= T\left(A^p_{ij}\right) \Big/ \left[\sum_j T\left(A^p_{ij}\right) + T\left(A^p_{i,\text{end}}\right)\right]$
$A^p_{i,\text{end}}$	$= T\left(A^p_{i,\text{end}}\right) \Big/ \left[\sum_j T\left(A^p_{ij}\right) + T\left(A^p_{i,\text{end}}\right)\right]$
$\hat{\mu}^{p,i}_n$	$= T\left(\mu^{p,i}_n\right) / \sum_n T\left(\mu^{p,i}_n\right)$
$\hat{\lambda}^{p,i}_n$	$= \begin{cases} \frac{T\left(\lambda^{p,i}_n\right)}{\sum_{t=1}^{T-1} \delta^{(n)}_{m_t} \delta^{(p)}_{z_{t+1}} \delta^{(0)}_{e_t}} & n < \mathcal{M} \\ \frac{T\left(\lambda^{p,i}_n\right)}{\sum_{t=1}^{T} \delta^{(p)}_{z_t} \delta^{(i)}_{x_t} \delta^{(n)}_{m_t}} & n = \mathcal{M} \end{cases}$
$\hat{B}_{vi}$	$= T\left(B_{v\|i}\right) \Big/ \sum_v T\left(B_{v\|i}\right)$

Table 5.4: ML solutions for the CxSHSMM when fully observed.

$T\left(\mu_n^{p,i}\right)$ in Tab. (5.4), the ESS follows as:

$$\begin{aligned}
\left\langle T\left(\mu_n^{p,i}\right)\right\rangle &= \sum_{S_{1:T}} \Pr\left(S_{1:T} \mid y_{1:T}\right) T\left(\mu_n^{p,i}\right) \\
&= \sum_{S_{1:T}} \Pr\left(S_{1:T} \mid y_{1:T}\right) \left[\delta_{m_1}^{(n)}\delta_{z_1}^{(p)}\delta_{x_1}^{(i)} + \sum_{t=1}^{T-1} \delta_{m_{t+1}}^{(n)}\delta_{z_{t+1}}^{(p)}\delta_{x_{t+1}}^{(i)}\delta_{e_t}^{(1)}\right] \\
&= \sum_{\{m,z,x,e\}_{1:T}} \Pr\left(\{m,z,x,e\}_{1:T} \mid y_{1:T}\right) \left[\delta_{m_1}^{(n)}\delta_{z_1}^{(p)}\delta_{x_1}^{(i)} + \sum_{t=1}^{T-1} \delta_{m_{t+1}}^{(n)}\delta_{z_{t+1}}^{(p)}\delta_{x_{t+1}}^{(i)}\delta_{e_t}^{(1)}\right] \\
&= \sum_{m_1,z_1,x_1} \Pr\left(m_1,z_1,x_1 \mid y_{1:T}\right) \delta_{m_1}^{(n)}\delta_{z_1}^{(p)}\delta_{x_1}^{(i)} \\
&\quad + \sum_{\{m,z,x\}_{2:T},e_{1:T}} \Pr\left(\{m,z,x\}_{2:T}, e_{1:T} \mid y_{1:T}\right) \sum_{t=1}^{T-1} \delta_{m_{t+1}}^{(n)}\delta_{z_{t+1}}^{(p)}\delta_{x_{t+1}}^{(i)}\delta_{e_t}^{(1)} \\
&= \Pr\left(m_1^n, z_1^p, x_1^i \mid y_{1:T}\right) + \sum_{t=1}^{T-1} \Pr\left(m_{t+1}^n, z_{t+1}^p, x_{t+1}^i, e_t^1 \mid y_{1:T}\right)
\end{aligned}$$

which is easily computed by marginalizing the two smoothing distributions:

$$\left\langle T\left(\mu_n^{p,i}\right)\right\rangle = \sum_{l,k} \gamma_1\left(p,l,i,n,k\right) + \sum_{t=1}^{T-1} \sum_{p',l',i',n',l,k} \xi_t\left(p',l',i',n',1,p,l,i,n,k\right) \quad (5.11)$$

Next, similar to the fully observed case, as a direct result of theorem 2.1 (Lagrange multiplier), in the M-step the estimation solutions are set to the normalized ESS's. For instance, given $\langle T\left(\mu_n^{p,i}\right)\rangle$ in Eq. (5.11), the re-estimated formula for the Coxian initial phase parameter $\mu_n^{p,i}$ is given by:

$$\hat{\mu}_n^{p,i} = \frac{\left\langle T\left(\mu_n^{p,i}\right)\right\rangle}{\sum_n \left\langle T\left(\mu_n^{p,i}\right)\right\rangle}$$

Finally, the full set of re-estimated formulas is shown in Tab. (5.5).

5.5.3 Learning with missing observations or labeled states

The effect of missing observations and the presence of labeled states is taken care of during inference (section 4.3.5.3). Consequently consistency over observations has been ensured during the computation of all ESS's, except for the emission probability $\left\langle T\left(B_{v|i}\right)\right\rangle$ as the observation arises again in a separate term. The M-step is essentially a normalization of ESS's, hence the presence of labeled states or missing

Expected sufficient statistics

At the top level

$$\begin{aligned}
\left\langle T\left(\pi_p^*\right)\right\rangle &= \Pr(z_1^p \mid y_{1:T}) \\
\left\langle T\left(A_{pq}^*\right)\right\rangle &= \textstyle\sum_{t=1}^{T-1} \Pr(z_{t+1}^q, z_t^p, \epsilon_t^1 \mid y_{1:T})
\end{aligned}$$

At the bottom level

$$\begin{aligned}
\langle T\left(\pi_i^p\right)\rangle &= \Pr\left(x_1^i, z_1^p \mid y_{1:T}\right) + \textstyle\sum_{t=1}^{T-1} \Pr(x_{t+1}^i, z_{t+1}^p, \epsilon_t^1, e_t^1 \mid y_{1:T}) \\
\left\langle T\left(A_{ij}^p\right)\right\rangle &= \textstyle\sum_{t=1}^{T-1} \Pr(x_{t+1}^j, x_t^i, z_{t+1}^p, \epsilon_t^0, e_t^1 \mid y_{1:T}) \\
\left\langle T\left(A_{i,\text{end}}^p\right)\right\rangle &= \textstyle\sum_{t=1}^{T-1} \Pr(\epsilon_t^1, x_t^i, z_t^p, e_t^1 \mid y_{1:T}) \\
\langle T\left(\mu_m^{p,i}\right)\rangle &= \Pr\left(m_1^n, z_1^p, x_1^i \mid y_{1:T}\right) + \textstyle\sum_{t=1}^{T-1} \Pr\left(m_{t+1}^n, z_{t+1}^p, x_{t+1}^i, e_t^1 \mid y_{1:T}\right) \\
\langle T\left(\lambda_n^{p,i}\right)\rangle &= \begin{cases} \sum_{t=1}^{T-1} \Pr(m_{t+1}^{n+1}, m_t^n, x_{t+1}^i, z_{t+1}^p, e_t^0 \,|\, y_{1:T}) & n < \mathcal{M} \\ \sum_{t=1}^{T} \Pr(e_t^1, m_t^n, x_t^i, z_t^p \,|\, y_{1:T}) & n = \mathcal{M} \end{cases} \\
\left\langle T\left(B_{v|i}\right)\right\rangle &= \textstyle\sum_{t=1}^{T} \Pr(x_t^i \mid y_{1:T}) \delta_{y_t}^{(v)}
\end{aligned}$$

Re-estimation formulae

At the top level

$$\begin{aligned}
\hat{\pi}_p^* &= \left\langle T\left(\pi_p^*\right)\right\rangle \Big/ \textstyle\sum_p \left\langle T\left(\pi_p^*\right)\right\rangle \\
\hat{A}_{pq}^* &= \left\langle T\left(A_{pq}^*\right)\right\rangle \Big/ \textstyle\sum_q \left\langle T\left(A_{pq}^*\right)\right\rangle
\end{aligned}$$

At the bottom level

$$\begin{aligned}
\hat{\pi}_i^p &= \langle T\left(\pi_i^p\right)\rangle \,/ \textstyle\sum_i \langle T\left(\pi_i^p\right)\rangle \\
\hat{A}_{ij}^p &= \left\langle T\left(A_{ij}^p\right)\right\rangle \Big/ \left[\textstyle\sum_j \left\langle T\left(A_{ij}^p\right)\right\rangle + \left\langle T\left(A_{i,\text{end}}^p\right)\right\rangle\right] \\
A_{i,\text{end}}^p &= \left\langle T\left(A_{i,\text{end}}^p\right)\right\rangle \Big/ \left[\textstyle\sum_j \left\langle T\left(A_{ij}^p\right)\right\rangle + \left\langle T\left(A_{i,\text{end}}^p\right)\right\rangle\right] \\
\hat{\mu}_n^{p,i} &= \langle T\left(\mu_n^{p,i}\right)\rangle \,/ \textstyle\sum_n \langle T\left(\mu_n^{p,i}\right)\rangle \\
\hat{\lambda}_n^{p,i} &= \begin{cases} \frac{\left\langle T\left(\lambda_n^{p,i}\right)\right\rangle}{\sum_{t=1}^{T-1} \Pr(m_t^n, x_{t+1}^i, z_{t+1}^p, e_t^0 \,|\, y_{1:T})} & n < \mathcal{M} \\ \frac{\left\langle T\left(\lambda_n^{p,i}\right)\right\rangle}{\sum_{t=1}^{T} \Pr(m_t^n, x_t^i, z_t^p \,|\, y_{1:T})} & n = \mathcal{M} \end{cases} \\
\hat{B}_{vi} &= \left\langle T\left(B_{v|i}\right)\right\rangle \Big/ \textstyle\sum_v \left\langle T\left(B_{v|i}\right)\right\rangle
\end{aligned}$$

Table 5.5: EM solutions for the CxSHSMM. The ESS's are marginalized from $\gamma_t(.)$ and $\xi_t(.)$.

observations plays no role. Therefore, all re-estimation formulae, apart from $\hat{B}_{v|i}$, for the CxSHSMM in section 5.5.2 remain valid.

The emission probability is re-estimated similar to the CxHSMM case, as shown below:

$$\begin{aligned}\left\langle T\left(B_{v|i}\right)\right\rangle &= \sum_{S_{1:T}}\sum_{t=1}^{T}\delta_{y_t}^{(v)}\delta_{x_t}^{(i)}\Pr\left(S_{1:T} \mid g_{1:T}\right) \\ &= \begin{cases}\sum_{t=1}^{T}\Pr\left(x_t^i \mid g_{1:T}\right)\delta_{y_t}^{(v)}, & y_t \in g_t \\ \sum_{\tau=\{1:T\}\setminus t}\Pr\left(x_\tau^i \mid g_{1:T}\right)\delta_{y_\tau}^{(v)}, & y_t \notin g_t\end{cases}\end{aligned}$$

Then:

$$\hat{B}_{v|i} = \frac{\left\langle T\left(B_{v|i}\right)\right\rangle}{\sum_v \left\langle T\left(B_{v|i}\right)\right\rangle}$$

5.5.4 Learning with duration models other than Coxian

Section 5.3.2 shows that a SHSMM whose state duration at the bottom level is modeled by distributions such as the Multinomial, or more generally the Exponential Family, can be expressed in a generic DBN as shown in Fig. (5.8). Hence, its parameters can be estimated iteratively by the EM algorithm. This section concentrates on estimating the duration parameter $D_i^p(\tau)$ as other parameters are very similar to that of the CxSHSMM. Since a generic DBN is used, the (expected) sufficient statistic computed in the E-step is the same regardless of which distribution from the Exponential Family is used. The sufficient statistic for $D_i^p(\tau) = \Pr\left(m_{t+1}^\tau \mid m_t^1, z_{t+1}^p, x_{t+1}^i\right)$ is collected over the configuration $\left\{m_{t+1}^\tau \mid m_t^1, z_{t+1}^p, x_{t+1}^i\right\}$ in the DBN; hence:

$$T\left(D_i^p(\tau)\right) = \sum_{t=1}^{T-1}\delta_{m_{t+1}}^{(\tau)}\delta_{m_t}^{(1)}\delta_{z_{t+1}}^{(p)}\delta_{x_{t+1}}^{(i)}$$

leading to:

$$\begin{aligned}\left\langle T\left(D_i^p(\tau)\right)\right\rangle &= \sum_{S_{1:T}}\Pr\left(S_{1:T} \mid y_{1:T}\right)T\left(D_i^p(\tau)\right) \\ &= \sum_{t=1}^{T-1}\Pr\left(m_{t+1}^\tau, m_t^1, z_{t+1}^p, x_{t+1}^i \mid y_{1:T}\right)\end{aligned}$$

The next step (M-step), however, depends on the choice of duration distributions. For example, if we work with a Multinomial: $D_i^p \sim Mult\left(D_i^p(1), \ldots, D_i^p(M)\right)$, with

the constraint $\sum_{\tau=1}^{M} D_i^p(\tau) = 1$; then theorem 2.1 can be used to maximize the expected complete log likelihood associated with the duration parameter $\langle \mathcal{L}_D \rangle = \sum_{p,i,\tau} \langle T\left(D_i^p(\tau)\right) \rangle \log \left\{ D_i^p(\tau) \right\}$ resulting in:

$$\hat{D}_i^p(\tau) = \frac{\langle T\left(D_i^p(\tau)\right) \rangle}{\sum_\tau \langle T\left(D_i^p(\tau)\right) \rangle}$$

Other distributions from the Exponential Family may require a bit more effort in the maximization step, and some can be re-estimated only by approximation methods (readers may refer back to section 3.2.2 for optimization examples when distributions such as Poisson and Inverse Gaussian are used in the (flat) HSMM).

5.6 Deep Hierarchical Models

In this thesis we mainly focus on a class of rather "shallow" hierarchical models since they have been adequate for the applications we consider (chapter 6). Also, from existing work in activity recognition and video surveillance (chapter 2), we rarely see a case where the depth of the hierarchical model is more than two or three. This is largely attributed by the overwhelming computational cost when doing inference in "deep" hierarchical models, which is often exponential in depth, and thus hinders applicability in real-world problems. For example, for D-levels Hierarchical Hidden Markov Models, the complexity is $O\left(|Q|^{2D} T\right)$ when the inference process is done as in a standard DBN framework [Murphy and Paskin, 2001]. While being linear in time, it is exponential in D and thus could be a serious bottleneck. Alternatively one can follow a linear-in-D algorithm as in [Fine et al., 1998, Bui et al., 2004, Phung, 2005b] with the complexity of $O\left(D\,|Q|\,T^3\right)$ where the inference follows an inside-outside-style algorithm adopted from the probabilistic context free grammar (PCFG) community. This comes with the cost of cubic time complexity.

Nonetheless, deep hierarchical models seem beneficial and attractive for other domains such as: computer simulation where inference can be sidestepped and the interest is in the random phenomena presented hierarchically, e.g., game simulation; parsing in natural language processing, where the depth can be unbounded; parsing and summarizing web contents where again the depth (e.g., from XML structure) can be high; plan recognition where the deep policy hierarchy is desirable. The inference bottleneck in these works is usually overcome by incorporating further constraints

and information, and often a fixed tree topology is assumed, i.e., segmental information for each state at every level is assumed, or observing more states are observed. Next, we present two ways to extend the CxSHSMM into a deep hierarchy.

The Coxian Hierarchical Hidden Markov/semi-Markov Model

The first approach continues to build Markovian layers on top of the CxSHSMM[6]. States at the production level have theirs own Coxian duration distributions, while states at any upper levels have durations inferred from the lower ones. To be more precise, a parent state p is forced to finish its duration when the p-initiated (semi-) Markov chain in the immediate lower level ends. This structure is named the Coxian Hierarchical Hidden Markov/semi-Markov Model (CxHHMsMM), since the lowest layer is a series of concatenated CxHSMMs, while upper layers (apart from the highest modeling a single HMM) are a series of HMMs. Each CxHSMM at the bottom layer or each HMM at higher layers (except for the top layer) is represented by a state at its immediate upper layer. Fig. (5.9) shows the DBN structure, which is built based on the following network assumptions:

- **At the semi-Markov level ($d = D$):** Since its duration follows a Coxian distribution, a state must carry on to the next time slice if the associated Coxian has not reached its last phase $\mathcal{M}$:

$$m_t < \mathcal{M} \Rightarrow e_t^D = 0 \qquad \text{(assumption 1)}$$

  On the other hand, when the Coxian leaves its last phase and goes to the absorbing state, it signals the end of the current state, opening two possibilities at the next time slice: (i.) the semi-Markov chain carries on to the next time slice ($e_t^{D-1} = 0$), and the current state makes a transition to a new state with probability $\frac{A_{ij}^{D,p}}{1-A_{i,\text{end}}^{D,p}}$, or (ii.) the semi-Markov chain ends with probability $A_{i,\text{end}}^{D,p}$, and a new semi-Markov chain is initiated with probability $\pi_i^{D,p}$.

- **At the Markov levels ($1 \leq d < D$):** A state at level d cannot finish if its child state at the immediate lower level $(d+1)$ is still active:

$$e_t^{d+1} = 0 \;\; \Rightarrow e_t^d = 0 \qquad \text{(assumption 2)}$$

[6]Muncaster and Ma [Muncaster and Ma, 2007] have also discussed extensions to our CxSHSMM in this fashion, however, the author only presented the DBN representation and semantics for a shallow hierarchical, which a two-layer structure, and thus identical to our CxSHSMM.

At level $d = 1$**:**

$$\pi_i^d = \Pr\left(x_t^d = i\right),\ t = 1$$
$$A_{ij}^d = \Pr\left(x_{t+1}^d = j \mid x_t^d = i, e_t^d = 1\right), \forall t \in [1, T-1]$$

At level $2 < d \leq D$**:**

$$\pi_i^{d,p} = \Pr\left(x_{t+1}^d = i \mid x_{t+1}^{d-1} = p, e_t^d = 1, e_t^{d-1} = 1\right), \forall t \in [0, T-1]$$
$$A_{ij}^{d,p} = \Pr\left(x_{t+1}^d = j, e_t^{d-1} = 0 \mid x_t^d = i, x_{t+1}^{d-1} = p, e_t^d = 1\right), \forall t \in [1, T-1]$$
$$A_{i,\text{end}}^{d,p} = \Pr\left(e_t^{d-1} = 1 \mid x_t^{d-1} = p, x_t^d = i, e_t^d = 1\right), \forall t \in [1, T]$$

Also for $d = D$**:**

$$D_i^p = \text{Cox}\left(\boldsymbol{\mu}^{p,i}, \boldsymbol{\lambda}^{p,i}\right)$$
$$\mu_n^{p,i} = \Pr\left(m_{t+1} = n \mid x_{t+1}^{d-1} = p, x_{t+1}^d = i, e_{t+1}^d = 1\right), \forall t \in [0, T-1]$$
$$\lambda_{n<\mathcal{M}}^{p,i} = \Pr\left(m_{t+1} = n+1 \mid m_t = n, x_{t+1}^{d-1} = p, x_{t+1}^d = i, e_t^d = 0\right), \forall t \in [1, T-1]$$
$$\lambda_{\mathcal{M}}^{p,i} = \Pr\left(e_t^d = 1 \mid x_t^{d-1} = p, x_t^d = i, m_t = \mathcal{M}\right), \forall t \in [1, T]$$
$$B_{v|i} = \Pr\left(y_t = v \mid x_t^d = i\right), \forall t \in [1, T]$$

Table 5.6: Mapping from the CxHHMsMM's model parameters to its DBN parameters.

> Alternatively, when $e_t^{d+1} = 1$ there are three (or two if $d = 1$) different scenarios at the next time slice: (i.) $\left\{e_t^d = 0, e_t^{d-1} = 0\right\}$, the current state $x_t^d = i$ proceeds to the next time slice, (ii.) $\left\{e_t^d = 1, e_t^{d-1} = 0\right\}$, state $x_t^d = i$ switches to a new state $x_{t+1}^d = j$ with probability $\frac{A_{ij}^{d,p}}{1 - A_{i,\text{end}}^{d,p}}$, and (iii.) if $d > 1$, $\left\{e_t^d = 1, e_t^{d-1} = 1\right\}$, a new Markov chain is initiated at the next time slice with probability $\pi_i^{d,p}$ for an arbitrarily i.

Based on the above assumptions, the mapping of parameters is easily obtained and the full results are shown in Tab. (5.6). Finally, for this DBN structure standard inference would result in a complexity of $O\left(|Q|^{2D}\,\mathcal{M}^2 T\right)$, which can be reduced to $O\left(|Q|^{2D}\,\mathcal{M} T\right)$ as the phase variable always moves upwards: $m_{t+1} \in [m_t, m_t + 1]$. On the other hand, if the inside-outside-style algorithm is adopted, the inference complexity is $O\left((D - 1 + \mathcal{M})\,|Q|\,T^3\right)$.

The Coxian Hierarchical Hidden Semi-Markov Model

The second approach is to allow state durations to be modeled at each level, and the resulting structure is then called the Coxian Hierarchical Hidden Semi-Markov

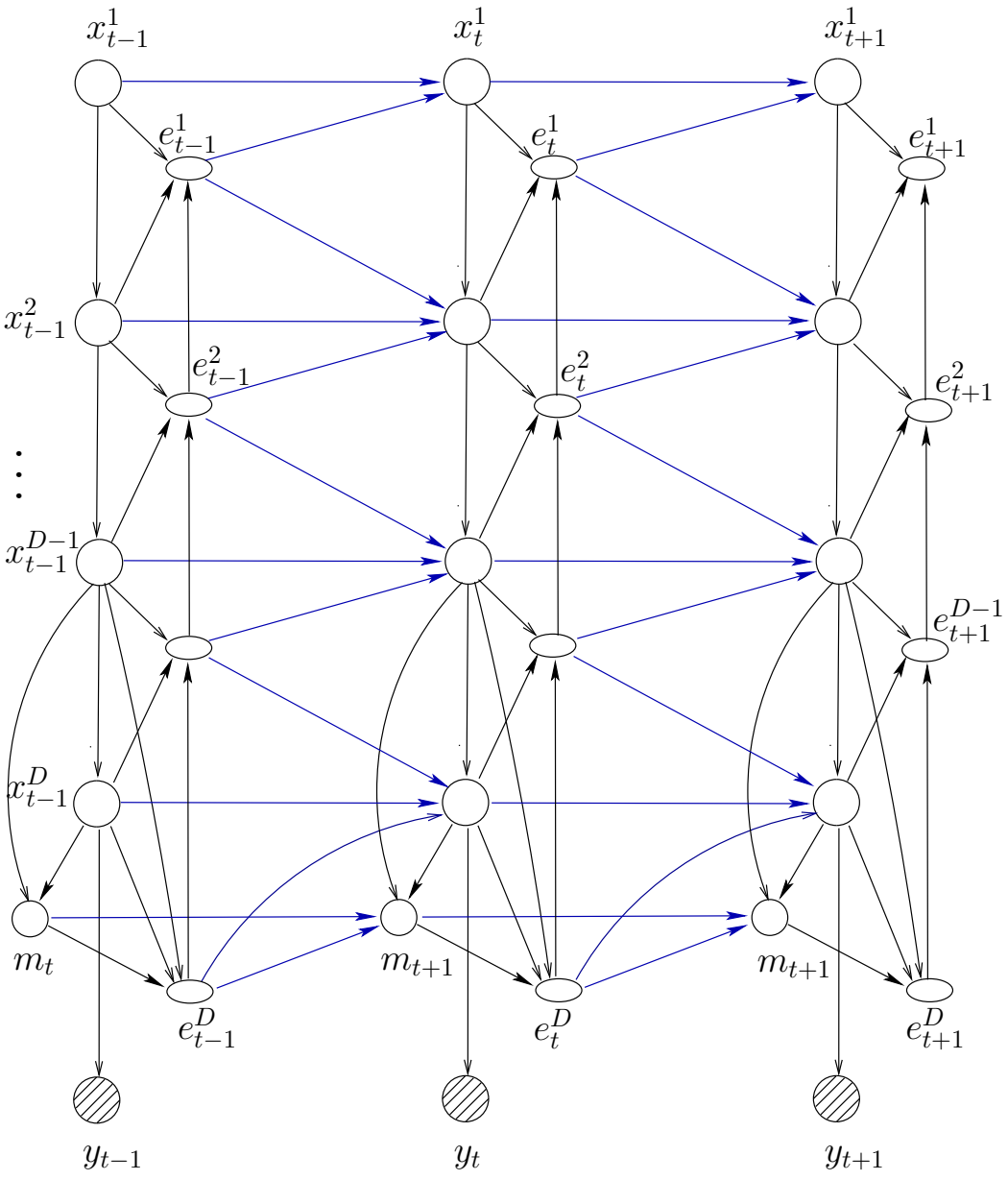


Figure 5.9: The Coxian Hierarchical Hidden Markov/semi-Markov model (CxHHMsMM).

