Curtin University
CURTIN UNIVERSITY LIBRARY
ESPACE@CURTIN
CURTIN'S INSTITUTIONAL RESEARCH REPOSITORY

Efficient Duration and Hierarchical Modeling for Human Activity Recognition

Thi Duong [a,*] Dinh Phung [a] Hung Bui [b] Svetha Venkatesh [a]

[a] *Department of Computing, Curtin University of Technology, Perth, Western Australia*

[b] *AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA, 94025, USA*

Abstract

A challenge in building pervasive and smart spaces is to learn and recognize human activities of daily living (ADLs). In this paper, we address this problem and argue that in dealing with ADLs, it is beneficial to exploit both their typical duration patterns and inherent hierarchical structures. We exploit efficient duration modeling using the novel Coxian distribution to form the Coxian hidden semi-Markov model (CxHSMM) and apply it to the problem of learning and recognizing ADLs with complex temporal dependencies. The Coxian duration model has several advantages over existing duration parameterization using multinomial or exponential family distributions, including its denseness in the space of non-negative distributions, low number of parameters, computational efficiency and the existence of closed-form estimation solutions. Further we combine both hierarchical and duration extensions of the hidden Markov model (HMM) to form the novel switching hidden semi-Markov model (SHSMM), and empirically compare its performance with existing models. The model can learn what an occupant normally does during the day from unsegmented training data and then perform online activity classification, segmentation and abnormality detection. Experimental results show that Coxian modeling outperform a range of baseline models for the task of activity segmentation. We also achieve a recognition accuracy competitive to the current state-of-the-art multinomial duration model, whilst gain a significant reduction in computation. Furthermore, cross-validation model selection on the number of phases K in the Coxian indicates that only a small K is required to achieve the optimal performance. Finally, our models are further tested in a more challenging setting in which the tracking is often lost and the set of activities considerably overlap. With a small amount of labels supplied during training in a partially supervised learning mode, our models are again able to deliver reliable performance, again with a small number of phases, making our proposed framework an attractive choice for activity modeling.

Key words: duration modeling, Coxian, hidden semi-Markov model, human activity recognition, smart surveillance.

[*] Corresponding author at: Department of Computing, Curtin University of Technology, Perth, Western Australia.
Email addresses: thi.duong@postgrad.curtin.edu.au (Thi Duong), phung@cs.curtin.edu.au (Dinh Phung), bui@ai.sri.com (Hung Bui), svetha@cs.curtin.edu.au (Svetha Venkatesh).

1. Introduction

Activity recognition is an important aspect in building pervasive smart environments. Our motivating application is the construction of a safe and smart house for the aged that facilitates automatic monitoring and support of its occupants. 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 *activities of daily living* (ADLs) through 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, and to detect if there is any deviation from the normal activity patterns to alert the caregiver if necessary.

Most of the existing work on activity recognition has focused on representing and learning sequential and temporal characteristics in activity sequences. This has led to the widespread use of dynamic models such as the Hidden Markov Model (HMM) [1] [42,40]. 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.

To overcome these limitations, two popular classes of extensions to the HMM have been proposed. The first relaxes the strong Markov assumption by modeling state duration, and the second enriches the basic HMM by introducing hierarchical structure. In the *former* effort, the semi-Markov model and its hidden variants, including explicit duration HMMs [34] and segmental HMMs [11], have been explored. In these models, a state is assumed to remain unchanged for some duration of time [2] before it transits to a new state. If the state duration distribution is non-geometric, the corresponding semi-Markov model is strictly non-Markov. Research into semi-Markov models has been an active topic since the late 1980's, driven mainly by applications in the field of speech processing and recognition. Recently, it has also gained attention in other fields, such as modeling web access traffic patterns [43], or high-level behavioral patterns in human activities [18]. The *latter* extension introduces rich stochastic models that supplement the basic HMM with a hierarchical structure, aim to exploit the natural hierarchical organization of human behaviors. Examples of these models include the Abstract HMM [4], the Hierarchical HMM [10,17,3], and the Layered HMMs [30]. Long-term dependency is captured in these models via the additional layers designed to model higher-level activities evolving at slower timescales.

Critical to a semi-Markov model is the choice of distributions for state durations. *Our first contribution* in this paper is a novel form of semi-Markov model with Coxian duration distribution. We provide its definition, algorithms for inference and learning in a dynamic Bayesian network setting, and its applications in learning and recognizing ADLs in smart environments. In most existing work, the state duration is modeled explicitly via the multinomial distribution [38,34,11,21,18]. The multinomial requires a large number of free parameters (in order of the maximum duration M, which needs to be predefined), and can be prone to overfitting if there is insufficient training data. More importantly, the burden in computation complexity (in order of $O(M)$) in both training and classification makes the multinomial an unsuitable choice for a wide range of applications, including activity recognition, where M could be arbitrarily large. More compact parameterization has been attempted to overcome this problem, including Poisson [38], Gamma [16], or more generally, the exponential family distribution [20]. Nevertheless, while keeping the number of free parameters low, these methods still suffer from the same computational problem as the multinomial (i.e., time complexity is still $O(M)$). In addition, when mapping continuous distributions (e.g., Gamma) into the discrete time domain, additional numerical approximation is required in the M-step during EM estimation (with complexity of $O(M)$) resulting in an even longer learning/classification time.