Model (CxHHSMM). To achieve this, the direct link pointing from state variable x_t^d to the ending variable of the immediate upper level e_t^{d-1} (in Fig. (5.9)) is discarded. Consequently, the probability $A_{i,\text{end}}^{d,p}$ of child state i ending (i.e. terminating the p-initiated semi-Markov chain or equivalently ending the parent state p) is dismissed, and any parent state decides on its own ending based on its Coxian duration. Nevertheless, a parent state still cannot end unless its child does so first. The model's DBN representation is shown in Fig. (5.10), and at each layer the following two assumptions must hold:

$$\begin{aligned} m_t^d < \mathcal{M} \quad &\Rightarrow \quad e_t^d = 0,\ \forall d \in [1, D] \\ e_t^{d+1} = 0 \quad &\Rightarrow \quad e_t^d = 0,\ \forall d \in [1, D-1] \end{aligned}$$

Similar to the previous structure, at each time slice, the set of ending variables $e_t^{1:D}$ decide how the next time slice is derived from the current time slice. For a given configuration $\left\{e_t^{1:d} = 1, e_t^{d+1:D} = 0\right\}$, at next time slice $t+1$:

- At any levels $d' \leq d-1$, new semi-Markov chains are initiated.
- At level d, state $x_t^d = i$ makes a transition to a new state $j \neq i$ with probability $A_{ij}^{d,p}$.
- At any level $d' > d$, the same states continue.

The complete list of parameter mappings is shown in Tab. (5.7). It is important to note the difference between the transition probabilities $A_{ij}^{d,p}$ defined in Tab. (5.6) and Tab. (5.7). Also, the expressive power of having duration models at each layer comes with expensive inference complexity. The standard DBN inference becomes $O\left(|Q|^{2D}\,\mathcal{M}^D T\right)$ in comparison with $O\left(|Q|^{2D}\,\mathcal{M} T\right)$ in the previous case. The inside-outside-style inference complexity also increases from $O\left((D-1+\mathcal{M})\,|Q|\,T^3\right)$ in the former case to $O\left(D\mathcal{M}\,|Q|\,T^3\right)$. The choice between the standard inference and the inside-outside-style is application-driven. Standard inference is preferable when the hierarchy is not deep and data sequence is long, while the latter algorithm is more appealing for deep hierarchies with reasonable observation lengths. Nevertheless, both approaches benefit by the Coxian duration parameterization. Existing duration models such as Multinomial and Exponential Family distributions would lead to complexities of $O\left(|Q|^{2D}\,M^D T\right)$ for standard inference and $O\left(DM\,|Q|\,T^3\right)$ for inside-outside-style (in which the maximum duration span M is possibly as large as T, resulting in complexities of $O\left(|Q|^{2D}\,T^{D+1}\right)$ and $O\left(D\,|Q|\,T^4\right)$, respectively).

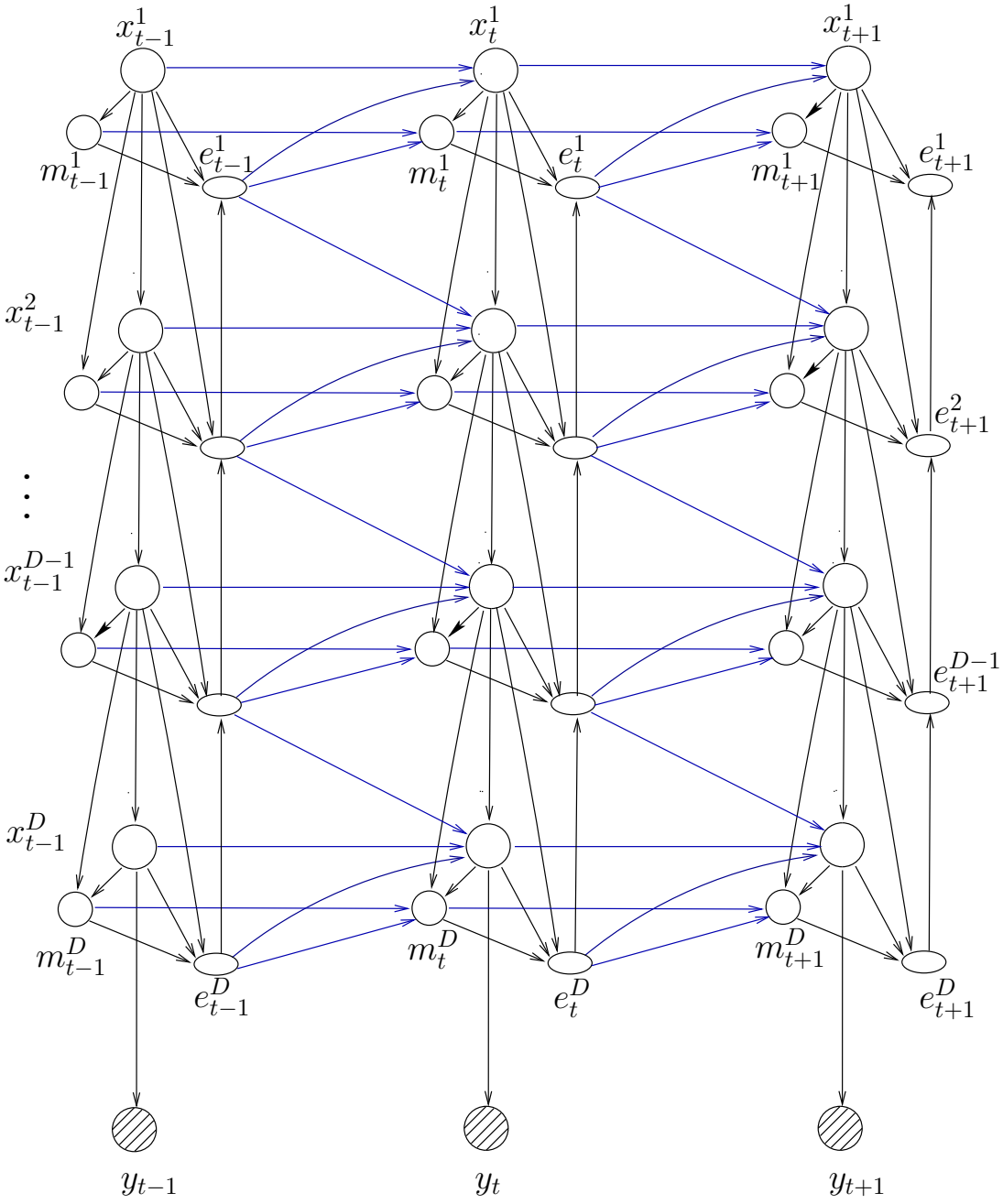


Figure 5.10: The Coxian Hierarchical Hidden Semi-Markov model (CxHHSMM).

At level $d = 1$**:**

$$\pi_i^d = \Pr\left(x_t^d = i\right),\ t = 1$$
$$A_{ij}^d = \Pr\left(x_{t+1}^d = j \mid x_t^d = i, e_t^d = 1\right), \forall t \in [1, T-1]$$

At level $2 < d \leq D$**:**

$$\pi_i^{d,p} = \Pr\left(x_{t+1}^d = i \mid x_{t+1}^{d-1} = p, e_t^d = 1, e_t^{d-1} = 1\right), \forall t \in [0, T-1]$$
$$A_{ij}^{d,p} = \Pr\left(x_{t+1}^d = j \mid x_t^d = i, x_{t+1}^{d-1} = p, e_t^{d-1} = 0, e_t^d = 1\right), \forall t \in [1, T-1]$$

For all $1 \leq d \leq D$

$$D_i^p = \text{Cox}\left(\boldsymbol{\mu}^{p,i}, \boldsymbol{\lambda}^{p,i}\right)$$
$$\mu_n^{p,i} = \Pr\left(m_{t+1} = n \mid x_{t+1}^{d-1} = p, x_{t+1}^d = i, e_{t+1}^d = 1\right), \forall t \in [0, T-1]$$
$$\lambda_{n<\mathcal{M}}^{p,i} = \Pr\left(m_{t+1} = n+1 \mid m_t = n, x_{t+1}^{d-1} = p, x_{t+1}^d = i, e_t^d = 0\right), \forall t \in [1, T-1]$$
$$\lambda_{\mathcal{M}}^{p,i} = \Pr\left(e_t^d = 1 \mid x_t^{d-1} = p, x_t^d = i, m_t = \mathcal{M}\right), \forall t \in [1, T]$$

Emission probability

$$B_{v|i} = \Pr\left(y_t = v \mid x_t^D = i\right), \forall t \in [1, T]$$

Table 5.7: Mapping the CxHHSMM's model parameters to its DBN parameters.

5.7 The SHSMM in literature

This section presents a selective review of the SHSMM [Duong et al., 2005] in current literature. In particular, the expressive power and computational efficiency of the proposed SHSMM and CxSHSMM have been recognized by other researchers. Muncaster and Ma [Muncaster and Ma, 2007] discuss hierarchical extensions to the CxSHSMM and specify necessary constraints in constructing DBN representations for such models. Natarajan and Nevatia [Natarajan and Nevatia, 2007b], on the other hand, aim for applications involving multiple agents. Hence, they argue for the importance of incorporating multi-channel modeling into the HMM. They employ a similar structure as our SHSMM but replace the HSMM at the bottom layer by multi-channel HSMMs to form the Hierarchical Parallel HSMMs, or accept a less rich structure at the ease of having less parameters by moving duration modeling to the top level and using multi-channel HMMs at the bottom level, building the Hierarchical-Semi Parallel HMMs. Nevertheless, both structures do not allow the top-layer states to share child states at the bottom level, thus lessening flexibility. More importantly, state durations are modeled explicitly, causing model complexity

to be dependent on the maximum duration span. Consequently, duration has to be restricted to a certain range to avoid costly computation. Also, Zhu et al. [Zhu et al., 2007] propose an undirected generalization of our SHSMM for the problem of integrated web-page understanding. Their model was an integration of Hierarchical Conditional Random Fields and Semi-Markov Conditional Random Fields for both web-page structure and text content understanding, respectively.

5.8 Closing Remarks

This chapter provides a formulation and thorough analysis of the two-layer CxSHSMM and demonstrates that the model can be extended to hierarchical models with arbitrary depths[7]. With this theoretical analysis, the next chapter presents applications of the CxSHSMM to the problem of classifying, segmenting activities of daily livings (ADLs) and detecting abnormal behaviors, as well as topic transition detection in educational videos.

[7]Note that all the proposed models are able to handle multiple observations, even though it is not discussed in this thesis.

Chapter 6

Applications with Coxian Switching Hidden Semi-Markov Models

The HMM and related models have been applied in human activity recognition and anomaly detection as well as detection of semantic concepts in video data (as shown in chapter 2). These works suffer from two major drawbacks. First, they do not allow automatic modeling of *both* the natural hierarchical decomposition (with sub-structure sharing) and temporal variations in activities and video topics. Second, existing temporal modeling methods are computationally inefficient, making them undeployable in many real-life applications. To overcome these limitations, we have introduced the Coxian duration Switching Hidden Semi-Markov Model (CxSHSMM) in the previous chapter. In this chapter, we detail two main applications of the CxSHSMM: (1) activity recognition and anomaly detection in activities of daily living (ADLs), and (2) topic transition detection in educational videos.

As empirically shown in chapter, temporal information plays an important part in accurate learning and recognition of ADLs. In addition, study in psychology [Zacks and Tversky, 2001] shows that human actions are naturally hierarchical, and this is even more evident in the familiar every day tasks such as "cooking dinner" or "doing dishes". For example, "cooking dinner" would involves a number of related steps such as preparing the ingredients (e.g. cutting, washing, etc.), seasoning, cooking, setting up table, and "doing dishes" would consists of bringing dishes to sink, washing, drying and carrying dishes to cupboard. Also, both activities share the use of the sink, but "doing dishes" would generally occupy the sink only once but

for a long period while "cooking dinner" may need the sink for a number of times and each over a relatively shorter duration. The experiments in this chapter set out to explore deeply into these inherent characteristics of ADLs and our contributions are:

- The first to investigate both hierarchical and duration properties of complex high-level ADLs, which leads to significant improvements in activity recognition performance as compared to using only either hierarchical or duration knowledge.
- A novel scheme to detect anomalies in durations of ADLs – a more subtle form of anomaly – which is practically important in elderly-care and has usually been overlooked in literature.
- An application of the semi-supervised CxSHSMM with limited labeled data to deal with more difficult real world problems. With a small number of labeled data, our framework is shown to cope well with uncertainties and failures in a vision tracking system on a rich class of complex activities.

Next, the topic detection problem for professionally made videos is partially challenging due to the following three reasons, as identified in [Sundaram, 2002]: (i.) the differences in directional styles, (ii.) the semantic relationships of neighbouring scenes, and (iii.) the world knowledge of the viewer. While the last aspect is beyond the scope of this work, the first two clearly imply that effective modeling of high-level semantics requires the domain knowledge (directional style) and the modeling of the complex correlations of the video dynamics (neighboring semantic relationship). The modeling problem, however, is difficult as the underlying semantics naturally possess a hierarchical decomposition with possible existence of tight structure sharing between high-level semantics. In addition, the typical duration for these structures usually varies for each of its higher semantic. This thesis concentrates on the class of education-oriented videos because its hierarchy of semantic structure is more defined, exposing strong temporal correlation in time, and thus make it more desirable to probabilistic modeling, while organization of content in generic videos (e.g. movies) is too diverse to be fully characterized by statistical models. Our contributions are:

- A coherent hierarchical probabilistic framework for topic detection in educational videos, which can be readily applied to other similar video genres such

as news and documentaries.

- The first to investigate and effectively exploit both hierarchical and duration information in video segmentation in a unified framework. Similar to the ADL case, incorporating both duration and hierarchical properties results in superior performance.

The rest of the chapter is organized as follows. In section 6.1.1, we present an application of the CxSHSMM to automatically learn, segment, and recognize complex ADLs, in which the problem of phase number selection for the Coxian is also addressed. Section 6.1.2 uses the activity models learned in section 6.1.1 to construct a scheme to detect any deviation in the durations of unseen ADLs. Next, we tackle activity recognition in more challenging scenarios using partially labeled data in section 6.1.3. We then explore the use of the CxSHSMM in topic transition detection in educational videos in section 6.2. Finally, the chapter concludes in section 6.3.

6.1 Activity Recognition with the SHSMM

In chapter 4 we have experimented with flat models. In this section we set out to tackle more complex and hierarchical data, aiming to recognize and segment complex ADLs at multiple levels as well as to identify abnormal activity segments.

6.1.1 Recognition and Segmentation of Activities in Sequences

Given a morning routine consisting of sequential, but unlabeled and unsegmented ADLs such as "reading morning newspaper", "preparing breakfast", "having breakfast", etc., our objective is to be able to query what the occupant is doing and to detect when she changes activities. The CxSHSMM performance will be compared with that of a Multinomial duration SHSMM (MuSHSMM), a two-layer HHMM, and a flat Multinomial duration HSMM (MuHSMM). Also, we present the results of applying a cross-validated model selection to pick the best number of phases for the Coxian.

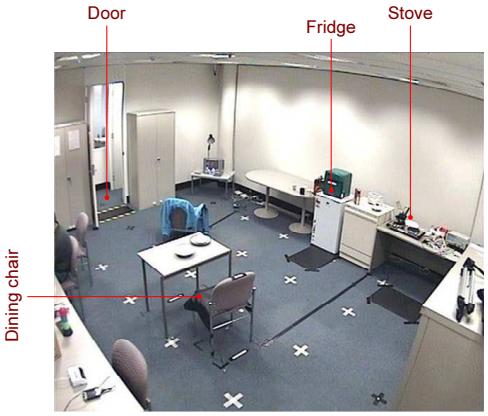
6.1.1.1 Descriptions of High-Level Activities

We consider a typical morning routine consisting of six high-level activities:

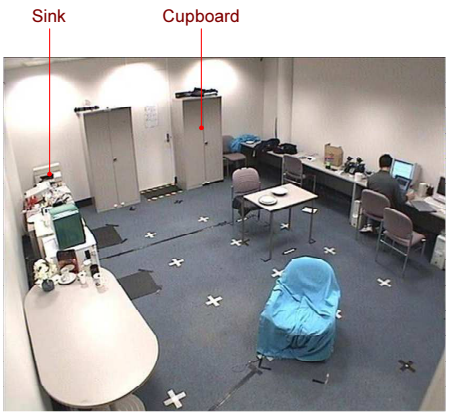
- (**a.1**) "entering the room & making breakfast"
- (**a.2**) "having breakfast"
- (**a.3**) "washing dishes"
- (**a.4**) "making coffee"
- (**a.5**) "reading morning newspaper & having coffee"
- (**a.6**) "leaving the room".

The routine generally follows the sequence "(**a.1**)-(**a.2**)-(**a.3**)-(**a.4**)-(**a.5**)-(**a.6**)" or "(**a.1**)-(**a.2**)-(**a.4**)-(**a.5**)-(**a.3**)-(**a.6**)", depending on whether the person washes the dishes before or after having coffee. The six activities and their typical trajectories are shown in Fig. (6.1). The shaded regular polygons in Fig. (6.1)(c) in the walking path imply that the person does not simply walk past the cell, but actually spends some time in the region (the darker the polygons, the longer the time). For example, in the first activity ("entering the room & making breakfast"), the occupant first walks into the room, then spends some time taking food from the fridge, as indicated by a dark polygon in cell number 13, and later spends more time cooking breakfast at the stove, as illustrated by a darker polygon in cell number 5. Also, some common landmarks are used by different activities for different time spans. For example, both the fridge and the stove are shared between activities (**a.1**) (cooking breakfast) and (**a.4**) (making coffee). Further, except activity (**a.1**) ("entering the room"), all other activities start almost in the same region (i.e. cell number 6 and its neighbors), making the segmentation task more interesting and challenging.

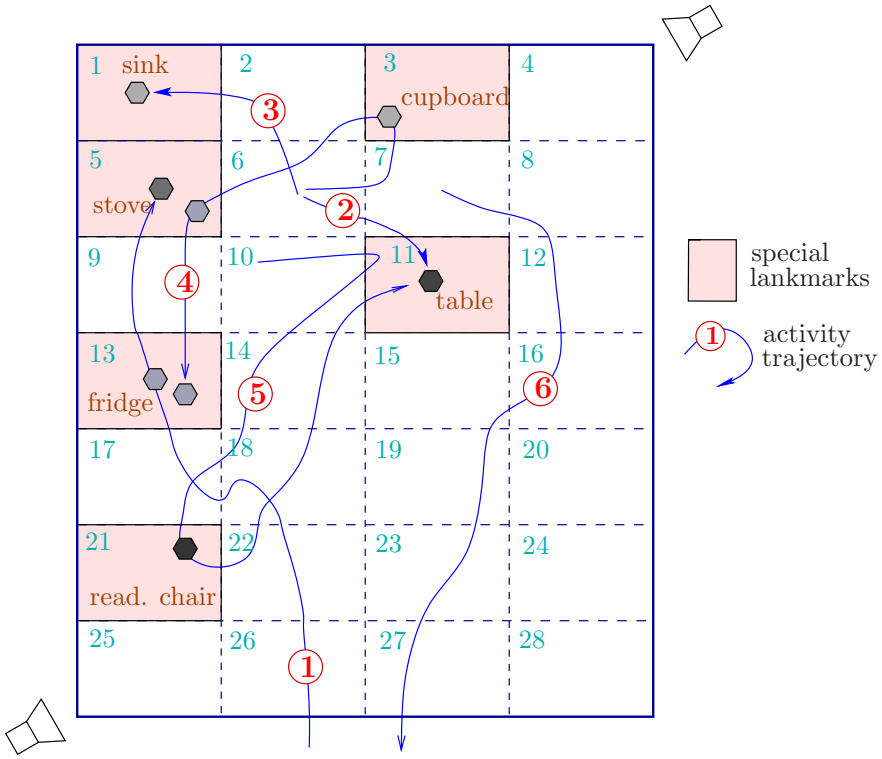
The above morning routine of approximately $130-140$(s) was recorded several times. The length, however, is not the same for all activities. Activity (**a.5**) "reading morning newspaper & having cofee" was the longest (about 35(s)), while activity (**a.6**) "leaving the room" was the shortest (approximately 7(s)). Activities (**a.1**) to (**a.4**) were around $28, 26, 16$ and 20(s), respectively. In each activity, most of the time was usually spent at special landmarks such as the fridge, stove, sink, etc. For instance, in activity (**a.1**), the occupant spends about $5-7$(s) at the fridge, $10-15$(s) at the stove, and the remaining time, around 10(s), was for moving between these designated places. A total of 62 unsegmented sequences of cells are returned from the tracking module (appendix A). Each consists of six activities with a total length



(a) View from camera 1.



(b) View from camera 2.



(c) The morning routine consists of activities (a.1) -> (a.6).

Figure 6.1: Shots of the laboratory kitchen (a-b) and sketches of the activity trajectories (c). The environment in (c) is a quantized version of that in (a) and (b) and each cell is 1m^2.

of around 135 sample points. To ensure an objective evaluation, we construct three different data sets, each consisting of 40 training and 22 testing sequences randomly partitioned from the 62 sequences.

6.1.1.2 Training Assumptions

We train four different kinds of models: a flat MuHSMM, a two-layer HHMM, a MuSHSMM and a number of $\mathcal{M}$-ph.CxSHSMMs with $\mathcal{M}$ ranging from 2 to 7. For the hierarchical models (CxSHSMM, MuSHSMM and HHMM), we set the number of states at the top level equal to the number of activities: $|Q|^* = 6$, and at the bottom level to the number of quantized cells in the kitchen: $|Q| = 28$. More precisely, activities presented by top level states $p \in Q^*$ are high-level activities[1], such as (**a.1**)"entering the room & making breakfast", while activities carried within a designated cell in the kitchen floor, such as "*cooking at stove*" (cell 5), are referred to as atomic activities and presented by bottom level states $i \in Q$. We use the estimated spatial extent of each high-level activity p to define the set of its children $\mathrm{ch}(p)$, as well as the sets of children it is allowed to start with ($\mathrm{chS}(p)$), or end with ($\mathrm{chE}(p)$). For example, activity (**a.1**)"entering the room & making breakfast" (Fig. (6.1)) presumably start in the door region consisting of cell 26 and any of its immediate neighbors: $\mathrm{chS}(1) = [21\ 22\ 23\ 25\ 26\ 27]$; activity (**a.2**)"having breakfast" is carried out within the stove and dining table areas: $\mathrm{ch}(2) = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 14\ 15\ 16]$; and activity (**a.3**)"washing dishes" is assumed to end when the occupant leaves the sink area: $\mathrm{chE}(3) = [1\ 2\ 5\ 6]$. For the MuSHSMM, the maximum duration M is set to 35 time-slices, which is the maximum time span of any individual activity (assumed to be known in advance). The flat HSMM has only a single layer with $|Q| = 28$. The same observation model obtained offline as in [Nguyen et al., 2004] (appendix A) is used. Except for the constraints outlined, all other parameters of these models are initialized randomly or uniformly.

Smoothing the Multinomial duration: A simple moving-average can roughly smooth out the learned Multinomial to avoid the overfitting problem. In addition to the learned MuSHSMM (i.e., the Multinomial duration has not been smoothed), we maintain a smoothed duration version to empirically test the effect of smoothing on the model performance. Henceforth, the learned duration MuSHSMM is denoted as $\widetilde{\mathrm{Mu}}$SHSMM, while its smoothed version as $\overline{\mathrm{Mu}}$SHSMM.

[1]For simplicity, in obvious context we refer high-level activities as only activities.

6.1.1.3 Recognition Results

First, by empirically examining the learned parameters of these models after training, we find that while both the CxSHSMM and the MuSHSMM can capture the patterns in the training data adequately, the two-layer HHMM has failed to do so. The left matrix below shows the transition of the six high-level activities A^*_{pq} obtained from the MuSHSMM (the CxSHSMM yields similar results), while the right matrix is obtained from the two-layer HHMM. While the SHSMM variants have learned reasonable transitions, for example from activities (**a.2**) to (**a.3**) or (**a.4**); from activities (**a.3**) to (**a.4**) or (**a.6**); and from activities (**a.5**) to (**a.3**) or (**a.6**), the HHMM has failed to capture these transitions.

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 & 0.2 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.27 & 0 & 0 & 0.73 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0.88 & 0.01 & 0.01 & 0.1 & 0 \\ 0 & 0 & 0.91 & 0.07 & 0 & 0.02 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0.32 & 0.19 & 0.01 & 0.29 & 0.19 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. (6.2) shows the duration spent at the stove in activity (**a.1**), whose "true" duration is usually centered at 14(s), learned by a 5-ph.CxSHSMM and a MuSHSMM. Both models capture the duration reasonably well. The Coxian model tends to lean to the left as compared to the Multinomial model; however, it does an adequate job at smoothing out the spikes in the Multinomial model. For comparison, we also smooth the Multinomial duration distribution using a simple moving-window averaging method.

Fig. (6.3) demonstrates the ability of the CxSHSMM at accurately capturing different temporal properties of a shared sub-structural unit. The atomic activity "*cooking at stove*" is shared by both activities (**a.1**)"entering the room & making breakfast" and (**a.4**)"making coffee", and the CxSHSMM has successfully learned that "making breakfast" requires more time at the stove than "making coffee".

Next, we compare the performances of the trained models (various $\mathcal{M}$-ph.CxSHSMMs, a MuSHSMM, and a two-layer HHMM) in terms of *segmentation accuracy, early detection* and *running time* on unseen and unsegmented sequences from the three data

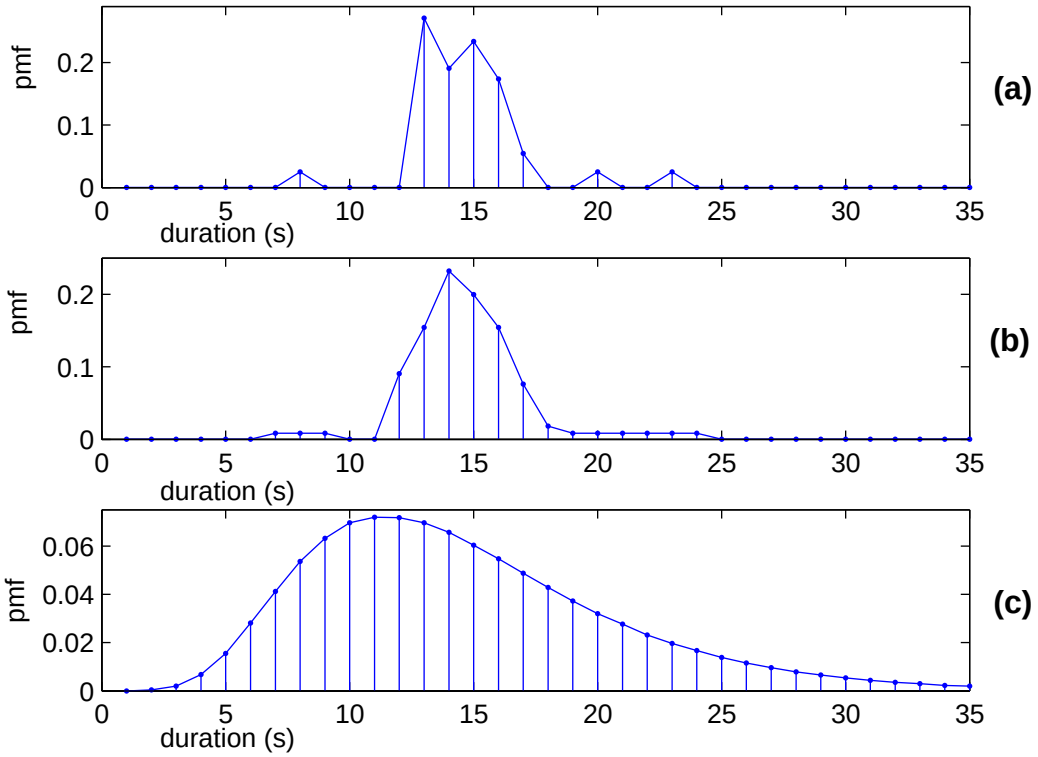


Figure 6.2: Duration "at-stove" learned by **(a)** a $\widetilde{\text{Mu}}$SHSMM, **(b)** a $\overline{\text{Mu}}$SHSMM, and **(c)** a 5-ph.CxSHSMM.

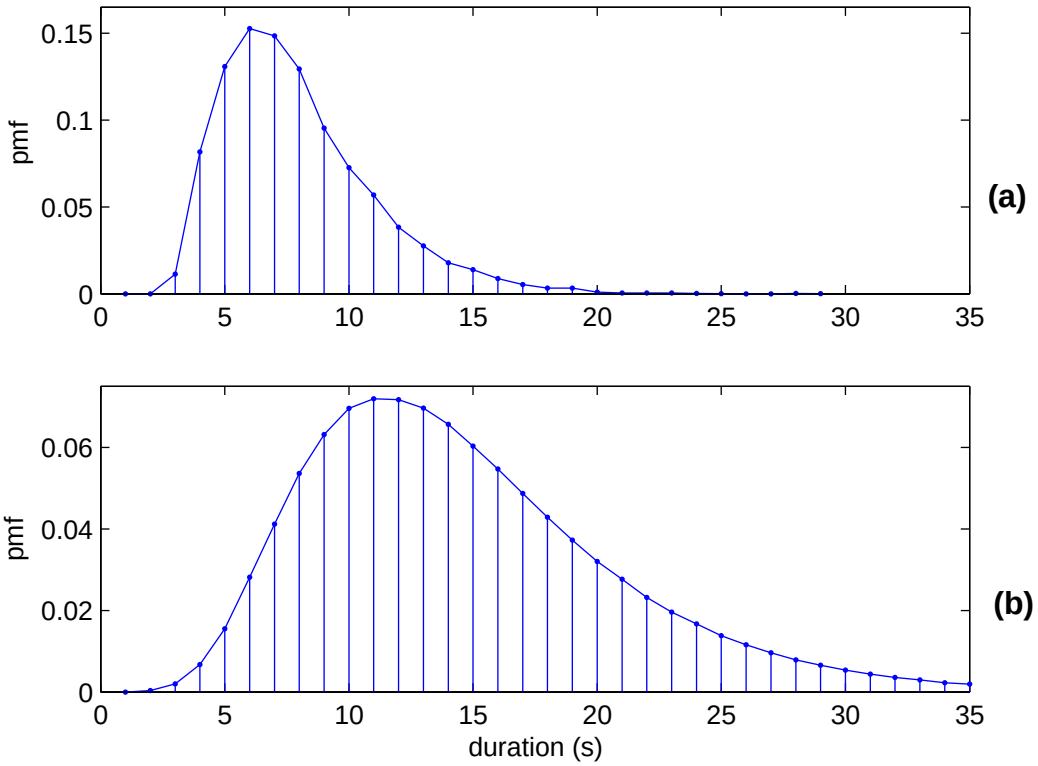


Figure 6.3: Durations "at stove" spent to make coffee **(a)** in activity (**a.4**) and to cook breakfast **(b)** in activity (**a.1**) learned by the 5-ph.CxSHSMM.

Models	**Segmentation accuracy (%) of activities** (a.1) → (a.6)						
	(a.1)	(a.2)	(a.3)	(a.4)	(a.5)	(a.6)	**Avg.**
$\mathcal{M} = 2$	56.06	66.67	80.30	100	93.94	95.45	82.07
$\mathcal{M} = 3$	100	0	100	100	98.48	96.97	82.58
$\mathcal{M} = 4$	0	98.48	100	100	93.94	90.91	80.56
$\mathcal{M} = 5$	100	98.48	100	100	96.97	90.91	97.73
$\mathcal{M} = 6$	100	98.48	100	92.42	100	89.39	96.72
$\mathcal{M} = 7$	100	98.48	100	100	100	87.88	97.73
$\widetilde{\text{MuSHSMM}}$	98.48	98.48	100	100	95.45	65.15	92.93
$\overline{\text{MuSHSMM}}$	98.48	98.48	100	100	100	65.15	93.69
HHMM	19.69	100	100	19.69	77.27	68.18	64.14

Table 6.1: Activity Segmentation on *unseen* data with the CxSHSMMs ($\mathcal{M} = 2, \ldots 7$), the originally learned duration MuSHSMM ($\widetilde{\text{MuSHSMM}}$), the smoothed duration MuSHSMM ($\overline{\text{MuSHSMM}}$), and 2-layer HHMM.

sets[2]. We use the learned models for segmenting and classifying segments of the test sequences into the six high-level activities (the flat HSMM is not included in the test since it cannot model the high-level activities). The filtering distributions of the top-level state given the observation $\Pr(z_t|y_{1:t})$ (Fig. (5.3)) and the most likely label z_t are computed for each time t. The labels z_t at the end of each true segment are used to measure segmentation accuracy. Early detection rate (EDR) is the ratio $t_0/activityLength$ where t_0 is the earliest time from which the activity label z_{t_0} stays accurate.

Tabs. (6.1) and (6.2) present the segmentation and early detection results averaged across the three data sets. Firstly, Tab. (6.1) shows that while the 2-ph.CxSHSMM suffers from low accuracy for the first two activities, the 3-ph.CxSHSMM and the 4-ph.CxSHSMM completely fail to recognize activities (**a.2**), and (**a.1**), respectively. More specifically, plots of online recognition in Fig. (6.4) show that the 2-phase model sometimes segments activity (**a.1**) earlier than its true ending time, while the 4-phase always does so. One possible explanation is that the last two states of activity (**a.1**) (corresponding to cells 9 and 5 in Fig. (6.1)) are also included in the

[2]Our definition of "segmentation accuracy" is somewhat similar to "classification accuracy" in definition 4.3 (section 4.4.2), except that with segmentation we use one hierarchical model instead of a separate number of flat models as in classification case.

Models	Early Detection Rate (%) of activities (a.1) → (a.6)						
	(a.1)	(a.2)	(a.3)	(a.4)	(a.5)	(a.6)	**Avg.**
$\mathcal{M} = 2$	0	0.84	13.44	10.68	14.89	21.39	10.21
$\mathcal{M} = 3$	0	NA	6.97	14.36	4.18	25.93	10.29
$\mathcal{M} = 4$	NA	0	13.95	9.98	1.09	28.66	10.74
$\mathcal{M} = 5$	0	0.41	12.29	10.19	1.23	21.22	7.56
$\mathcal{M} = 6$	0	0.46	12.94	8.88	2.68	29.76	9.12
$\mathcal{M} = 7$	0	0.46	10.84	10.41	2.78	31.77	9.38
$\widetilde{\text{Mu}}$SHSMM	0	0.91	11.88	9.86	2.99	36.04	10.28
$\overline{\text{Mu}}$SHSMM	0	0.60	9.77	9.54	2.86	37.77	10.09

Table 6.2: Early detection rate on *unseen* data with the CxSHSMMs ($\mathcal{M} = 2, \ldots 7$), the originally learned MuSHSMM ($\widetilde{\text{Mu}}$SHSMM), the smoothed duration MuSHSMM ($\overline{\text{Mu}}$SHSMM), and 2-layer HHMM.

starting children set chS(2) of activity 2. As a result, the 2-phase and 4-phase variants may have inaccurately learned that these cells are in activity (**a.2**) even though the probabilities of having these cells as the starting cells for activity (**a.2**) are initialized to low values. Also, the confusion matrix obtained from the 3-ph.CxSHSMM shows all test cases of activity (**a.2**) as having been mistakenly classified as activity (**a.3**). This could be because most of ch(2) are in ch(3), and activities (**a.2**) and (**a.3**) also have the same starting children set, chS(2) ≡ chS(3). The 3-phase Coxian is not strong enough to separate them based mainly on the differences at the ending states chE(2) and chE(3). For $\mathcal{M} \geq 5$, the CxSHSMM yields consistent and adequate segmentation across all activities, and achieves above 96% accuracy, on average. It is, however, observed that the model performance does not continue to increase after $\mathcal{M}$ reaches 5. In fact, $\mathcal{M} = 5$ *achieves the best performance in both segmentation accuracy (*97.73%*) and early detection (*7.56%*)* (Tab. (6.2)). The MuSHSMM, with the disadvantage of having *far more parameters*, now suffers *noticeable degradation* in recognizing activity (**a.6**) (only 65.15% accuracy as compared with 90.91% achieved by the 5-ph.CxSHSMM). Also, smoothing does not help the Multinomial enough to win over the Coxian model. Finally, as expected, the two-layer HHMM, without duration knowledge, has not learned an adequate transition model at the high level (i.e. the transition matrix A^*_{pq}), resulting in its poor and inconsistent performance, i.e. occasionally correctly detecting some activities, such as (**a.2**), (**a.3**), and (**a.5**), while generally failing to detect the others.

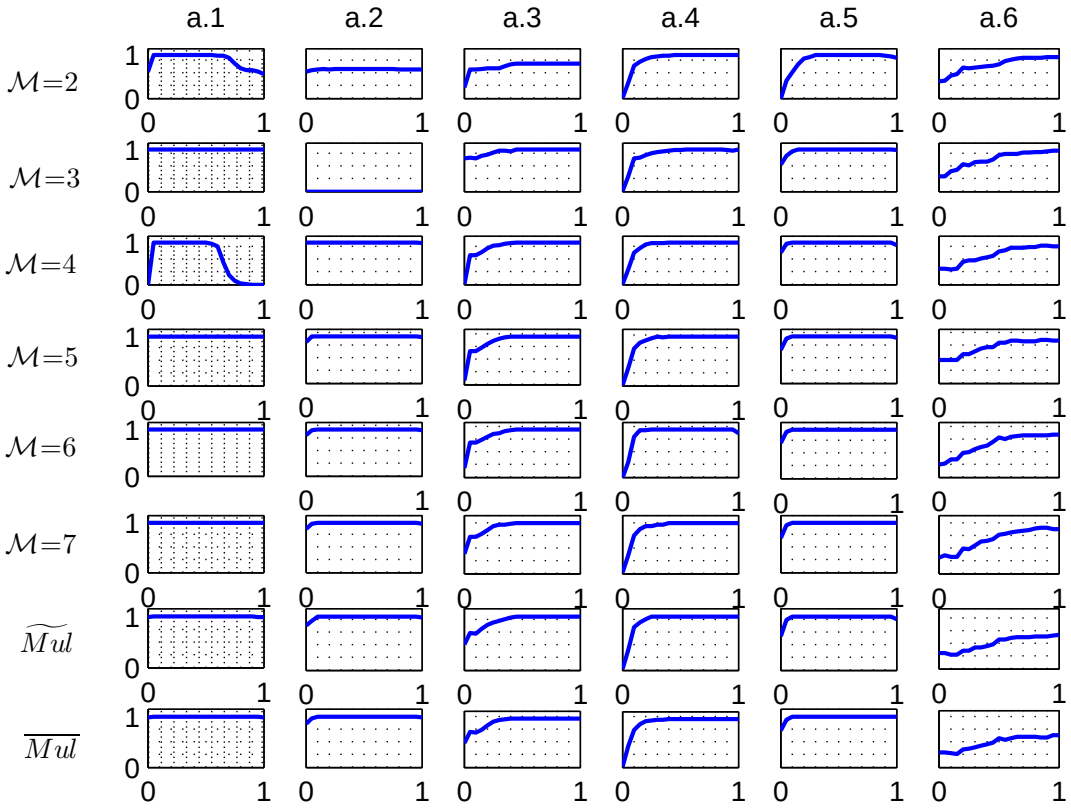


Figure 6.4: Recognition accuracy averaged over three data sets obtained from the CxSHSMMs ($\mathcal{M} = 2, \ldots, 7$), the $\widetilde{\text{Mu}}$SHSMM ($\widetilde{Mul}$) and the $\overline{\text{Mu}}$SHSMM ($\overline{Mul}$). The x axis shows the true segmentation of each activity from the start $\rightarrow$ the end (i.e., $0 \rightarrow 1$). The y axis shows the accuracy rate.

Fig. (6.5) illustrates an example of activity recognition with the 5-ph.CxSHSMM and the two-layer HHMM. While the CxSHSMM sharply segments each activity right at the end of each true segmentation, the HHMM does not have clean cuts between activities and tends to make early transitions (e.g., (**a.1**) $\rightarrow$ (**a.2**) and (**a.4**) $\rightarrow$ (**a.5**)), or wrong transitions (e.g., (**a.5**) to (**a.2**) near to the end of the sequence).

With respect to computational efficiency, Fig. (6.6) shows running time required to complete one EM iteration by a 5-ph.CxSHSMM and a MuSHSMM in our MATLAB implementation tested on ten randomly chosen sequences. The CxHSMM has made the EM process faster by more than a thousand times, which is much larger than

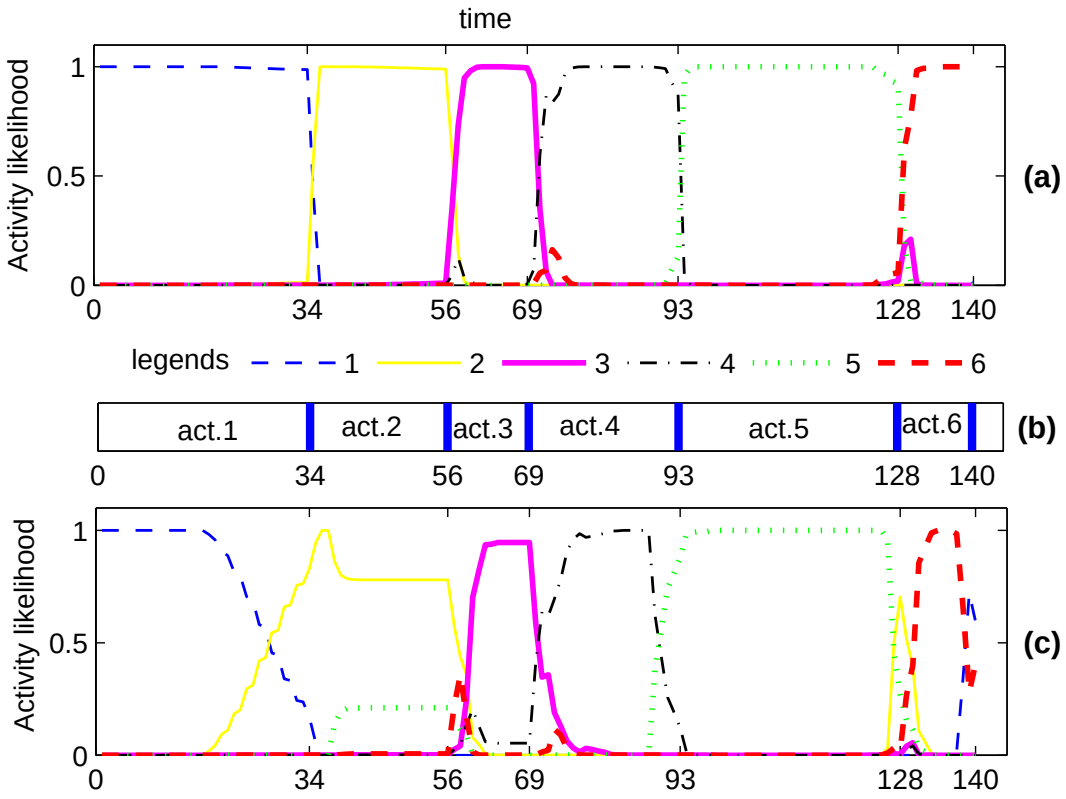


Figure 6.5: Comparing a 5-ph.CxSHSMM in **(a)** with a HHMM in **(c)** at activity recognition for an example sequence, consisting of 6 activities, whose true segmentation is depicted in **(b)**.

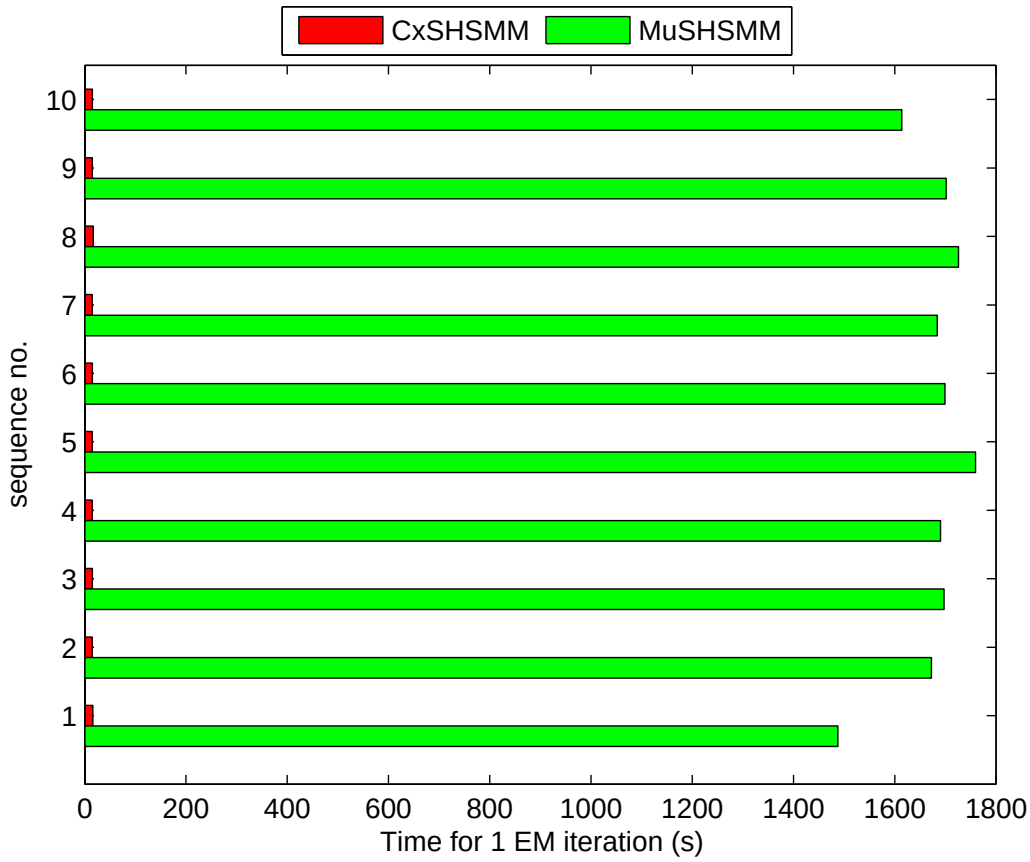


Figure 6.6: Comparison between the 5-ph.CxSHSMM and the MuSHSMM at computation time for 1 EM iteration over 10 randomly chosen sequences.

the theoretical factor defined by $^{M}/_{\mathcal{M}}$. This can be understood as in real implementations the true saving time is also affected by other factors such as the reduction in memory stacks required for the storage and manipulation of variables (e.g. the two time-slices smoothing distributions of the MuSHSMM are very high-dimensional as compared to that of the CxSHSMM). The more data that is required to be stored, the slower the program. Thus, in addition to having a favorable performance result, the Coxian duration modeling has reduced the process of learning and filtering by a significant factor which makes it feasible for real-world applications.

To round up this section, we make four important remarks: **(i.)** Our experiment demonstrates that both duration and hierarchical modeling (with substructure sharing) are vital for learning ADLs and that the SHSMM is a powerful model to deal with this type of data, **(ii.)** The two-layer HHMM has completely failed to tackle data with complicated durations and by having a flat structure the HSMM is unable to perform hierarchical decomposition as expected. **(iii.)** The Coxian duration model is beneficial in that a relatively small phase CxSHSMM ($\mathcal{M} = 5$) needs much fewer parameters, thus requiring significantly less computation time, however, it maintains robustness and outperforms the Multinomial; and **(iv.)** Smoothing the Multinomial duration results in an improvement; nevertheless, this improvement is not substantial.

6.1.2 Detecting Anomalies in Durations of Activities

Anomaly in the duration of activities, if detected, can provide important information in alert systems. For example, in the care of the elderly, a person staying at a location for a longer duration than usual might indicate the onset of an illness. Therefore, given a daily routine consisting of several activities in sequence, our aim is to be able to and learn and query if the occupant is performing her daily patterns normally at each location and detecting any anomaly in activity sequences. For evaluation of anomaly detection, we capture 18 abnormal morning routine sequences (the morning routine is as defined in section 6.1.1.1), which are also unlabeled and unsegmented. In the abnormal data the activity trajectories are kept unchanged with respect to the normal data (i.e., the morning routine in section 6.1.1.1), but the duration spent at each cell has been altered so that a person spends too little or too much time at some locations. We attempt to use the SHSMMs, including the CxSHSMMs and the MuSHSMMs trained in section 6.1.1.2, to serve as models for normal data.

6.1.2.1 The Duration Anomaly Detection Scheme

We implement an online anomaly detection scheme as follows. Suppose that at time t, the online classification algorithm has recognized that p is the winning activity in the period starting from some $t_p \leq t$. The decision to classify p as normal or abnormal is based on examining the likelihood ratio $R_p(t) = \frac{\Pr(y_{t_p:t}|\theta_p)}{\Pr(y_{t_p:t}|\bar{\theta}_p)}$ where θ_p is the parameter of the p-initiated semi-Markov sequence (the learned normal model for p), and $\bar{\theta}_p$ is the abnormal model for p. The abnormal model $\bar{\theta}_p$ is the same as θ_p except for the duration parameter.

For the $\mathcal{M}$-ph.CxSHSMM, the duration parameter $\bar{D}_i^p$ of $\bar{\theta}_p$ is a randomly generated 2-phase Coxian which satisfies: $\text{mean}(\bar{D}_i^p) = \text{mean}(D_i^p) - 0.5M$, if $\text{mean}(D_i^p) > 0.5M$; otherwise $\text{mean}(\bar{D}_i^p) = \text{mean}(D_i^p) + 0.5M$. In other words, we try to "shift" the Coxian towards the less likely part in the duration domain. The 2-phase Coxian is chosen to represent the abnormal data, not only because it involves least computation, but more importantly it is known to have a very high variance [Osogami and Harchol-Balter, 2003] which may suit the variable characteristics of anomalies. For comparison, we also perform anomaly detection with $\bar{D}_i^p$, being a randomly generated $\mathcal{M}$-phase Coxian ($\mathcal{M}$ is the number of phase of D_i^p) whose mean is equal to that of the 2-phase Coxian $\bar{D}_i^p$. These two detection schemes are then compared against the background scheme, where $\bar{D}_i^p$ is a uniform Multinomial distribution. For the MuSHSMM, we intend to set the duration parameter $\bar{D}_i^p$ of $\bar{\theta}_p$ to be either uniform or "inverted", where the "inverted" distribution of $Mult(\mu_n)$ is $Mult(\bar{\mu}_n)$ with $\bar{\mu}_n = \frac{\max(\mu_{1:M}) - \mu_n}{M \times \max(\mu_{1:M}) - \sum_{n=1}^{M} \mu_n} = \frac{\max(\mu_{1:M}) - \mu_n}{M \times \max(\mu_{1:M}) - 1}$.

We argue that the abnormal model $\bar{\theta}_p$, constructed by only changing the duration model, suffices to capture anomalies since our aim is to focus on detecting a more *subtle form of anomaly*, which is the *anomaly only in the state durations* and not in the sequential order. In addition, by automatically constructing a general abnormal model for each normal activity class, our scheme offers two immediate advantages: it saves the network from excessive growth by the need to add new abnormal models in response to unseen data, and it removes the laborious and practically difficult task of manually constructing abnormal models using prior knowledge about the data and speculations on possible abnormal scenarios. Furthermore, by deriving an abnormal model $\bar{\theta}_p$ and taking the likelihood ratios $R_p(t)$, we avoid the unsettling problem of having to normalize the likelihood after setting a threshold because of

the uneven length in observation sequences. Further, we can examine the anomaly for every p-initiated semi-Markov sequence independently instead of considering the whole morning routine of six activities. This is to avoid the residual effects of previous activities in the likelihood, which is especially important in the case where only some activities in the routine are abnormal as it is common that people sometimes deviate from normal behavior in a short time, and then come back to normal. For example, instead of skimming through the morning newspaper in activity (**a.5**)"reading morning newspaper & having coffee", the occupant may want to enjoy a good book, consequently spending more time at the table, before leaving the room (activity (**a.6**)). Thus, by examining every p-initiated semi-Markov sequence independently, our scheme can reset the anomaly measure when the occupant switches to a new activity and conducts it normally. The ability to point out when the behavior has become abnormal or returned to normal is equally important in issuing timely and necessary alerts to carers. To illustrate the capability of our model in solving this difficult problem, some of the 18 abnormal test sequences (each captures the entire morning routine) have only one or two activities containing abnormal durations.

6.1.2.2 Online Segmentation of Activities with Abnormal Durations

We aim to construct different abnormal models for different p-initiated semi-Markov chains. This requires that our detection scheme must first be able to segment the abnormal sequences into different activities. Thus, our model is expected to be robust to temporal disturbance so as to perform adequate online segmentation at the top level, and yet be sensitive enough to detect duration abnormality at the bottom level. In particular, given any morning routine, our objective is to determine if any or all of its comprised activities are abnormal. Our approach involves two steps. First, we use the trained models, including the $\mathcal{M}$-ph.CxSHSMM with $\mathcal{M} = 2, \ldots, 7$ and the MuSHSMM, trained in section 6.1.1 to perform online classification at the top level. As soon as an activity p is identified, we move to the second step, which is to apply our detection scheme that involves only the trained model for the p-initiated semi-Markov chain θ_p and its inverted counterpart $\bar{\theta}_p$, to determine if p is abnormal.

Tab. (6.3) shows the average segmentation results obtained in the first step when testing the set of 18 *abnormal* sequences using the CxSHSMM and the MuSHSMM trained with the three normal data sets in section 6.1.1. Similar to the case of

Models	Segmentation accuracy (%) of activities (a.1) → (a.6)						
	(a.1)	(a.2)	(a.3)	(a.4)	(a.5)	(a.6)	**Avg.**
$\mathcal{M} = 2$	75.93	62.96	77.78	100	100	87.04	83.95
$\mathcal{M} = 3$	100	0	94.44	100	100	92.59	81.17
$\mathcal{M} = 4$	29.63	94.44	87.04	100	100	87.04	83.02
$\mathcal{M} = 5$	100	98.15	83.33	100	100	87.04	94.75
$\mathcal{M} = 6$	100	100	83.33	100	100	85.19	94.75
$\mathcal{M} = 7$	100	100	83.33	100	100	87.04	95.06
$\widetilde{\text{MuSHSMM}}$	100	96.30	77.78	100	100	66.67	90.12
$\overline{\text{MuSHSMM}}$	100	96.30	79.63	100	100	66.67	90.43

Table 6.3: Activity segmentation on *unseen abnormal data* with the CxSHSMMs ($\mathcal{M} = 2, \ldots, 7$), the originally learned duration MuSHSMM ($\widetilde{\text{MuSHSMM}}$), and the smoothed duration MuSHSMM ($\overline{\text{MuSHSMM}}$).

normal data (Tab. (6.1)), all the CxSHSMMs with $\mathcal{M} \leq 4$ fail to segment the activities adequately. The MuSHSMM segments reasonably accurately for activities (**a.1**), (**a.2**), (**a.4**), and (**a.5**), but fails to accurately recognize (**a.6**) one third of the time, and occasionally fails for (**a.3**); hence its segmentation performance cannot be viewed as satisfactory, and thus is not competent for duration anomaly detection (and hence removed in the next step). All the CxSHSMMs with $\mathcal{M} \geq 5$, however, still accurately segment the six activities and are thus the *sole models competent of anomaly detection*. On average, the 7-phase Coxian gives a marginally better segmentation results (95.06%). With respect to EDR (Tab. (6.4)), the 5-phase Coxian offers the smallest EDR upper bound of all activities (31.45% - activity (**a.6**)) while the 6-phase Coxian delivers a slightly better average EDR (12.89%). Hence, we conclude that for $\mathcal{M} \geq 5$-phase, the CxSHSMMs give similar performance. This result is consistent with the normal case in the previous section.

6.1.2.3 Duration Anomaly Detection with CxSHSMM

Our objective is to find the most effective anomaly detection scheme for the CxSHSMMs empirically. The detection effectiveness is measured based on the *true positive* and the *false positive rates*. The true positive rate (TP) is the ratio of the abnormal activities, which are correctly identified as abnormal, to the total abnormal activities tested; while the false positive rate (FP) is the percentage of normal activities,

Models	**Early Detection Rate (%) of activities** (a.1) $\rightarrow$ (a.6)						
	(a.1)	(a.2)	(a.3)	(a.4)	(a.5)	(a.6)	**Avg.**
$\mathcal{M} = 2$	0	30.84	28.84	3.97	14.64	29.64	17.99
$\mathcal{M} = 3$	0	NA	23.04	10.55	6.29	34.15	14.81
$\mathcal{M} = 4$	0	15.98	23.54	6.67	3.46	32.35	13.67
$\mathcal{M} = 5$	0	17.96	19.83	6.74	3.42	31.45	13.23
$\mathcal{M} = 6$	0	14.69	20.31	5.17	2.68	34.49	12.89
$\mathcal{M} = 7$	0	14.18	17.22	7.22	2.99	37.67	13.21
$\widetilde{\text{MuSHSMM}}$	0	13.18	27.44	8.10	5.91	46.41	16.84
$\overline{\text{MuSHSMM}}$	0	12.50	22.18	7.10	4.31	45.68	15.30

Table 6.4: Early detection rate on *unseen normal data* with the CxSHSMMs ($\mathcal{M} = 2, \ldots 7$), the originally learned duration MuSHSMM ($\widetilde{\text{MuSHSMM}}$), and the smoothed duration MuSHSMM ($\overline{\text{MuSHSMM}}$).

which are incorrectly recognized as abnormal, to the total normal activities tested.

Fig. (6.7) presents the Receiver Operating Characteristic (ROC) curves obtained from the three data sets in which the learned normal models used are $\mathcal{M} \geq 5$-ph.CxSHSMMs. The ROC is obtained by varying the threshold for the likelihood ratio $R_p(t)$ with t being set to the true ending time of each activity. All the three data sets yield similar results. The detection scheme using the $\bar{D}_i^p$ as a randomly generated $\mathcal{M}$-phase Coxian with shifted mean (in which $\mathcal{M}$ is the number of phase of D_i^p and $\mathcal{M} = 5, 6, 7$) is the least effective. On the other hand, both the schemes using 2-phase Coxian $\bar{D}_i^p$ and background uniform Multinomial $\bar{D}_i^p$ perform reasonably well as all their ROC curves follow closely the left-hand border and the top border of the ROC spaces. Nevertheless, the inverted 2-phase Coxian model is better as its ROC curves generally rise faster and stay above those of the uniform Multinomial model at the upper left corners of the ROC spaces. The advantages of our proposed inverted 2-phase Coxian model is more evident in Tab. (6.5) as all the top performances highlighted in red fall in the 2-phase Coxian scheme. More specifically, in the region of false alarm not greater than 10% (i.e. FP $\leq 10\%$), the 2$-$phase Coxian $\bar{D}_i^p$ scores best with TP $\sim 77\% \rightarrow 91\%$ compared with TP $\sim 73\% \rightarrow 80\%$ of the background uniform Multinomial. Given that duration is a very subtle form of anomaly, an anomaly detection rate of $77\% \rightarrow 91\%$ is a promising result.

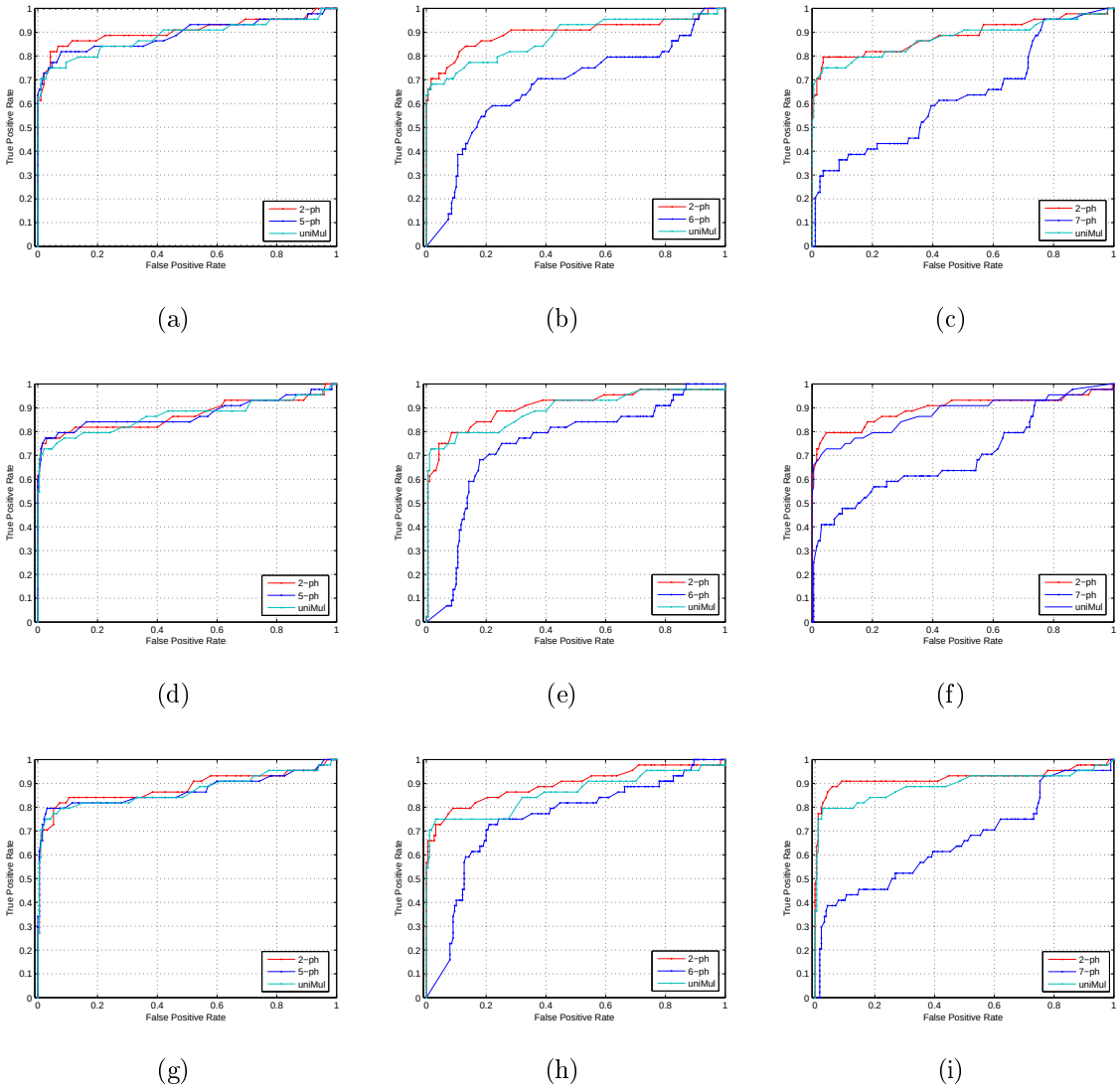


Figure 6.7: Data set 1: plots (a)-(b)-(c), data set 2: plots (d)-(e)-(f), and data set 3: plots (g)-(h)-(i). The ROC curves are obtained from the likelihood ratios set by the learned 5-ph.CxSHSMM - plots (a)-(d)-(g), 6-ph.CxSHSMM - plots (b)-(e)-(h), and 7-ph.CxSHSMM - plots (c)-(f)-(i) and their respective inverted models. The legends "$\mathcal{M}$-ph", and "uniMul" mean the state durations of the inverted models are $\mathcal{M}$-phase Coxian, and uniform Multinomial distributions, respectively.

	Best True Positive Rates (%)								
Learned models	5-ph.CxSHSMM			6-ph.CxSHSMM			7-ph.CxSHSMM		
Inverted models	2-ph.	5-ph.	Mul.	2-ph.	6-ph.	Mul.	2-ph.	7-ph.	Mul.
Data set 1	84.09	81.82	77.27	77.27	25.00	72.73	79.55	36.36	75.00
Data set 2	79.55	79.55	77.27	79.55	15.91	75.00	79.55	45.45	72.73
Data set 3	81.82	79.55	79.55	79.55	38.64	75.00	90.91	40.91	79.55

Table 6.5: Best TPs selected from the ROCs in the region of "FP $\leq$ 10%". For each data set and each learned model, the highest TPs crossed three inverted models are highlighted in red.

6.1.2.4 SHSMM vs. HSMM

We also compare the use of the SHSMM versus a flat HSMM in anomaly detection. Since the HSMM cannot segment the sequence into the six activities, it learns only a normal duration model at each cell location for the entire morning routine. This makes the HSMM less flexible and unable to isolate the abnormal segments in a sequence. Fig. (6.8) shows an example of a sequence comprising activities in order (**a.1**) $\rightarrow$ (**a.6**), in which the first two activities (**a.1**) and (**a.2**) are abnormal, while the rest ((**a.3**) $\rightarrow$ (**a.6**)) are normal. While the 5-ph.CxSHSMM successfully deals with this scenario by pointing out only the first two activities are abnormal, the HSMM continues to label the sequence as abnormal until the sequence is about to end. The ability of the SHSMM to recognize early that activities have returned to normal is practically important in the context of monitoring ADLs in a smart home (e.g. for the aged) as it prevents unnecessary alerts.

In short, this section demonstrates other advantages of the CxSHSMM. First, by accurately learning normal temporal patterns, it is capable of detecting anomalies in activity durations. Second, by having a hierarchical structure with substructure sharing, it is able to recognize and isolate abnormal segments in the activity sequence. Further, the experiment has verified the effectiveness of our proposed anomaly detection scheme, especially the use of "inverted" 2-phase Coxian models to capture abnormal durations.

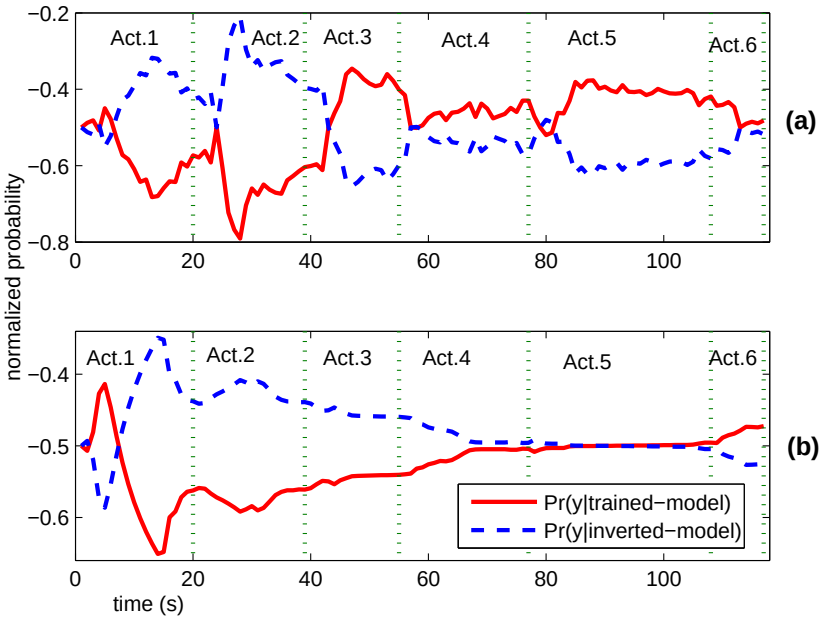


Figure 6.8: Anomaly detection with **(a)** the 5-ph.CxSHSMM and its inverted 2-phase duration model, and **(b)** the flat HSMM and its inverted duration model.

6.1.3 Improvement in Activity Recognition and Segmentation with Partially Labeled Data

In previous sections, care has been taken in the course of data capture such that missing trajectories are minimized. In this section we move to tackle noisy data in which the occupant is free to move or sit where he prefers, which includes sitting behind tables (and thus occluded), staying still for long periods on the sofa (and thus getting confused with the background as the tracking algorithm tends to classify still objects as belonging to the background), and occasionally moving fast between places (cameras may fail to track) or out of camera range. This leads to a significant portion (35.34% on the average) of the tracks being lost. Further, we aim for a more realistic scenario in which the trajectories of high-level activities overlap considerably (more as compared to those in section 6.1.1), and for some activities, they are totally overlapping. Thus, although our objectives remain as in section 6.1.1 (i.e., classifying and segmenting high-level ADLs), we set out to explore the more challenging task of handling noisy tracking data and overlapping activities. We again employ our CxSHSMM and compare it with the standard two-layer HHMM at recognizing and

segmenting high-level activities. The MuSHSMM is not included here because of its costly computation. Furthermore, we use partially labeled data to assist the learning phase in dealing with the catastrophic nature of our data set.

6.1.3.1 Data Descriptions

In this experiment we use the same environment as in previous sections (Fig. (6.1)(a,b)) and capture an evening routine consisting of seven high-level activities:

- (**a.1**) "walking into kitchen & taking food out for cooking"
- (**a.2**) "cooking dinner"
- (**a.3**) "having dinner"
- (**a.4**) "relaxing on sofa & watching tv"
- (**a.5**) "cleaning stove"
- (**a.6**) "sweeping floor"
- (**a.7**) "emptying bin"

The occupant does not strictly follow the sequential order from activity (**a.1**) to (**a.7**), but occasionally makes a deviation, such as choosing to clean the stove (**a.5**) before/after watching television (**a.4**). Firstly, the segmentation tasks at high-level activities is challenging because the time slots are not distributed fairly among activities, for instance emptying the bin takes noticeably less time than sweeping the floor or watching television, and thus is possibly overlooked by the model. Another challenge is due to the limitation of the tracking module which occasionally loses the path. Every sequence (including unseen test sequences) suffers from missing observations ranging from $\sim$ 22% to $\sim$ 44%, except for two sequences: one has only 14% missing trajectories, while another fully loses 60% of its track. Fig. (6.9) shows an example of trajectories returned by the tracking system in which missing observations are shown by the discontinuities in the plot. Finally, a total of 63 sequences are captured, of which 39 (accounting for about 60%) are used for training, and the remaining 24 sequences are for testing.

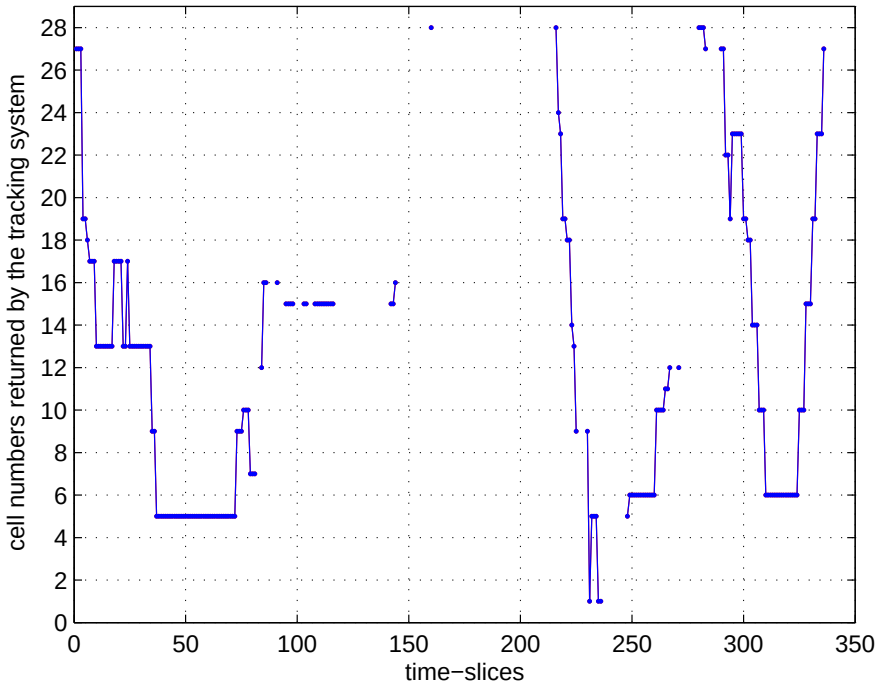


Figure 6.9: An example of sequence of the designated cells, which were sequentially visited by the occupant, returned by the tracking system. The discontinuities in the graph show missing observations.

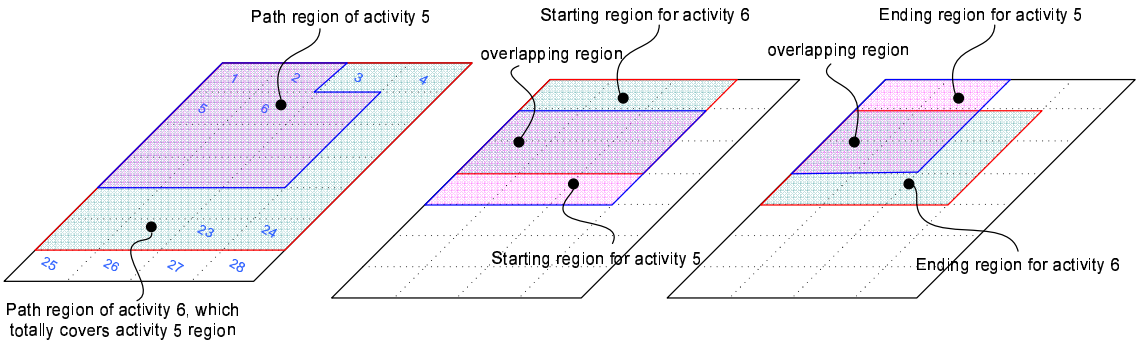


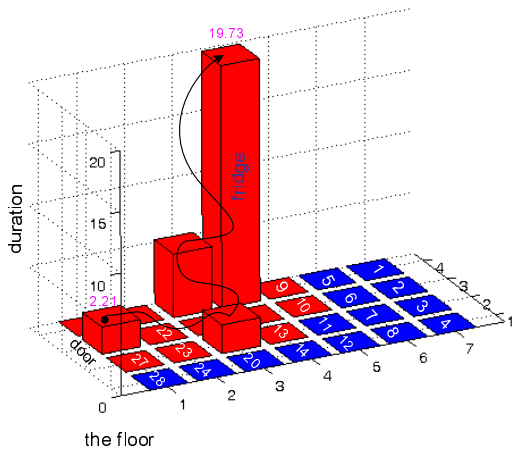
Figure 6.10: Illustrations for path, starting, and ending regions for activity (a.5)'cleaning-stove' and (a.6)'sweeping-floor' .

6.1.3.2 Training Assumptions

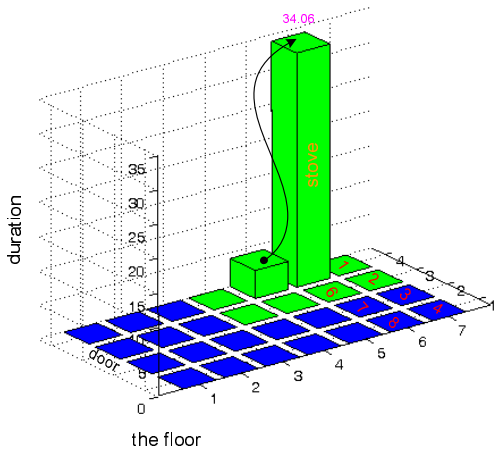
We run five different cases. In the first case, training data (39 sequences) are unlabeled. In the next four cases, each sequence in the training set is randomly labeled at rates of 1%, 4%, 8% and 16%, respectively. In each case, we employ the CxSHSMM to learn the training data, and then perform activity classification and segmentation on unseen and unlabeled test data, and compare its performance with the two-layer HHMM. Again we run the tests on different $\mathcal{M}-$phase CxSHSMMs (for $\mathcal{M} \in [2, 10]$) for phase selection. Similar to section 6.1.1, for both the CxSHSMMs and HHMM we set the number of parent states at the top level to the number of high-level activities $|Q^*| = 7$, and the number of children states at the bottom level is mapped to the number of quantized cells in the kitchen floor $|Q| = 28$. The children set $\mathrm{ch}(p)$, the starting children set $\mathrm{chS}(p)$, and the ending children set $\mathrm{chE}(p)$, for $p \in Q^*$, are then defined by our prior knowledge of the activities (i.e., these sets contain the kitchen regions where corresponding activities may occur). There are significant overlaps between these sets for different p. For instance, Fig. (6.10) shows the estimated spatial extents of activities (**a.5**)"cleaning stove" and (**a.6**)" sweeping floor". We observe that $\mathrm{ch}(5) \subset \mathrm{ch}(6)$ as "cleaning stove" concentrates only around the stove area while "sweeping floor" is done on the whole floor. There are also major overlaps between $\mathrm{chS}(5)$ and $\mathrm{chS}(6)$, and between $\mathrm{chE}(5)$and $\mathrm{chE}(6)$ as sweeping starts and ends around the stove area.

6.1.3.3 Decoding learned sequences of activities

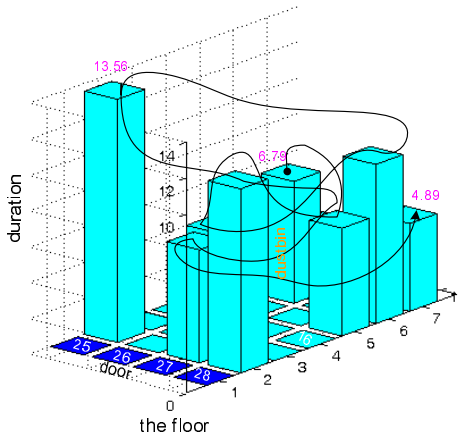
Fig. (6.11) shows the most likely trajectories obtained from the 6-ph.CxSHSMM trained with 8% labeled data. As the observation matrix B, which is not updated



(a) Activity 1: entering-room-taking-food-out.



(b) Activity 2: cooking-dinner.



(c) Activity 6: sweeping-floor.

Figure 6.11: The most likely trajectories learned by the 6-ph.CxSHSMM using 8% labeled data. The red, green and cyan cells mark the child sets (ch(p)) of activity (**a.1**), activity (**a.2**) and activity (**a.6**), respectively; whereas the blue cells show states not belonging to the child sets of the corresponding activities.

during learning, has non-zero entries mainly in its diagonal (i.e., the floor cell numbers returned by the tracking module are mostly corresponding to the respective state numbers at the bottom level), it can be stated that the bottom state sequences decoded generally can be directly mapped into the floor cell numbers, and they more or less show the occupant's walking paths. Thus, the walking path (wp) of an activity p can be inferred from the learned parameters π_i^p and A_{ij}^p , while its state durations are the Coxian means computed from the learned parameter $D_i^p = \mathrm{Cox}\,(\mu^{p,i}, \lambda^{p,i})$.

$$\mathrm{wp} \;=\; \{i^*_{1:N}\} \qquad \text{where} \qquad i^*_n = \begin{cases} \underset{i}{\mathrm{argmax}}\,\{\pi_i^p\} & n = 1 \\ \underset{i}{\mathrm{argmax}}\left\{A^p_{i^*_{n-1}i}\right\} & n = 2, \ldots, N \end{cases}$$
$$A^p_{i^*_N,\mathrm{end}} = \max_i \left\{A^p_{i^*_N i}\right\}$$

Despite the fact that the training data suffers a large portion ($\sim$ 35%) of missing observation, and the activities are quite complex with overlap trajectories (e.g., activity (**a.2**) is totally overlapped by activity (**a.6**), and partially overlapped by activity (**a.1**)), Fig. (6.11) shows that the CxSHSMM has correctly identified not only the most likely cells that would be visited in each activity but also their visiting durations. In both activities (**a.1**)"walking into kitchen & taking food out for cooking" and (**a.2**)"cooking dinner", the model accurately learns that the occupant spends significant time at predesignated locations to complete some specific tasks, i.e., stopping at the fridge (19.73 time units) to take out food in (**a.1**) or standing at the stove (34.08 time units) to cook dinner in (**a.2**); whereas the trajectory scatters in (**a.6**)"sweeping floor" without notable durations at any particular area.

6.1.3.4 Recognition Results with Unlabeled and Partially Labeled Data

We compare the performance of various $\mathcal{M}$-ph.CxSHSMMs and the standard HHMM on *segmentation accuracy,* and *early detection.* The segmentation accuracy and early detection rates are defined as in section 6.1.1.3. We train the CxSHSMMs and the HHMM on unlabeled data as well as on 1% to 16% labeled data and test them on unseen, unsegmented, and unlabeled data that contains 36.30% missing trajectories on the average. Tab. (6.6) shows that without labeling, even though the 3-ph.CxSHSMM significantly outperforms the HHMM (49% accuracy vs. 29%), it

Trained with unlabeled data

HHMM (Avg. 29.17%)

$$\begin{bmatrix} 25.0 & 0 & 0 & 0 & 75.0 & 0 & 0 \\ 0 & 12.5 & 0 & 87.5 & 0 & 0 & 0 \\ 0 & 0 & 4.2 & 95.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 12.5 & 0 & 87.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 37.5 & 0 & 0 & 62.5 \end{bmatrix}$$

3−phase CxSHSMM (Avg. 49.40%)

$$\begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8.3 & 79.2 & 8.3 & 0 & 0 & 4.2 & 0 \\ 0 & 0 & 8.3 & 79.2 & 0 & 12.5 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 4.2 & 16.7 & 8.3 & 66.7 & 0 & 4.2 & 0 \\ 4.2 & 0 & 0 & 95.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 41.2 & 0 & 0 & 58.3 \end{bmatrix}$$

Trained with 1% labeled data

HHMM (Avg. 31.55%)

$$\begin{bmatrix} 25.0 & 0 & 0 & 0 & 75.0 & 0 & 0 \\ 0 & 37.5 & 0 & 62.5 & 0 & 0 & 0 \\ 0 & 0 & 29.2 & 70.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 16.7 & 4.2 & 79.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 4.2 & 0 & 0 & 29.2 & 0 & 37.5 & 29.2 \end{bmatrix}$$

3−phase CxSHSMM (Avg. 73.81%)

$$\begin{bmatrix} 95.8 & 4.1667 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 45.8 & 54.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 95.8 & 0 & 4.2 & 0 \\ 0 & 8.3 & 58.3 & 20.8 & 12.5 & 0 & 0 \\ 0 & 0 & 0 & 29.2 & 0 & 70.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.2 & 95.8 \end{bmatrix}$$

Table 6.6: Confusion matrices showing the segmentation accuracy of the 7 activities.

does not deliver a satisfactory performance with this catastrophic data. However, when we supply labels, as little as 1%, the 3-ph.CxSHSMM dramatically increases its accuracy to 73% as compared with a modest rise of only 2%, from 29% to 31%, for the HHMM.

The significant advantages of the CxSHSMM over the HHMM are confirmed by the full results in Tabs. (6.7) - (6.8) and by the average results illustrated in Fig. (6.12). In particular, Fig. (6.12) shows that the HHMM ($\mathcal{M} = 1$) stays at only around 60% segmentation accuracy even though we supply it with data being labeled from 4% up to 16%. On the contrary, with 4% labels and above, as we add in more geometric phases into the state durations ($\mathcal{M} = 2,\ 3, \ldots$) the CxSHSMMs continue to improve their performance until stabilizing around 90% for $\mathcal{M} \geq 4$.

With only 1% labels, some CxSHSMMs, such as $\mathcal{M} \in [4,\ 5,\ 6,\ 9,\ 10]$, perform reasonably well with an average of around 80% (Fig. (6.12)); nevertheless they occasionally fail to recognize some activities as illustrated by their worst performance in Fig. (6.14). For example, with 1% labels (Tab. (6.7)) the 4-ph.CxSHSMM has an average segmentation accuracy of 79.76%, but it often mislabels activity (**a.3**) (41.67%

1% **labeled data used in training**								
Models	Segmentation accuracy (%) of activities (a.1) → (a.7)							
	(a.1)	(a.2)	(a.3)	(a.4)	(a.5)	(a.6)	(a.7)	*Avg.*
HHMM	25.00	37.50	29.17	100	0	0	29.17	31.55
$\mathcal{M} = 2$	100	87.50	58.33	100	0	33.33	50.00	61.31
$\mathcal{M} = 3$	95.83	100	45.83	95.83	12.50	70.83	95.83	73.81
$\mathcal{M} = 4$	100	91.67	41.67	95.83	41.67	91.67	95.83	79.76
$\mathcal{M} = 5$	100	100	75.00	100	0	83.33	91.67	78.57
$\mathcal{M} = 6$	100	100	58.33	83.33	58.33	79.17	83.33	80.36
$\mathcal{M} = 7$	100	95.83	58.33	95.83	0	50.00	83.33	69.05
$\mathcal{M} = 8$	100	95.83	33.33	100	66.67	8.33	58.33	66.07
$\mathcal{M} = 9$	100	100	70.83	95.83	79.17	75.00	75.00	85.12
$\mathcal{M} = 10$	100	100	79.17	100	8.33	100	66.67	79.17
4% **labeled data used in training**								
HHMM	93.83	100	45.83	95.83	4.17	20.83	54.17	59.52
$\mathcal{M} = 2$	100	100	66.67	100	70.83	95.83	79.17	87.50
$\mathcal{M} = 3$	95.83	100	91.67	100	66.67	95.83	95.83	92.26
$\mathcal{M} = 4$	100	87.50	95.83	100	79.17	91.67	83.33	91.07
$\mathcal{M} = 5$	100	91.67	95.83	100	62.50	95.83	91.67	91.07
$\mathcal{M} = 6$	100	100	79.17	100	87.50	87.50	54.17	86.90
$\mathcal{M} = 7$	100	100	54.17	95.83	79.17	95.83	75.00	85.71
$\mathcal{M} = 8$	100	100	95.83	100	70.83	66.67	79.17	87.50
$\mathcal{M} = 9$	100	100	95.83	95.83	83.33	100	54.17	89.88
$\mathcal{M} = 10$	95.83	83.33	91.67	91.67	83.33	91.67	87.50	89.29

Table 6.7: Segmentation Accuracy Results obtained from the HHMM and the CxSHSMM ($\mathcal{M} = 2,\ 3,\ \ldots,\ 10$) trained with 1% and 4% labeled data when tested on unseen data with missing observations.

8% **labeled data used in training**								
Models	Segmentation accuracy (%) of activities (a.1) → (a.7)							
	(a.1)	(a.2)	(a.3)	(a.4)	(a.5)	(a.6)	(a.7)	*Avg.*
HHMM	95.83	100	58.33	95.83	4.17	20.83	58.33	61.90
$\mathcal{M} = 2$	100	87.50	91.67	91.67	4.17	75.00	66.67	73.81
$\mathcal{M} = 3$	95.83	100	70.83	100	62.50	91.67	91.67	87.50
$\mathcal{M} = 4$	100	95.83	91.67	91.67	75.00	95.83	95.83	92.26
$\mathcal{M} = 5$	100	91.67	91.67	91.67	83.33	91.67	87.50	91.07
$\mathcal{M} = 6$	100	100	91.67	95.83	91.67	87.50	91.67	94.05
$\mathcal{M} = 7$	95.83	95.83	95.83	100	70.83	91.67	91.67	91.67
$\mathcal{M} = 8$	100	100	87.50	100	41.67	91.67	95.83	88.10
$\mathcal{M} = 9$	100	100	95.83	91.67	79.17	100	75.00	91.67
$\mathcal{M} = 10$	100	100	100	100	66.67	95.83	83.33	92.26
16% **labeled data used in training**								
HHMM	95.83	100	66.67	95.83	0	20.83	50.00	61.31
$\mathcal{M} = 2$	100	83.33	91.67	91.67	4.17	75.00	66.67	73.21
$\mathcal{M} = 3$	100	83.33	79.17	91.67	45.83	83.33	79.17	80.36
$\mathcal{M} = 4$	100	83.33	91.67	91.67	66.67	95.83	95.83	89.30
$\mathcal{M} = 5$	100	87.50	91.67	91.67	83.33	91.67	87.50	90.48
$\mathcal{M} = 6$	100	100	95.83	83.33	79.17	87.50	87.50	90.48
$\mathcal{M} = 7$	100	91.67	95.83	91.67	75.00	95.83	87.50	91.07
$\mathcal{M} = 8$	100	100	100	100	70.83	83.33	79.17	90.48
$\mathcal{M} = 9$	100	91.67	100	100	70.83	91.67	91.67	92.26
$\mathcal{M} = 10$	100	100	79.17	95.83	66.67	91.67	75.00	86.90

Table 6.8: Segmentation Accuracy Results obtained from the HHMM and the CxSHSMM ($\mathcal{M} = 2, 3, \ldots, 10$) trained with 8% and 16% labeled data when tested on unseen data with missing observations.

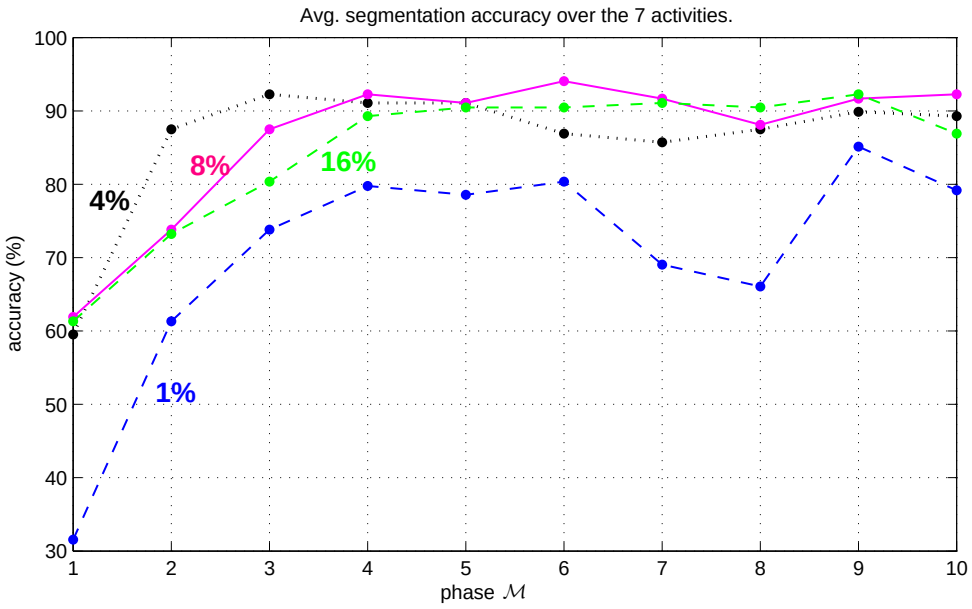


Figure 6.12: Average segmentation accuracy obtained from the HHMM ($\mathcal{M} = 1$) and the CxSHSMMs ($\mathcal{M} = 2,\ 3,\ \ldots,\ 10$) trained with 1%, 4%, 8% and 16% labeled data.

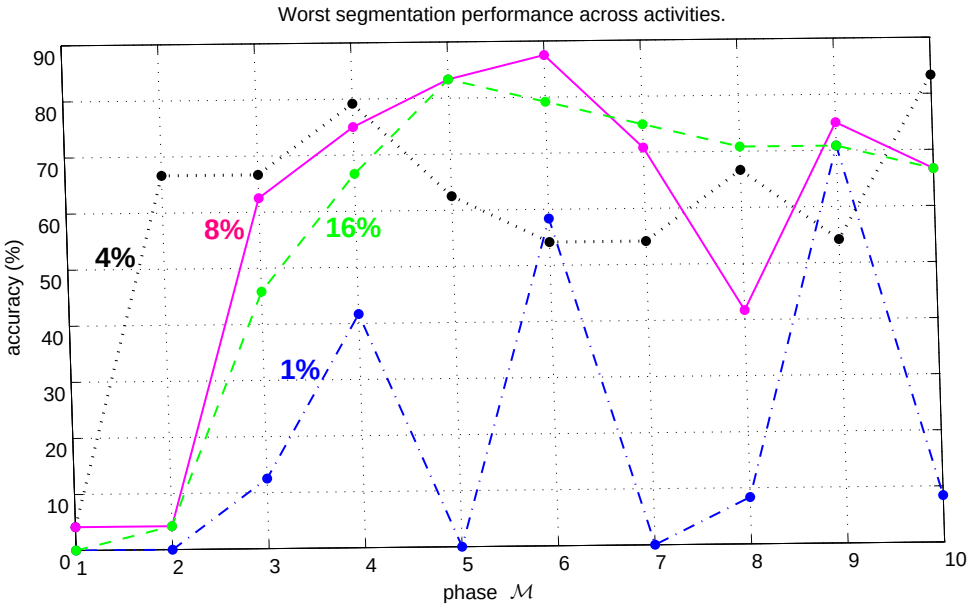


Figure 6.13: The lowest segmentation accuracy among (**a.1**) to (**a.6**) (Tabs. (6.7) & (6.8)) obtained from the HHMM ($\mathcal{M} = 1$) and the CxSHSMMs ($\mathcal{M} = 2,\ 3,\ \ldots,\ 10$) trained with 1%, 4%, 8% and 16% labeled data.

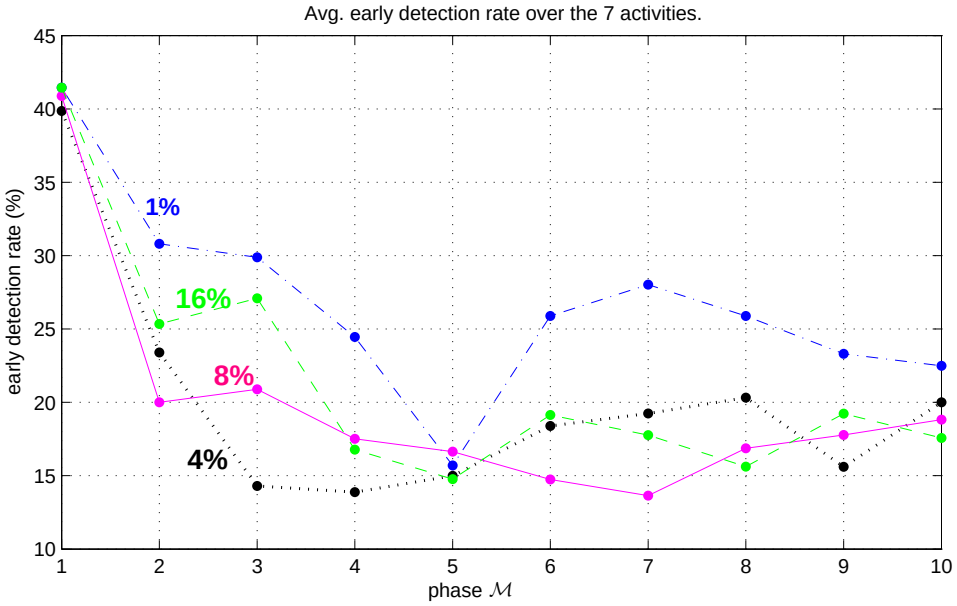


Figure 6.14: Average early detection rates obtained from the HHMM ($\mathcal{M} = 1$) and the CxSHSMMs ($\mathcal{M} = 2,\ 3,\ \ldots,\ 10$) trained with 1%, 4%, 8% and 16% labeled data.

accuracy) and (**a.5**) (41.67% accuracy). Another example is the 5-ph.CxSHSMM (Tab. (6.7)), which has $\sim$ 79% average accuracy but totally misses activity (**a.5**). As the activities mostly overlap, some activities are easy to be mislabeled, thus an overall satisfactory performance does not guarantee individual success.

We also observe from Fig. (6.12) that on average there is no noticeable difference among 4%, 8% or 16% labeled data when the models used are $\mathcal{M} \geq$ 4-ph.CxSHSMMs, even though the segmentation accuracy is more stable across all activities (Fig. (6.13)) when trained with 16% labeled data. This is similar for early detection rates, as shown in Fig. (6.14). On average, all $\mathcal{M} \geq$ 4-ph.CxSHSMMs can correctly identify activities around 15% to 20% of their executable time, which is reasonable in terms of applicability.

Finally, the experiment again confirms one of the advantages of the Coxian duration model is that it can work well with a small number of phases ($\mathcal{M}$ is as small as 4), thus requiring minimal increase in computation cost as compared with the two-layer HHMM with dramatic increase in the model performance. Also, the incorporation of both duration and hierarchical properties in our CxSHSMM leads to good results

even on complicated and overlapping ADLs. Lastly, our CxSHSMM models can directly handle noisy data returned from tracking systems with the help of a small amount of labels (as little as 4%) being supplied in training.

6.2 Topic Transition Detection in Educational Videos with the SHSMM

In this section we present an application for the CxSHSMM in a completely different area, that is, to detect topic transitions in educational videos. Our topic detection framework consists of two phases. The first phase performs shot detection and low level feature extraction and then classifies a shot in a meaningful label set Σ. This phase is described in section 6.2.1. In the next phase we train a HHMM and a CxSHSMM over the alphabet space Σ from the training data and use them in conjunction with the Viterbi to perform segmentation and annotation. The architecture of the framework is depicted in Fig. (6.15).

6.2.1 Short-based semantic classification

In this section we detail the first phase of the detection framework. This includes the formulation of an alphabet set Σ for shot labeling, low-level feature extraction and shot classification.

6.2.1.1 Short labels set: Σ

Existing work on the educational videos analysis (e.g., [Phung and Venkatesh, 2005, Phung, 2005a]) has studied the nature of this genre carefully. As noted in [Phung and Venkatesh, 2005], the axiomatic distinction of the educational genre is in its purpose: *teaching* and *training*. Further, a well-crafted segment that moves viewers to actions, or retains a long-lasting message, requires elaborate directing skills [3]. Based on a narrative analysis used in the educational domain and observed rules and conventions in the production of this media, the authors in [Phung and Venkatesh, 2005] propose a hierarchy of narrative structures at the shot level as shown in Fig.

[3] We note that the two closest video genre to educational videos is news and documentaries. In the description of what follows on educational genre, we can spot several similarities across these genres.

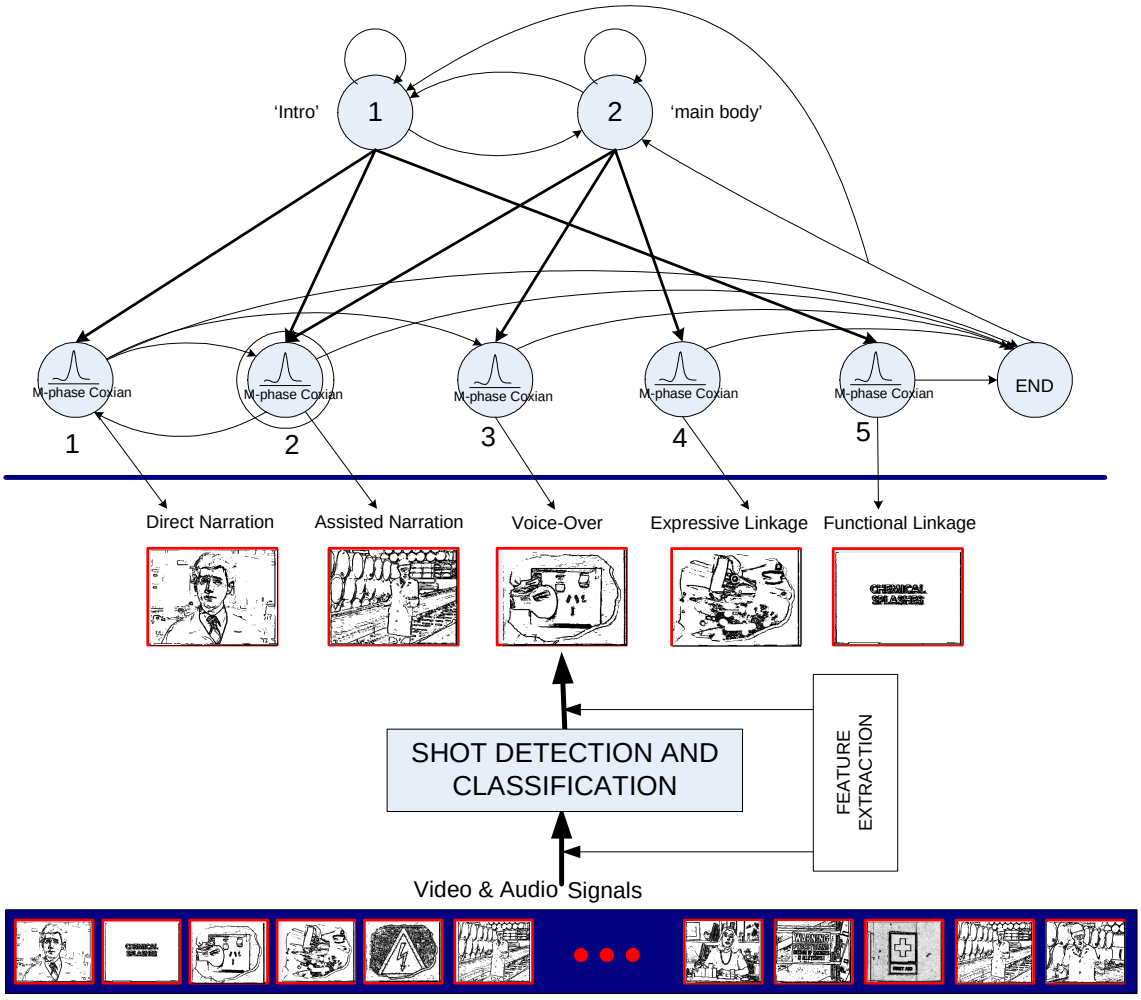


Figure 6.15: The architecture for topic detection framework.

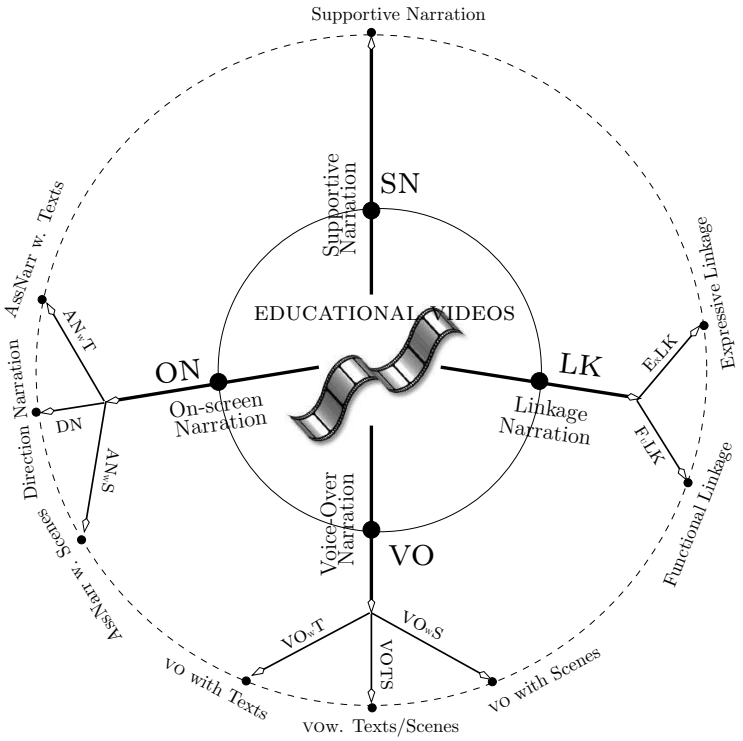


Figure 6.16: The hierarchy of narrative structures in educational videos proposed in [Phung and Venkatesh, 2005].

(6.16).

In this work we select the five most meaningful structures from this hierarchy for experimentation. This set Σ includes: *direct-narration* (DN), *assisted-narration* (AN), *voice-over* (VO), *expressive-linkage* (EL), and *functional-linkage* (FL). Direct-narration (DN) and assisted-narration (AN), referred to jointly as on-screen narration, refer to segments with the appearance of the narrator. The purpose of these sections is to speak to viewers with the "voice of authority", and is commonly used to demarcate a new topic or subtopic, to clarify a concept or to lead viewers through a procedure with examples. DN is a more strict form of on-screen narration. It involves eye-to-eye contact where the narrator speaks to the viewers directly. An analogy from news video is the anchor-shot. AN refers to parts of the video when a narrator appears in a more diverse style, and the attention of the viewers is not necessarily focused on him or her. Here, the purpose is not only to talk to the viewers, but also to emphasize a message by means of text captions and/or to convey an experience via background scenes. A similar structure from news for AN is

the reporting shot. Assisted narration can be used both in the introduction of a topic or in the main body, and thus this structure should be shared[4] by both higher semantics 'introduction' and 'main body'. As we see later, this knowledge is explicitly modeled and incorporated in the design of the topology for the SHSMM. An important feature is that although the semantics of AN is shared, typical durations are different when it is used in the introduction or the main body respectively. An AN section used to demarcate a new topic usually contains only one, and sometimes two shots, while an AN section used in the main body is typically long, spanning a number of shots. Conditioning on the parent (i.e., introduction or main body), the typical duration distribution of the AN section is learned automatically for each case by our model.

The voice-over (VO) structure is identified as sections where the audiotrack is dominated by the voice of the narrator, *but without* his or her appearance. The purpose of these segments is to communicate with the viewers via the narrator's voice. Additional pictorial illustration is usually further shown in the visual channel.

Expressive linkage (EL) and Functional linkage (FL) belong to the same broader linkage group in the hierarchy in Fig. (6.16). The purpose of the linkage structure is to maintain the continuity of a story line but there is *neither* on-screen *nor* voice-over narration involved. Functional linkage contains transition shots encountered in switching from one subject to the next. Usually, large superimposed text captions are used and the voice narration is completely stopped, with possibly music playing in the background. Expressive linkage, on the other hand, is used to create 'mood' for the subject being presented. For example, in the video presenting the fire safety topic, there is a segment in which the narration is completely stopped and then a sequence of pictures of the house on fire is shown. These scenes obviously do not give any direct instruction, rather they create a sense of 'mood' that helps the video to be more appealing and interesting.

6.2.1.2 Feature extraction and shot classification

The feature set and method for shot classification described in [Phung and Venkatesh, 2005] is employed in this work. The feature set is extracted from both visual and audio streams at the shot-based level. From the image sequence, we choose to de-

[4] In terms of parameterization, it is a form of *parameter tying*.

tect the frontal faces to reflect the appearance of the narrator using the CMU face detection algorithm [Rowley et al., 1998]; and captioned texts as one of the common means of conveying information in educational videos using the algorithm described in [Shim et al., 1998]. In order to classify a shot into direct-narration, voice-over, linkage, etc., further information is sought from the audio stream. Audio features are computed as the percentage of the following audio classes within a shot: vocal speech, music, silence, and non-literal sound. A shot is then classified into one of the elements of $\Sigma = \{DN, AN, VO, EL, FL\}$ using the classification framework reported in [Phung and Venkatesh, 2005]. Since we claim no contribution at this stage we shall refer readers to [Phung and Venkatesh, 2005] for full details on this classification scheme.

6.2.2 Experimental Results

6.2.2.1 Data and Shot-based classification

Our dataset D consists of 12 educational and training videos containing different types of subjects and presentational styles, and thus this constitutes a relatively noisy set of data. We manually provide groundtruth for these videos with topic transitions. In some cases, the groundtruth for topic transitions comes directly from the hardcopy guidelines supplied by the producer.

At the pre-processing stage, Webflix [Mediaware-Company, 1999] is used to perform shot transition detection and all detection errors are corrected manually. Since our contribution from this work is at the semantic level, the latter step is to ensure an error at the shot detection does not influence the performance of the system at higher levels. Since educational videos mainly contain cut and dissolve transitions, the shot detection accuracy is found to be very high with rare cases being erroneous. Given shot indices, each video is processed as described in section 6.2.1, and then each shot S is labeled as one of the elements of $\Sigma = \{DN, AN, VO, EL, FL\}$.

6.2.2.2 Model topology and parameterization

We will use four models in this experiment: the flat HMM and CxHSMM (as the baseline cases), the HHMM and the CxSHSMM. The observation space is set to Σ for every model. We train the flat HMM and CxHSMM with different numbers of states ranging from 2 to 5, where 2 is intended to be the minimum number of

states required (like 'intro' and 'main body') and 5 is the number of alphabets (i.e., in the relaxed way that the number of states equates to the number of alphabets). The semi-Markov version CxHSMM is further parameterized by a 3-phase Coxian distribution as the duration distribution of the states.

The topology shown in the top of Fig. (6.15) is used to construct the HHMM and the CxHSMM in this experiment. This topology specifies $Q^* = 2$ states at the top level where states 1 and 2 correspond to the introduction and the main body of the topic, respectively. The Markov chain at this level is similar to the flat HMM used in [Chaisorn et al., 2004] for news story segmentation[5]. We incorporate the assumed prior knowledge that a topic usually starts with either direct-narration, assisted-narration or functional linkage, thus state 1 has $\{1, 2, 5\}$ as its child set. Similarly, the main body can contain assisted-narration, voice-over or expressive linkage, hence its child set is $\{2, 3, 4\}$. Here state 2 (assisted narration) is shared by both parent state 1 ('intro') and 2 ('main body'). The bottom level has 5 states corresponding to 5 shot labels. To map the labels to the bottom states, we construct a diagonal-like B observation matrix and fix it, i.e., we do not learn B. The diagonal entries of B are set to 0.99 to relax the uncertainty during the classification stage. The duration models in the CxSHSMM are used with $\mathcal{M} = 3$ (3-ph.CxSHSMM) and $\mathcal{M} = 5$ phases Coxian (5-ph.CxSHSMM).

6.2.2.3 Detection Results

Given the dataset D, our evaluation employs a *leave-one-out strategy* to ensure an objective cross-validation. We sequentially pick out a video V and use the remainder set $\{D \setminus V\}$ to train the model, and then use V for testing. In the results that follow, this method is used for all cases including the flat HMM, the flat CxHSMM, hierarchical HMM, and the CxSHSMMs. A topic transition is detected when the introduction state at the top level is reached during the Viterbi decoding. Examples of Viterbi decoding with the 3-ph.CxSHSMM and HHMM are shown in Fig. (6.17).

To measure the performance, in addition to recall (RECALL) and precision (PREC) metrics, we include the F-score ($\mathrm{F-SCORE}$) metric defined as:

[5]They called 'transition' and 'internal' states instead of 'introduction' and 'main body'.

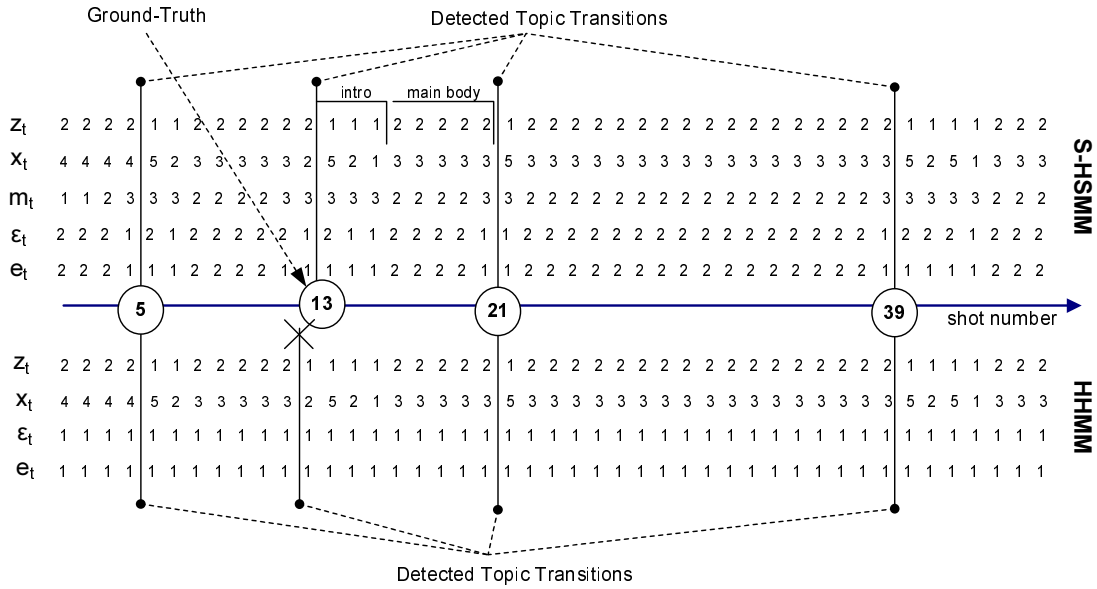


Figure 6.17: Example of Viterbi decoding for the 3-ph.CxSHSMM and the HHMM for the first 45 shots of video 'EESafety'. Readers may view these results in accordance with Fig. (5.3) for a clearer picture of the semantics of the DBN structure.

$$\mathrm{F-SCORE} = 2 \times \frac{\mathrm{RECALL} \times \mathrm{PREC}}{\mathrm{RECALL} + \mathrm{PREC}} = 2 \times \left(\frac{1}{\mathrm{RECALL}} + \frac{1}{\mathrm{PREC}} \right)^{-1}$$

While the recall rate measures how well the system can recover the true topic transitions, and a high precision ensures that it does not over-segment the video, the F-score shows the overall performance of the system. The ideal case is when $\mathrm{RECALL} = \mathrm{PREC} = 100\%$, $\mathrm{F-SCORE} = 1$, i.e., the highest performance the system can achieve.

The baseline cases: flat HMM and CxHSMM

Since initialization is crucial during EM learning, we apply multiple random restart points when conducting the experiments, including a uniform initialization. Although several restarts were used, the flat HMM is found to yield extremely poor results in all cases. Even when we train and test on the same dataset the flat HMM still produces poor detection results, proving unsuitable for our topical transition detection settings.

The flat CxHSMM gives slightly better results than the flat HMM, but still in all

		RECALL (%)	PREC (%)	F − SCORE (%)
HHMM	Uniform	42.58	81.48	0.559
	Random 1	83.23	84.47	0.840
	Random 2	83.23	84.87	0.840
	Random 3	83.23	84.87	0.840
	Random 4	41.29	80.00	0.545
	Random 5	83.87	83.87	0.839
3-ph SHSMM	Uniform	84.52	87.92	0.862
	Random 1	84.52	88.51	0.865
	Random 2	83.87	87.25	0.855
	Random 3	84.52	88.51	0.865
	Random 4	83.87	87.25	0.855
	Random 5	84.52	88.51	0.865
5-ph SHSMM	Uniform	84.52	87.92	0.862
	Random 1	83.87	88.44	0.861
	Random 2	83.87	88.44	0.861
	Random 3	83.87	88.44	0.861
	Random 4	83.87	88.44	0.861
	Random 5	83.87	88.44	0.861

Table 6.9: Detection Performances for the SHSMMs and the HHMM. Best performance for each case is highlighted in red (we note that best performances are attained in multiple cases and we select one of them to highlight).

ten runs the performance is still very low (RECALL = 7.74% and PREC = 48% in the best case). The poor performance of the HMM and CxHSMM is not surprising, since their forms are too strict to model a rather high level concept - the 'topic'. Furthermore, with the flat structures, they offer no mechanism to incorporate prior domain knowledge such as those that we use in the topology of the CxSHSMM and HHMM. This clearly shows that hierarchical models are much more suitable for video analysis than the flat ones. Given the poor results in the flat structure cases, we will omit the HMM and CxHSMM in the discussion that follows.

	RECALL (%)	PREC (%)	F − SCORE (%)
HHMM	83.23	84.87	0.840
3-ph SHSMM	84.52	87.92	0.862
5-ph SHSMM	83.87	88.44	0.861

Table 6.10: Best model selection at detection performances for the SHSMMs and the HHMM.

Video	TP			FP			Miss			GT
	CxSHSMM		HHMM	CxSHSMM		HHMM	CxSHSMM		HHMM	
	3-ph.	5-ph.		3-ph.	5-ph.		3-ph.	5-ph.		
1 - EESafety	10	10	8	1	1	3	3	3	5	13
2 - SSFall	4	3	4	1	1	1	2	3	2	6
3 - ElectS	6	6	6	2	2	1	2	2	2	8
4 - TrainHaz	18	18	20	2	2	2	3	3	1	21
5 - EyeS	10	10	10	0	0	1	0	0	0	10
6 - FootS	10	10	10	1	1	1	1	1	1	11
7 - HKeeping	11	11	11	3	3	3	1	1	1	12
8 - Maintn	9	9	8	1	1	3	4	4	5	13
9 - HandS	9	9	9	1	1	1	1	1	1	10
10 - SBurning	19	19	19	1	1	1	2	2	2	21
11 - HeadProt	6	6	5	1	1	3	1	1	2	7
12 - WeldingS	19	19	19	3	3	3	4	4	4	23
Sum	**131**	**130**	**129**	**17**	**17**	**23**	**24**	**25**	**26**	**155**

Table 6.11: Detection results for each video in the best performance cases of the CxSHSMMs and the HHMM (TP: True Positive, FP: False Positive, GT: Ground Truth).

Detection with the CxSHSMM and HHMM

The recall rate, precision and F-score for representative runs are reported in Tab. (6.9), in which the best performances are highlighted in red. The detection results for each individual video for the best case are shown in Tab. (6.10). With different random restarting points, including the uniform initialization, the performance of the HHMM ranges from poor to very good: 41.29% $\rightarrow$ 83.23% for recall and 80.00% $\rightarrow$ 84.47% for precision. On the contrary, both the CxSHSMMs consistently yield good results: the 3-ph.CxSHSMM has a recall rate ranging from 83.37% $\rightarrow$ 84.52% and precision from 87.92% $\rightarrow$ 88.51%, especially the 5-ph.CxSHSMM returning identical results across all the 5 different random initializations.

Since testing examples are not used during training, we also report in Tab. (6.10) the performances of the HHMM and CxSHSMMs in a likelihood-based 'best model selection' scheme. This scheme works as follows. As in the leave-one-out strategy, let V be a video selected from D, and N is the number of times we train the model using the dataset $\{D \setminus V\}$ (i.e., without V). Let $\theta_i(V)$ and $\mathcal{L}_i(V)$ $(i = 1 \dots N)$ respectively be the learned model and the likelihood (at convergence) obtained for i-th run. We then use the model θ_{i^*} to test on the unseen video V where $i^* = \underset{i=1\dots N}{\text{argmax}} \mathcal{L}_i(V)$. Simply speaking, we sequentially 'throw away' a video V, then select the best model (i.e., highest likelihood) among all runs to test on V. For the HHMM, the result stays the same as when we choose the best performance based on the F-score. For the 3-ph.CxSHSMM, the recall stays the same, while the precision slightly decreases from 88.51% to 87.92%. On the other hand, the 5-ph.CxSHSMM results are the same as best results in Tab. (6.9) as its performance is unaffected by random initialization. Both the CxSHSMMs are superior to the HHMM.

Tabs. (6.9) - (6.11) show that the 3-ph.CxSHSMM and 5-ph.CxSHSMM perform comparably and both are better than the HHMM in both recall and precision rates. As a result, the F-score improves from 0.840 to 0.861 (3-ph.CxSHSMM) and 0.865 (5-ph.CxSHSMM). While the recall rate improves only slightly, the $\sim 4\%$ improvement in the precision indicates that the HHMM tends to over-segment the video more frequently than the CxSHSMM. Also, the CxSHSMM is highly stable over random initialization while the HHMM's F-score fluctuates from 0.545 to 0.840. This confirms our belief that duration information is an important factor in our topic transition detection settings. The semi-Markov modeling has effectively overcome

the limitation of the strict Markov assumption of {future $\perp$ past | present}[6]} in the flat HMM, allowing longer temporal dependencies to be captured via the duration of the state. In addition, this experiment confirms the effectiveness of the Coxian duration model at exploiting the temporary dependency in the data, and furthermore shows its robustness to initialization.

Nevertheless, given a somewhat more contained set of data used in this experiment, the results from *both* the CxSHSMM and HHMM are better than the previous detection results of news story reported in [Chaisorn et al., 2004] (which came first in TRECVIC2003 testbed) and the heuristics and Bayesian approaches on topic detection in [Phung et al., 2002, Phung and Venkatesh, 2005]. Thus, it is also worthy to noting that our experiments not only imply the advantages of the CxSHSMM over the HHMM, but also show the contribution of hierarchical modeling alone.

6.3 Closing Remarks

In this chapter, we have presented two applications of the SHSMM: the first is for the problem of automatic learning and recognizing human daily activities and detecting anomalies in activity durations, and the second for detecting topic transitions in educational videos. Our contributions first include the construction of a robust and computationally efficient probabilistic framework for ADLs recognition and segmentation. We also build a novel detection scheme that is capable of detecting deviations in durations of activities - a very subtle form of activity anomaly. In addition, our novel topic transition detection framework proves to be efficient in working with educational videos and is potentially useful for other video types, such as news and documentaries.

[6] i.e., the future is conditionally independent of the past given the present

Chapter 7

Conclusion

7.1 Summary

This thesis has presented an investigation into modeling complex temporal data. It tackles two main inherent properties, namely *duration* and *hierarchy,* in a coherent and unified probabilistic graphical model. These properties arise naturally in many applications, and our study has focused on two important realms: recognition of activities of daily living in surveillance videos and high-level segmentation of professionally made educational videos. The common theme running through this thesis is the achievement of the discrete Coxian distribution and the integration of temporal and hierarchical inforamtion in our modeling framework and their successful applications. We start with a theoretical analysis in chapter 3 to address duration modeling problems in Markov models, while the next two chapters 4 and 5 propose solutions to this problem and bring together the duration and hierarchical modeling tasks under one framework, respectively. The applications of these models are presented in chapters 4 and 6.

Recognizing that critical to semi-Markov modeling is the choice of distributions to model state durations, our first contribution begins in chapter 3. It revises and re-represents existing modeling choices, being either discrete or continuous, in the HSMM under the same DBN representation. This study helps to shed light on the computation involved and shows that computational cost scales in order of the maximum possible duration and could grow unmanageable in many cases. It also brings to attention the problem of numerical instability for parameter estimation if continuous distribution is used.

This motivates us to the novel use of Coxian distribution, a mixture of geometric distributions ordered in phases, presented in chapter 4. The choice of Coxian has proven to be a good answer in many cases: it is dense in the field of non-negative distributions, and thus very flexible; it has analytical solutions for learning; it possesses a low number of parameters; and most importantly inference complexity does not depend on duration lengths but instead on the number of phases, which is typically much smaller.

Chapter 4 constitutes the first key theoretical contribution in the thesis. It introduces the Coxian duration Hidden Semi-Markov Model (CxHSMM) together with a full analysis, including its dynamic Bayesian network representation, inference and maximum likelihood estimation using EM. To make the CxHSMM more applicable to address real-world problems, we have also addressed how inference and learning can be achieved when there are missing observations (i.e. intermittent losses in tracking data) and when some part of the state sequence can be observed, for example, via the availability of other sensory data. We then presented an application of the CxHSMM to learn and recognize a set of behaviors in a smart home. We evaluated it against existing models (Geometric, Multinomial, Poisson and Inverse Gaussian). The results indicate that performance is greatly improved from explicit modeling of a state lifetime. In addition, our Coxian modeling is consistently superior to the Poisson and Inverse Gaussian cases. It further achieves a comparable performance with the Multinomial, whilst gaining a substantial improvement in computation time, which is a constant proportional to the maximum observation length. Our experiment ran approximately 25 times faster with the CxHSMM, but this factor could be much more in larger scales.

In addition to being *sequential*, latent structures found in real data are often *recursive* and *hierarchical*. This observation motivates the next theoretical contribution presented in chapter 5. In this chapter we introduce a novel and tractable form of stochastic model, termed the Coxian Switching Hidden Semi-Markov Model (CxSHSMM), that incorporates both duration and hierarchical modeling. The CxSHSMM has a shallow structure: the bottom layer can be viewed as a concatenation of many CxHSMMs, each initiated by different top-layer states switching in a Markovian manner. An important feature is that each child CxHSMMs at the

lower can share multiple parental states at the top while its duration model can be parameterized conditionally on the parent. This results in parameter compactness, and more importantly, reduces training sample complexity greatly – a similar advantage achieved by structure sharing in the HHMM presented in [Bui et al., 2004]. When duration distributions other than the Coxian are used, this modelling process reduces to a normal Switching Hidden Semi-Markov Model (SHSMM). We formalize inference and parameter estimation for these models under different settings and again point out the advantage of the Coxian parameterization. Finally, notwithstanding shallow structures, we note that deep hierarchical models could potentially be useful in some other applications (e.g., game simulation and parsing in natural language processing). We, thus, show how our models can be extended to full hierarchical hidden semi-Markov models of arbitrary depth and discussed relevant inference methods, in particular, the use of the Asymmetric Inside/Outside algorithm when the depth is high.

The latter part of this thesis presents our application contributions, organized in chapter 6. The first part is a framework for learning, segmenting, recognizing and detecting anomalies in activities of daily livings (ADLs) in the smart home environment using the CxSHSMM. States at the bottom layer are used to represent atomic activities (such as cooking-at-the-stove) while the top level represents finer activities made of sub-activities (such as preparing-breakfast). Prior domain knowledge is further incorporated into the structural topology of the model. The first experiment is carried in a standard supervised setting: models for complex ADLs are first learned from unlabeled data and later used to segment and recognize activities online. Again, it is shown empirically that hierarchical modeling, but without duration, as in the standard HHMM performs much worse than the proposed CxSHSMM. In addition we observe that the CxSHSMM delivers a more robust performance across activities (e.g. as compared with the Multinomial duration SHSMM). This may be attributed to the fact that our modeling is more compact in the parameter and thus less sensitive to initialization and data variations.

Detection of abnormalities is important from a surveillance perspective and is addressed in our next experiment. In particular, we focus on the duration aspect of anomalies, which is more subtle and harder to detect, but important in the elderly-care domain. We believe that this kind of abnormality is yet addressed explicitly

in the literature. Our detection scheme is built on top of the previous 'normal' activity recognition framework where trained models on normal data are utilized and adapted for the tasks. Our framework offers the following advantages: the ability to automatically infer models for abnormal behaviours from the learned normal models, hence there is no need to train on abnormal data, and thus be able to deal with the scarcity of negative training data as abnormalities are rare and varied; a general abnormal model is generated for each activity class, thus preventing the network from continuing to grow by adding new abnormal models for any unseen data; and the ability to detect when the abnormal activities return to normality, therefore minimizing false alarms.

Our last experiment in the ADL domain is to tackle more challenging scenarios including lossy observations (occlusion, out of camera view, etc.) and overlaps in activities' trajectories[1], and we set out to automatically learn, segment and classify a richer set of activities in sequences. The experiment shows that the CxHSMM can robustly handle lossy data (35% of observations missing) and overlapping activities, delivering an average recognition accuracy of 91% with the help of a small amount of labels (as little as 4%) supplied in training.

Chapter 6 continues to explore the CxSHSMM in a different domain, but related modeling issues: segmentation of educational video into topical units. In this framework states at the bottom layer are mapped into shots while states at the top layer represent higher-order semantics such as introductions or main body sections of a video segment about a particular topic. We evaluate the CxSHSMM against different modeling issues including no hierarchy with duration (flat HSMM) or without duration (flat HMM) and hierarchy without duration (HHMM). As expected the flat structure has delivered relatively poor results. The CxSHSMM detects topic transitions most accurately (88.44% precision achieved with 5-ph.CxSHSMM), even though the HHMM also delivers a relatively good performance. This result demonstrates that the modeling of duration, together with hierarchy, is a powerful tool in the extraction of higher level semantics. In addition, although the experiments are carried out on educational and training film genres, the method can easily be applied to other genres. We believe that the promise of the approach lies in its unified probabilistic handling of temporal properties and shared hierarchical structures,

[1] Overlap is shown by the excessive share of common child states in the CxSHSMM topology.

allowing it to handle long video sequences with inherent variability and complicated semantics. Finally, our last important remark is that under different settings in all our experiments in both activity recognition and topic detection domains, the Coxian duration models deliver good performances whilst requiring a relatively small number of phases (e.g. 5 phases) and thus saving huge computational costs. Thus, Coxian modeling is a promising choice for real-life deployments in the concerned domains and potentially for many other applications.

7.2 Future work

Opportunities for further work lie in several directions. From a theoretical point of view, some extensions stand out. One is to address interleaved executions in activity modeling. It is common in daily life that we often pursue multiple plans concurrently. That is to suspend the current activity, start on a new one, then come back to the previous activity at a later point. For example, cooking can be interrupted by a phone call. This problem is known to be difficult and little work has been done. In [Avrahami-Zilberbrand et al., 2005], agents are allowed to have interleaved plans, but learning of either plan durations or interleaving is not supported. The work in [Marhasev et al., 2006] is somewhat related to this issue by allowing non-stationary transition modeling, but nonetheless still suffers from the huge number of parameters required. The biggest challenge in handling interleaving is perhaps the expense of modeling and computational cost because as soon as suspension happens, information for the entire current execution needs to be stored for later resumption. By having fewer parameters and computational efficiency, we speculate that the Coxian distribution can be a good candidate for this task. For example, instead of putting the entire state information of the interrupted activity on the stack for later resumption, we may only need to remember the 'phase' at which the activity is interrupted. That could be the current phase of Coxian distribution (as an indication of time spent) associated with the interleaving activity. When resuming, the activity only needs to return the next phase of its execution. In fact, it would be interesting to see the work of [Avrahami-Zilberbrand et al., 2005] and [Marhasev et al., 2006] to extend in this direction.

Shallow hierarchical modelling has proven to work well in this thesis for the applications of activity recognition and video segmentation. However, as mentioned

earlier, deep hierarchical probabilistic models can potentially be useful in some particular domains, in particular, in computer simulation, such as to simulate moving behaviours of an agent whose plan is decomposed in deep hierarchy, or the problem of parsing in natural language processing. Since the HHMM can be shown as a special kind of stochastic context free grammar (SCFG) with bounded stack [Phung, 2005b], our full hierarchical hidden semi-Markov models presented in chapter 5 can also be represented as a form of SCFG, but clearly richer than that of the HHMM. This raises a very interesting question since, to our knowledge, no work has addressed duration modeling with tractable computation and numerical stability. An exception is the recent work of the dynamic conditional random field (DCRF) for text parsing and chunking in [Sutton et al., 2007] which can potentially incorporate duration information. But the DCRF is parameterized as undirected graphical models and duration distribution is hard to learn, in addition, it lacks probabilistic interpretation. An application of our theoretical development in section 5.6 could potentially be attractive for the natural language processing community.

Our current setting in this thesis is essentially non-Bayesian. Even though we have shown empirically via cross-validation model selection that a small number of phases in the Coxian is often sufficient for our experiment, more robust schemes can be developed by further 'smoothing' the model by going Bayesian, very similar to Bayesian extension made to the standard HMM in [Beal, 2003]. Essentially, the parameters are further endowed with conjugate prior distributions and are integrated out during inference to achieve embedded smoothing behaviours, making it more robust against overfitting. However, this is done at the cost of losing tractability for exact inference. However, approximate inference such as Gibbs sampling or variational as presented in [Beal, 2003] can still be applied. Such Bayesian extension would expect to deliver a better version of the Bayesian HMM in [Beal, 2003].

Finally, we are particularly interested in seeing the direct applications of the CxHSMM and CxSHSMM in three domains: speech processing, computational biology and desktop activity recognitions. Speech recognition has been a holy-grail in research and it is well known that the state-of-the-art modeling choice is the HMM and its semi-Markov extension. It has been reported that semi-Markov extension delivers more robust solutions than the HMM in various speech recognition tasks. However, as we have mentioned repeatedly, computation hinders its applicability. Our

CxHSMM can be an excellent replacement for this task. Secondly, there is a recent surge in modeling biological data with the HMM [Durbin, 1998, Koski, 2001] and good results have been reported. But, again, these schemes have difficulty with capturing long-range correlations, which is often the case in biological data. It is natural to wonder how our modeling would help improving modelling accuracy in this domain. Lastly, activities on the desktop are closely aligned to the work done in the thesis. The recent CALO project[2] is an example of the need for recognizing and detecting 'outlier' activities performed on the desktop [Stumpf et al., 2005]. Our models are again particularly attractive for this domain since human behaviours on the desktop often have peculiar duration distribution, e.g., checking emails would be much shorter than composing a talk slide. Activities are also often interleaved; for example, we often pause to read an email or surf the web for information while programming.

[2]http://caloproject.sri.com/

Appendix A

Data Collection and Observation Model

Generally there are two steps in a complete activity recognition framework. The first step involves activity tracking and data conversion. That is to detect the occurrence of activities via the use of video cameras or various sensors and algorithms to convert the sensory data to appropriate features. The next step is activity analysis on the feature data and includes modeling, synthesis, prediction, recognition and anomaly detection. Since the focus of this thesis lies in the second step (i.e. modeling, recognition and anomaly detection), we briefly describe the distributed surveillance system used for data collection in our work as well as the interface between this low level tracking module and our high-level activity recognition scheme.

The distributed tracking system, developed by [Nguyen, 2004b], is chosen for data collection because of its reliability, low cost and its availability. The tracking module consists of cheap static cameras positioned at ceiling corners of the kitchen environment (dubbed the vision laboratory in [Nguyen, 2004b]). Each camera has a Camera Processing Module (CPM) connected to a Central Module (CM). The CM maintains an in-the-scene object database and coordinates different CPMs. The CPM first performs a blob segmentation by background subtraction and then tracking. At each time slice the CM receives the matched blobs and tracked states output from the CPM to form trajectories of tracked objects. When an object is in the overlapping of different cameras' field of views (FOVs), the CM correlates the tracking between CMPs by choosing the most suitable camera to track. That is the nearest camera to the object that provides a non-occluded view. When an object is lost (whether

due to occlusion or out of cameras' FOVs), a matching procedure is performed at the CM to recover the track.

Nguyen [Nguyen, 2004b] also specifies a compressed observation model for each camera from manually labeled ground truth, which is to serve as an interface between the tracking module and the high-level activity recognition module. Assuming the statistics are spatially invariant, a compressed observation model for camera C is a 3×3 matrix $B_{o \mid s} = \Pr(o \mid s,\ C)$, in which o is the tracked position returned by the tracking system and s is the true position.

$$B_{o|s} = \begin{bmatrix} \Pr(o_{\text{northwest}} \mid C) & \Pr(o_{\text{north}} \mid C) & \Pr(o_{\text{northeast}} \mid C) \\ \Pr(o_{\text{west}} \mid C) & \Pr(o_{\text{center}} \mid C) & \Pr(o_{\text{east}} \mid C) \\ \Pr(o_{\text{southwest}} \mid C) & \Pr(o_{\text{south}} \mid C) & \Pr(o_{\text{southeast}} \mid C) \end{bmatrix}$$

where $\Pr(o_{\text{northwest}} \mid C)$ is the probability that the tracked location falling into the north-west region of the true location and so on. The environment floor is divided into grids of 1m^2 squared cells, and the observation model is obtained by comparing 100 coordinate samples returned from the tracking system with the manually acquired true positions. We observe that the differences between observation models of different cameras are not very noticeable, hence, the observation model employed in this thesis is simply the average of those associated with the used cameras. It provides emission probabilities of the production layer (the lowest semi-Markovian layer) in our high-level activity recognition module, and is not updated during learning.

Apart from using cameras (e.g. [Grimson et al., 1998, Zhong et al., 2004, Du et al., 2006, Cheng et al., 2006]) like us, current work on tracking employs various kinds of sensors ranging from GPS [Liao et al., 2004] to simple and ubiquitous sensors including pressure mats, contact switches, heat sensors, current sensors, and break-beam sensors [Wilson, 2005]; or wearable sensors like accelerometers [Huynh et al., 2007] and RFID [Wyatt et al., 2005]; or sensors for detecting user's ambient conditions such as humidity and temperature[Wang et al., 2007]. Our high-level activity recognitions scheme is able to incorporate new knowledge from various sensory information if available by extending the observation model. In addition, readings from cameras and other sensors are commonly subjected to intermittent lost signals, however, as shown in experiments our models are also capable of effectively handling data with missing observations.

Bibliography

Brett Adams, Chitra Dorai, and Svetha Venkatesh. Automated film rhythm extraction for scene analysis. In *IEEE International Conference on Multimedia and Expo*, pages 1056–1059, Tokyo, Japan, August 2001.

Philippe Aigrain, Philippe Jolly, and V. Longueville. Medium knowledge-based macro-segmentation of video into sequences. In M. Maybury, editor, *Intelligent Multimedia Information Retrieval*, pages 159–174. AAAI Press/MIT Press, 1998.

E. L. Andrade, S. J. Blunsden, and R. B. Fisher. Performance analysis of event detection models in crowded scenes. In *ICINCO 2005, Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics*, Barcelona, Spain, 2005.

G. Arfken. *Mathematical Methods for Physicists.* Academic Press, Orlando, 3rd edition, 1985.

D. Avrahami-Zilberbrand, G. A. Kaminka, and H. Zarosim. Fast and complete symbolic plan recognition: Allowing for duration, interleaved execution, and lossy observations. In *In Proceedings of the IJCAI Workshop on Modeling Others from Observations (MOO-05)*, 2005.

Katrina Ball. Australia's ageing population and its implication for our future, 2003. TAFE Director's Australia 2003 Annual Conference, Adelaide.

L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Proceedings of PERVASIVE*, page 1âĂŞ17, 2004.

Richard E. Barlow and Frank Proschan. *Statistical theory of reliability and life testing : probability models.* To Begin With, Silver Spring, Md., 1981. Richard E. Barlow, Frank Proschan. ill. ; 24 cm. Reprint, with corrections, of title published in 1975 by Holt, Reinhart and Winston. Includes index.

F. I. Bashir, A. A. Khokhar, and D. Schonfeld. Object trajectory-based activity classification and recognition using hidden markov models. *Image Processing, IEEE Transactions on*, 16(7):1912–1919, 2007.

M.J Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, The Gatsby Computational Neuroscience Unit, University College London, 2003.

J. Ben-Arie, Z. Wang, P. Pandit, and S. Rajaram. Human activity recognition using multidimensional indexing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(8):1091–1104, 2002.

A. Bobbio and K.S. Trivedi. Computation of the distribution of the completion time when the work requirement is a PH random. *Stochastic Models*, 6:133–149, 1990.

A. Bobbio, A. Horvath, M. Scarpa, and M. Telek. Acyclic discrete phase type distributions: properties and a parameter estimation algorithm. *Performance Evaluation*, 54(1):1–32, 2003.

M. Brand. Coupled hidden markov models for modeling interacting processes. *Neural Computation*, 1996.

M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. *Computer Vision and Pattern Recognition*, page 994âĂŞ999, 1997.

Matthew Brand and Vera Kettnaker. Discovery and segmentation of activities in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8): 844–851, August 2000.

C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proc. CVPR*, volume 97, pages 569–574, 1997.

H. H. Bui. A general model for online probabilistic plan recognition. *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

H.H Bui, S. Venkatesh, and G. West. On the recognition of abstract Markov policies. In *Proceedings of the National Conference on Artificial Intelligence*, pages 524–530, 2000.

Hung H. Bui, Svetha Venkatesh, and G West. Policy recognition in the abstract hidden markov model. *Journal of Articial Intelligence Research 17*, pages 451–499, 2002.

Hung H. Bui, Dinh Q. Phung, and Svetha Venkatesh. Hierarchical hidden markov models with general state hierarchy. In Deborah L. McGuinness and George Ferguson, editors, *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 324–329, San Jose, California, USA, 2004. AAAI Press / The MIT Press.

J. Callut and P. Dupont. Sequence discrimination using phase-type distributions. In *1th European Conference on Machine Learning (ECML)*, pages 78–89, Berlin, Germany, 2006.

O. Cappe. Ten years of hmms. *On-line bibliography available from http://tsi. enst. fr/cappe/docs/hmmbib. html*, 2001.

Lekha Chaisorn, Tat-Seng Chua, Chin-Hui Lee, and Qui Tian. A hierarchical approach to story segmentation of large broadcast news video corpus. In *IEEE International Conference on Multimedia and Expo*, Taipei, Taiwan, June 2004.

M. T. Chan, A. Hoogs, J. Schmiederer, and M. Petersen. Detecting rare events in video using semantic primitives with hmm. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, 2004.

R. Chellappa, N. Vaswani, and A. Roy Chowdhury. Activity modeling and recognition using shape theory. In *Behavior Representation in Modeling and Simulation*, 2003.

C. Chen, J. Liang, H. Hu, L. Jiao, and X. Yang. *Factorial Hidden Markov Models for Gait Recognition*, volume 4642 of *LECTURE NOTES IN COMPUTER SCIENCE*. 2007.

M. Cheng, B. Pham, and D. Tjondronegoro. Tracking and video surveillance activity analysis. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 367–373, 2006.

R. T. Collins, Gross R., and R. J. Shi. Silhouette-based human identification from body shape and gait. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 351–356, 2002.

David Cox. A use of complex probabilities in the theory of stochastic processes. *Cambridge Philosophical Society*, 51:313–319, 1955.

Geiger Dan. Graphical models and exponential families. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, pages 156–165, San Francisco, CA, 1998. Morgan Kaufmann Publishers.

Thomas Dean and Jeiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(3):142 – 150, 1989.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via EM algorithm (with discussion). *Journal of the Royal Statistical Society*, 39 (1):1–38, 1977.

Y. Du, F. Chen, W. Xu, and Y. Li. Recognizing interaction activities using dynamic bayesian network. In *18th International Conference on Pattern Recognition, 2006 (ICPR 2006)*, volume 1, pages 618 – 621, 2006.

Thi V. Duong, Hung H. Bui, Dinh Q. Phung, and Svetha Venkatesh. Activity recognition and abnormality detection with the Switching Hidden Semi-Markov Model. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 838–845, San Diego, 20-26 June 2005. IEEE Computer Society.

R. Durbin. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

M. J. Faddy and S. I. McClean. Analysing data on lengths of stay of hospital patients using phase-type distributions. *Applied Stochastic Models in Business and Industry*, 15(4):311–317, 1999.

M.J. Faddy. Penalised maximum likelihood estimation of the parameters in a coxian phasetype distribution. In G. Latouche and P. Taylor, editors, *Matrix-Analytic Methods Theory and Applications*, page 107âĂŞ114. World Scientific, Singapore, 2002.

M.J. Faddy. Examples of fitting structured phase-type distributions. *Applied stochastic models and data analysis*, 10(44):247–255, 1994.

M.J. Faddy. On inferring the number of phases in a coxian phase-type distribution. *Communications in statistics: Stochastic models*, 14(1):407–417, 1998.

M.J. Faddy and S.I. McClean. Analysing data on lengths of stay of hospital patients using phase-type distributions. *Applied Stochastic Models in Business and Industry*, 15(4):311 – 317, 2000.

Shai Fine, Yoran Singer, and Nftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.

A. Galata, N. Johnson, and D. Hogg. Learning variable-length markov models of behavior. *Computer Vision and Image Understanding*, 81(3):398–413, 2001.

M. J. F. Gales and S. J. Young. The theory of segmental hidden markov models. Technical Report CUED/F-INFENG/TR133, Cambridge University Engineering Department, June 1993.

N. Ghanem, D. DeMenthon, D. Doermann, and L. Davis. Representation and recognition of events in surveillance video using petri nets. In *Computer Vision and Pattern Recognition Workshop, 2004 Conference on*, pages 112–112, 2004.

S. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 742–749, 2003.

P. A. Greiner, D. A. Snowdon, and F. A. Schmitt. The loss of independence in activities of daily living: The role of low normal cognitive function in elderly nuns. *American journal of public health(1971)*, 86(1):62–66, 1996.

W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in asite. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 22–29, 1998.

O. Haggstrom, S. Asmussen, and O. Nerman. EMPHT: A program for fitting phase type distributions, 1992. Chalmers University of Technology, The University of Goteborg.

K. Z. Haigh and H. A. Yanco. Automation as caregiver: A survey of issues and technologies. In *AAAI-02 Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care*, pages 39–53, 2002.

R. Hamid, Y. Huang, and I. Essa. ArgmodeâĂŞactivity recognition using graphical models. 2003.

R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, and G. Coleman. Detection and explanation of anomalous activities: representing activities as bags of event n-grams. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, 2005.

A. Hanjalic, R. L. Lagendijk, and J. Biemond. Automated high-level movie segmentation for advanced video retrieval systems. *IEEE Transactions in Circuits and Systems for Video Technology*, 9(4):580–588, 1999.

Alan Hanjalic. Shot-boundary detection: Unraveled and resolved? *IEEE Transaction in Circuits and Systems for Video Technology*, 12(2):90–105, 2002.

K. Hara, T. Omori, and R. Ueno. Detection of unusual human behavior in intelligent house. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 697–706, 2002.

Jesse Hoey. Hierarchical unsupervised learning of facial expression categories. In *IEEE Workshop on Detection and Recognition of Events in Video*, pages 99–106, Vancouver, BC, Canada, 2001.

S. Hongeng and R. Nevatia. Large-scale event detection using semi-hidden markov models. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1455–1462, 2003.

S. Hongeng, F. Bremond, and R. Nevatia. Bayesian framework for video surveillance application. In *15th International Conference on Pattern Recognition (ICPR'00)*, volume 1, page 1164, 2000.

S. Hongeng, R. Nevatia, and F. Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96(2):129–162, 2004.

Andras Horvath and Miklos Telek. PhFit: A general phase-type fitting tool. In *Lecture Notes in Computer Science*, volume 2324/2002, pages 1–14. Springer Berlin / Heidelberg, 2002.

W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1450–1464, 2006.

T. Huynh, U. Blanke, and B. Schiele. Scalable recognition of daily activities with wearable sensors. In *LoCA*, Lecture Notes in Computer Science. Springer, 2007.

Ichiro Ide, Koji Yamamoto, and Hidehiko Tanaka. Automatic video indexing based on shot classification. In *First International Conference on Advanced Multimedia Content Processing*, pages 99–114, Osaka, Japan, November 1998.

Claudia Isensee and Graham Horton. Approximation of discrete phase-type distributions. In *Proceedings of the 38th Annual Symposium on Simulation*, pages 99–106, 2005.

Eva Ishay. *Fitting Phase-Type Distributions to Data from a Telephone Call Center*. PhD thesis, Israel Institute of Technolgoy, 2002. Master Thesis.

U. Iurgel, R. Meermeier, S. Eickeler, and G. Rigoll. New approaches to audio-visual segmentation of TV news for automatic topic retrieval. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 3, pages 1397–1400, Salt Lake City, Utah, 2001.

Finn V. Jensen. *An Introduction to Bayesian Networks*. Springer Verlag, 1996.

M. A. Johnson and M. R. Taaffe. Matching moments to phase distributions: Mixtures of erlang distributions of common order. *Communications in Statistics: Stochastic Models*, 5(4):711–743, 1989.

M.A. Johnson and M.R. Taaffe. The denseness of phase distributions. In *Purdue School of Industrial Eng. Research Memoranda*, volume 88-20, 1988.

Mary A. Johnson and Michael R. Taaffe. Matching moments to phase distributions: nonlinear programming approaches. *Stochastic Models*, 6(2):259–281, 1990.

Mary A. Johnson and Michael R. Taaffe. An investigation of phase-distribution moment-matching algorithms for use in queueing models. *Queueing Systems*, 8 (1):129–147, 1991.

Norman Lloyd Johnson and Samuel Kotz. *Distributions in statistics*. Houghton Mifflin, New York, 1969. Norman L. Johnson, Samuel Kotz. 24 cm. Vol. 4 published by John Wiley & Sons, New York. Contents: [v.1] Discrete distributions– [v.2] Continuous univariate distributions 1.– [v.3] Continuous univariate distributions 2.– [v.4] Continuous multivariate distributions.

Norman Lloyd Johnson, Samuel Kotz, Adrienne W. Kemp, and Norman Lloyd Johnson. *Distribution in Statistics: Univariate Discrete Distributions*. Wiley series in probability and mathematical statistics. Applied probability and statistics. Wiley, New York, 2nd edition, 1993.

Norman Lloyd Johnson, Samuel Kotz, and N. Balakrishnan. *Distribution in Stastistics: Continuous Univariate Distributions*. Wiley series in probability and mathematical statistics. Wiley, New York, 2nd edition, 1994.

Michael I. Jordan. *Introduction to Graphical Models (X)*. MIT Press, Cambridge, MA, 2004. Forthcoming.

S. Katz, A. B. Ford, R. W. Moskowitz, B. A. Jackson, and M. W. Jaffe. Studies of illness in the aged. the index of adl: A standardized measure of biological and psychosocial function. *JAMA*, 185:914–9, 1963.

Henry Kautz, Oren Etzioni, Dieter Fox, and Dan Weld. Foundations of assisted cognition systems. Technical report, University of Washington, CSE, March 2003.

G. I. Kempen, N. Steverink, J. Ormel, and D. J. Deeg. The assessment of adl among frail elderly in an interview survey: self-report versus performance-based tests and determinants of discrepancies. *Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, 51(5):254–260, 1996.

E. Kijak, L. Oisel, and P. Gros. Hierarchical structure analysis of sport videos using HMMs. In *Int. Conf. on Image Processing*, volume 2, pages II–1025–8 vol.3, 2003.

K. M. Kitani, Y. Sato, and A. Sugimoto. Deleted interpolation using a hierarchical bayesian grammar network for recognizing human activity. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 239–246, 2005.

T. Koski. *Hidden Markov Models for Bioinformatics*. Kluwer Academic Publishers, 2001.

Gaitanis Kosta, Correa Hernandez Pedro, and Macq Benoit. Human action recognition using silhouette based feature extraction and dynamic bayesian networks. In *7th International Workshop on Image Analysis for Multimedia Interactive Services*, Incheon Korea, 2006.

M. P. Lawton. Aging and performance of home tasks. *Human Factors*, 32(5):527–36, 1990.

H. K. Lee and J. H. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):961–973, 1999.

J. Lester, T. Choudhury, and G. Borriello. *A Practical Approach to Recognizing Physical Activities*, volume 3968/2006 of *Lecture Notes in Computer Science: Pervasive Computing*. Springer Berlin / Heidelberg, 2006.

Stephen E. Levinson. Continuously variable duration hidden markov models for automatic speech recognition. *Computer Speech and Language*, 1(1):2945, March 1986.

Xiaolin Li, M. Parizeau, and R. Plamondon. Training hidden markov models with multiple observations-a combinatorial method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(4):371 – 377, 2000.

L. Liao, D. Fox, and H. Kautz. Location-based activity recognition using relational markov networks. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

Lin Liao, Dieter Fox, and Henry Kautz. Learning and inferring transportation routines. *Proc. of the National Conference on Artificial Intelligence(AAAI-04)*, 2004.

T. Lin and H. J. Zhang. Automatic video scene extraction by shot grouping. *Pattern Recognition*, 4:39–42, 2000.

Zhu Liu and Qian Huang. Detecting news reporting using audio/visual information. In *International Conference on Image Processing*, pages 24–28, Kobe, Japan, October 1999.

S. Luhr, Hung H. Bui, Svetha Venkatesh, and Geoff West. Recognition of human activity through hierarchical stochastic learning. In *Int. Conf. on Pervasive Computing and Communication*, pages 416–422, 2003.

S. Luhr, S. Venkatesh, G. West, and H. H. Bui. Duration abnormality detection in sequences of human activity. Technical report, Department of Computing, Curtin University of Technology, May 2004.

E. Marhasev, M. Hadad, and G. A. Kaminka. Non-stationary hidden semi markov models in activity recognition. In *In Proceedings of the AAAI Workshop on Modeling Others from Observations*, 2006.

A.H. Marshall and S.I. McClean. Using coxian phase-type distributions to identify patient characteristics for duration of stay in hospital. *Health Care Management Science*, 7(4):285–289, 2004.

A.H. Marshall, B. Shaw, and S.I. McClean. Estimating the costs for a group of geriatric patients using the coxian phase-type distribution. *Statistics in Medicine*, 26(13):2716–2729, 2007.

J. Mathews and K. Fink. *Numerical Methods with MATLAB*. Prentice-Hall, Upper Saddle River, NJ, 1999.

Mediaware-Company. Mediaware solution webflix professional V1.5.3, 1999. http://www.mediaware.com.au/webflix.html.

G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(8):873–889, 2001.

D. Minnen, I. Essa, and T. Starner. Expectation grammars: leveraging high-level expectations for activity recognition. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, 2003.

C. D. Mitchell and L. H. Jamieson. Modeling duration in a hidden markov model with the exponential family. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pages II.331–II.334, Minneapolis, Minnesota, April 1993.

Carl Mitchell, Mary Harper, and Leah Jamieson. On the complexity of explicit duration HMMs. *IEEE Transactions on Speech and Audio Processing*, 3(3), May 1999.

T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.

J. Muncaster and Y. Ma. Activity recognition using dynamic bayesian networks with automatic state selection. In *Motion and Video Computing, 2007. WMVC'07. IEEE Workshop on*, pages 30–30, 2007.

K. Murphy and M. Paskin. Linear-time inference in hierarchical HMMs. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, Cambridge, MA, 2001. MIT Press.

K. Murphy and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. *Sequential Monte Carlo Methods in Practice*, page 499âĂŞ516, 2001.

Kelvin Murphy. Learning switching kalman filter models. Technical report, Campaq Cambridge Research Lab, 1998.

Kevin Patrick Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Bekerley, 2002.

E. D. Mynatt, I. Essa, and W. Rogers. Increasing the opportunities for aging in place. In *Proceedings on the 2000 conference on Universal Usability*, pages 65–71, 2000.

Milind Ramesh Naphade and Thomas S. Huang. Discovering recurrent events in video using unsupervised methods. In *Int. Conf. om Image Processing*, volume 2, pages 13–16, Rochester, NY, USA, 2002.

P. Natarajan and R. Nevatia. Coupled hidden semi markov models for activity recognition. In *Motion and Video Computing, 2007. WMVC'07. IEEE Workshop on*, pages 10–10, 2007a.

P. Natarajan and R. Nevatia. Hierarchical multi-channel hidden semi markov models. In *the International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, 2007b.

Marcel F. Neuts. *Structured stochastic matrices of M/G/1 type and their applications*. Probability, pure and applied ; 5. Marcel Dekker, New York, 1989. Marcel F. Neuts. 24 cm.

Marchel F. Neuts. *Matrix-Geometric Solutions in Stochastic Models*. The Johns Hopkins University Press, Baltimore and London, 1981.

N. Nguyen and S. Venkatesh. Discovery of activity structures using the hierarchical hidden markov model. In *16th British Machine Vision Conference (BMVC 2005)*, Oxford, UK, 2005.

N. Nguyen, H. Bui, and S. Venkatesh. Recognising behaviour of multpile people with hierarchical probabilistic and statistical data association. In *17th British Machine Vision Conference (BMVC 2006)*, Edinburgh, Scotland, 2006.

Nam T Nguyen. *Recognising Human Behaviours in Complex Environments*. PhD thesis, Curtin University of Technology, 2004a.

Nam T. Nguyen. *Recognising Human Behaviours in Complex Environments*. PhD thesis, Curtin University of Technology, 2004b.

Nam T. Nguyen, Svetha Venkatesh, Goeff West, and Hung H. Bui. Learning people movement model from multiple cameras for behaviour recognition. In *Joint IAPR International Workshops on Structural and Syntactical Pattern Recognition and Statistical Techniques in Pattern Recognition*, pages 315–324, Lisbon, Portugal, August 2004.

Nam T Nguyen, Dinh Q. Phung, H. H. Bui, and S. Venkatesh. Learning and detecting activities from movement trajectories using the hierarchical hidden markov model. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 955–960, San Diego, 2005. IEEE Computer Soceity.

F. Niu and M. Abdel-Mottaleb. View-invariant human activity recognition based on shape and motion features. In *Multimedia Software Engineering, 2004. Proceedings. IEEE Sixth International Symposium on*, pages 546–556, 2004.

U. Nodelman, C. R. Shelton, and D. Koller. Expectation maximization and complex duration distributions for continuous time bayesian networks. In *Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence*, 2005.

S. M. Oh, J. M. Rehg, and F. Dellaert. Learning and inference in parametric switching linear dynamic systems. In *International Conference on Computer Vision (ICCV)*, Beijing, China, 2005.

N. Oliver and E. Horvitz. *A comparison of HMMs and dynamic bayesian networks for recognizing office activities*. Lecture notes in computer science. 2005.

N. Oliver, E. Horvitz, and A. Garg. Layered representations for recognizing office activity. *Proceedings of the International Conference on Multimodal Interaction (ICMI 2002)*, pages 3–8, 2002a.

N. M. Oliver, B. Rosario, and A. P. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

Nuria Oliver, Eric Horvitz, and Ashutosh Garg. Layered representations for human activity recognition. *Fourth IEEE International Conference on Multimodal Interfaces (ICMI'02)*, October 2002b.

Sarah Osentoski, Victoria Manfredi, and Sridhar Mahadevan. Learning hierarchical models of activity. *IEEE/RSJ International Conference on Robots and Systems (IROS)*, 2004.

Takayuki Osogami and Mor Harchol-Balter. A closed-form solution for mapping general distributions to minimal PH-distributions. In *Int. Conf. on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation*, pages 200 – 217, September 2003.

M. Ostendorf, V. Digalakis, and O. A. Kimball. From HMMs to segment models: A unified view of stochastic modeling for speech recognition. *IEEE Transactions of Speech and Audio Processing*, 4(5):360–378, 1996.

Donald J. Patterson, Dieter Fox, Henry Kautz, and Matthai Philipose. Sporadic state estimation for general activity inference. Technical report, Intel Research Seattle and the University of Washington, July 2004.

V. Pavlovic, J. M. Rehg, and J. MacCormick. Learning switching linear models of human motion. *Neural Information Processing Systems*, 13:981âĂŞ987, 2000.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. The Morgan Kaufmann Series in Representation and Reasoning. Morgan Kaufmann Publishers, Inc., San Francisco, 1988.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. The Morgan Kaufmann Series in Representation and Reasoning. Morgan Kaufmann Publishers, Inc., San Francisco, 2nd edition, 1998.

P. Peursum, H. H. Bui, S. Venkatesh, and G. West. Human action segmentation via controlled use of missing data in hmms. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, 2004.

Dinh Q. Phung. *Probabilistic and Film Grammar Based Methods for Video Content Analysis*. PhD thesis, Curtin University of Technology, Australia, 2005a.

Dinh Q. Phung. *Probabilistic and Film Grammar Based Methods for Video Content Analysis*. PhD thesis, Curtin University of Technology, Australia, 2005b.

Dinh Q. Phung and Svetha Venkatesh. Structural unit identification and segmentation of topical content in educational videos. Technical report, Department of Computing, Curtin University of Technology, 2005. TR-May-2005.

Dinh Q. Phung, Hung H. Bui, and Svetha Venkatesh. Content structure discovery in educational videos with shared structures in the hierarchical HMMs. In *Joint Int. Workshop on Syntactic and Structural Pattern Recognition*, pages 1155–1163, Lisbon, Portugal, August 18–20 2004a.

Dinh Q. Phung, S. Venkatesh, and Hung H. Bui. Automatically learning structural units in educational videos using the hierarchical HMMs. In *International Conference on Image Processing*, Singapore, 2004b.

Dinh Quoc Phung, Svetha Venkatesh, and Chitra Dorai. High level segmentation of instructional videos based on the content density function. In *ACM International Conference on Multimedia*, pages 295–298, Juan Les Pins, France, 1-6 December 2002.

R. B. Polana. *Temporal Texture and Activity Recognition*. PhD thesis, University of Rochester, 1994.

Detlef Prescher. A short tutorial on the expectation-maximization algorithm, 2003.

Keith Price. Inference, learning human actions, human activities, human behavior. *A bibliography at http://www.visionbib.com/bibliography/motion-f741.html*, 2007.

Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Procs. IEEE*, volume 77, pages 257–286, February 1989.

C. Rao and M. Shah. View-invariant representation and learning of human action. In *Proc. IEEE Workshop on Detection and Recognition of Events in Video*, 2001.

C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. In *International Journal of Computer Vision*, volume 50, pages 203–226, 2002.

Alma Riska, Vesselin Diev, and Evgenia Smirni. Efficient fitting of long-tailed data sets into phase-type distributions. *ACM SIGMETRICS Performance Evaluation Review*, 30(3):6 – 8, 2002.

F. Rivera-Illingworth, V. Callaghan, and H. Hagras. A neural network agent based approach to activity detection in ami environments. In *The IEE International Workshop on Intelligent Environments, 2005*, pages 92–99, 2005.

W. A. Rogers, B. Meyer, N. Walker, and A. D. Fisk. Functional limitations to daily living tasks in the aged: A focus group analysis. *Human Factors*, 40(1), 1998.

H. A. Rowley, S. Baluja, and T. Kanade. Neutral network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, January 1998.

M. J. Russell and R. K. Moore. Explicit modelling of state occupancy in hidden markov models for automatic speech recognition. In *Proceedings of IEEE Conference on Acoustics Speech and Signal Processing,*, pages 5–8, March 1985.

N. Sebe, Ira Cohen, Ashutosh Garg, and Thomas S. Huang. *Machine Learning in Computer Vision*, volume 29 of *Computational Imaging and Vision*. Springer, 2005.

V. Seshadri. *The Inverse Gaussian Distribution: A Case Study in Exponential Family*. Oxford Science Publications, 1993.

K. Shearer, C. Dorai, and S. Venkatesh. Incorporating domain knowlege with video and voice data analysis. In *Workshop on Multimedia Data Minning*, Boston, USA, August 2000.

Y. Shi, Y. Huang, D. Minnen, A. Bobick, and I. Essa. Propagation networks for recognition of partially ordered sequential action. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages 862–869, 2004.

Y. Shi, A. Bobick, and I. Essa. Learning temporal sequence model from partially labeled data. In *Computer Vision and Pattern Recognition, 2006. CVPR 2006. Proceedings of the 2006 IEEE Computer Society Conference on*, 2006.

Jae-Chang Shim, Chitra Dorai, and Ruud Bolle. Automatic text extraction from video for content-based annotation and retrieval. In *International Conference on Pattern Recognition*, volume 1, pages 618–620, Brisbane, Australia, August 1998.

J. M. Siskind and Q. Morris. A maximum-likelihood approach to visual event classification. In *Proceedings of the Fourth European Conference on Computer Vision*, page 347âĂŞ360, 1996.

Cees G.M. Snoek and Marcel Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools and Applications*, 2004. In Press.

B.H. Soong and J.A. Barria. A coxian model for channel holding time distribution forteletraffic mobility modeling. *IEEE Communications Letters*, 4(2):402–404, 2000.

Asmussen Soren, Nerman Olle, and Marita Olsson. Fitting phase-type distributions via the EM algorithm. *Scandinavian Journal of Statistics*, 23(4):419–441, 1996.

Thad Starner and Alex Pentland. Real-time american sign language recognition from video using hidden markov models. *Proceedings of SCV'95*, pages 265–270, 1995.

Gilbert Strang. *Introduction to linear algebra.* Wellesley-Cambridge, Wellesly, MA, 3rd edition, 2003.

S. Stumpf, X. Bao, A. Dragunov, T.G. Dietterich, J. Herlocker, K. Johnsrude, L. Li, and J. Shen. Predicting User Tasks: I Know What You're Doing. In *National*

Conference on Artificial Intelligence (AAAI-05), Workshop on Human Comprehensible Machine Learning, 2005.

Hari Sundaram. *Segmentation, Structure Detection and Summarization of Multimedia Sequences*. PhD thesis, Columbia University, 2002.

Hari Sundaram and Shih-Fu Chang. Computable scenes and structures in films. *IEEE Transactions in Multimedia*, 4(4):482–491, 2002.

C. Sutton, A. McCallum, and K. Rohanimanesh. Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data. *The Journal of Machine Learning Research*, 8:693–723, 2007.

Emmanuel Munguia Tapia. *Activity Recognition in the Home Setting Using Simple and Ubiquitous Sensors*. PhD thesis, Massachusetts Institute of Technology, 2003.

A. Thummler and M. Telek. A novel approach for phase-type fitting with the em algorithm. *IEEE Transactions on Dependable and Secure Computing*, 3(3):245–258, 2006.

Ba Tu Truong. *An Investigation into Structural and Expressive Elements in Film*. PhD thesis, Curtin University of Technology, 2004.

Ba Tu Truong, Chitra Dorai, and Svetha Venkatesh. Automatic scene extraction in motion pictures. *IEEE Transactions in Circuits and Systems for Video Technology*, 13(1):5–10, January 2002.

D. Tweed, R. Fisher, J. Bins, and T. List. Efficient hidden semi-markov model inference for structured video sequences. *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 247–254, 2005.

Namrata Vaswani, Amit Roy Chowdhury, and Rama Chellappa. "shape activity": A continuous state hmm for moving/deforming shapes with application to abnormal activity detection. *accepted for IEEE Trans. on Image Processing*, 2004.

J. Vendrig and M. Worring. Systematic evaluation of logical story unit segmentation. *IEEE Transactions on Multimedia*, 4(4):492–499, 2002.

C. Vogler and D. Metaxas. Parallel hidden markov models for american sign language recognition. In *Seventh International Conference on Computer Vision (ICCV'99)*, volume 1, page 116âĂŞ122, 1999.

C. Vogler, H. Sun, and D. Metaxas. A framework for motion recognition with applications to americansign language and gait recognition. In *Human Motion, 2000. Proceedings. Workshop on*, pages 33–38, 2000.

C. Wang, Y. Wang, H. Liu, and Y. He. Automatic story segmentation of news video based on audio-visual features and text information. In *Int. Conf. on Machine Learning and Cybernetics*, volume 5, pages 3008–3011, 2003a.

Jihua Wang, Tat-Seng Chua, and Liping Chen. Cinematic-based model for scene boundary detection. In *The Eight Conference on Multimedia Modeling*, Amsterdam, Netherland, 5-7 November 2001.

L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, 2003b.

Shiaokai Wang, William Pentney, Ana-Maria Popescu, Tanzeem Choudhury, and Matthai Philipose. Common sense based joint training of human activity recognizers. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, 2007.

A. D. Wilson and A. F. Bobick. Parametric hidden markov models for gesture recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(9):884–900, 1999.

D. H. Wilson. *Assistive Intelligent Environments for Automatic In-Home Health Monitoring*. PhD thesis, doctoral dissertation, tech. report CMU-RI-TR-05-42, Robotics Institute, Carnegie Mellon University., 2005.

D. Wyatt, M. Philipose, and T. Choudhury. Unsupervised activity recognition using automatically mined common sense. In *Proc. AAAI*, 2005.

T. Xiang and S. Gong. Video behaviour profiling for anomaly detection. *Accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2007.

L. Xie, S.-F. Chang, A. Divakaran, and H. Sun. Learning hierarhical hidden markov models for unsupervised structure discovery from video. Technical report, Columbia University, 2002.

Lexing Xie and Shih-Fu Chang. Unsupervised mining of statistical temporal structures in video. In A. Rosenfield, D. Doreman, and D. Dementhons, editors, *Video Mining*. Kluwer Academic Publishers, June 2003.

Junji Yamato, Jun Ohya, and Kenichiro Ishii. Recognizing human action in time-sequential images using hidden markov model. *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 379–385, June 1992.

J. Yin, Q. Yang, and J. J. Pan. Sensor-based abnormal human-activity detection. *IEEE Transactions on Knowledge and Data Engineering, 18 June 2007*, 2007.

Shun-Zheng Yu and Hisashi Kobayashi. An efficient forward-backward algorithm for an explicit-duration hiden markov model. *IEEE Signal Processing Letters*, 10 (1), Jan 2003.

J. M. Zacks and B. Tversky. Event structure in perception and conception. *Psychological Bulletin*, 127(1):3–21, 2001.

D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Semi-supervised adapted hmms for unusual event detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005.

H. Zhong, M. Visontai, and J. Shi. Detecting unusual activity in video. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 819–826, Washington,, 2004.

Jun Zhu, Bo Zhang, Zaiqing Nie, Ji-Rong Wen, and Hsiao-Wuen Hon. Webpage understanding: an integrated approach. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 903–912, San Jose, California, USA, 2007.

X. Zhu, L. Wu, X. Xue, X. Lu, and J. Fan. Automatic scene detection in news program by integrating visual feature and rules. In *IEEE Pacific-Rim Conference on Multimedia*, pages 837–842, Beijing, China, 2001.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.