To overcome the shortcomings of existing duration parameterization, we propose the use of the Coxian distribution [24]. This distribution is a mixture of the sums of independent geometric random variables where the number of phase, K, corresponds to the number of mixture components. This type of parameterization yields an elegant solution: it has a closed-form re-estimation solution; the number of free parameters is adequately low, scaling linearly with the number of phases K, where K is typically much smaller than the

[1] A summary of all acronyms are given in Table A.1 in the appendix.

[2] or equivalently, to emit a sequence of observations.

maximum duration M in practice; and it is theoretically flexible enough in approximating any arbitrary distribution [32] while maintaining computational efficiency as well as avoiding prior specification of the maximum possible duration M. Using the (discrete) Coxian parameterization, we introduce a novel form of hidden semi-Markov model, which we term the *Coxian hidden semi-Markov model* (CxHSMM) [3].

In application of the CxHSMM to the domain of ADLs, we map primitive behaviors, such as *cooking-at-stove* or *using-the-fridge*, to the hidden states of the model. The typical duration patterns spent at each location (stove, fridge, etc.) by the occupant are modeled by the discrete Coxian distributions. The entire dynamic execution of a behavior is modeled as a hidden semi-Markov model. We apply the CxHSMM to recognize a set of relatively complex behaviors in a smart house environment and compare results with other methods of duration modeling (Poisson, Inverse Gaussian, multinomial) and a standard HMM. We demonstrate that duration information is important in activity modeling and can be effectively exploited by the Coxian parameterization. We empirically show that high accuracy can be achieved with a relatively small number of phases used in the Coxian, thus greatly reducing the number of free parameters. More importantly, it removes the computational bottleneck faced by the multinomial and other generic exponential family distributions, making the Coxian duration model an attractive choice for activity modeling.

Our *second main contribution* is a novel witching Hidden Semi-Markov Model (SHSMM), that incorporates both duration and hierarchical modeling, and its application to activity segmentation and abnormality detection in smart environments. We provide formal definitions and methods for inference and maximum-likelihood (ML) parameter learning based on its dynamic Bayesian network representation. Additionally, as a by-product of the proposed model, we present an abnormality detection scheme without the need of defining or observing abnormal data. We note that previous work [14] has also recognized the need for combining both the hierarchical and semi-Markov extensions into a unified framework. However, there has been no attempt to formulate such a model, or to empirically demonstrate the usefulness of such joint modeling over other existing methods. Our SHSMM is a result from such an effort. It is a special case of the hierarchical model with two layers [4]. The top layer is a Markov sequence of *switching* variables, while the bottom layer is a sequence of concatenated HSMMs. In a special case where the concatenated HSMMs are the CxHSMMs, the model is referred to as a Coxian Switching Hidden semi-Markov Model (CxSHSMM). Parameters of these concatenated HSMMs are determined by the switching variable at the top. Thus, the dynamics and duration parameters of the HSMM at the bottom layer are not time invariant, but are "switched" from time to time, similar to the way linear Gaussian dynamics are "switched" in a switching Kalman filter [23].

We first apply the CxSHSMM to the problem of recognizing and segmenting high-level activities. The hidden states of the bottom layer are used in the same way as in the CxHSMM, i.e., to capture atomic activities such as spending time at the cupboard, stove, fridge, or moving between these designated places. Several of these atomic activities then form high-level activities in the house such as *making-breakfast, eating-breakfast, making-coffee*, or *washing-dishes*, and each of these high-level activities is represented by a state at the top layer. Transition from one top-level state to another represents sequences of high-level activities that are typical in a human's daily routine. The experiments show that the CxSHSMM significantly outperforms the HHMM (without duration model) and the MuSHSMM (multinomial duration) [5]. Furthermore, the Coxian parameterization requires a relatively small number of phases.

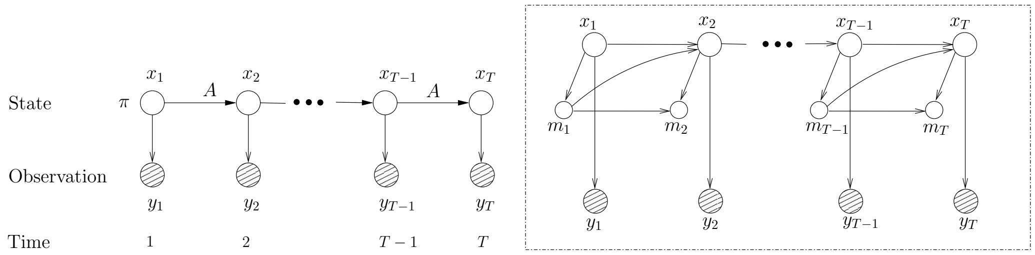
We further test the CxSHSMM in a more difficult experiment in which the object is permissible to move freely, be occluded or out of camera view, resulting in data with missing observation due to the failure of the visual tracking module. The set of activities is also more complicated in the sense that their trajectories can overlap considerably. Our results again show that it performs reasonably well in such situations. By supplying a small amount of activity labels during training, the model can achieve fairly accurate segmentation and recognition with a small number of phases required.

Finally, abnormality in the duration of activities, if detected, can provide vital clues to an alert system as it may indicate the onset of illness or sudden strokes. As the CxSHSMM can capture normal duration patterns of atomic activities spent at each location, we utilize this to construct a novel abnormality detection

[3] For quick reference, Table A.1 in the appendix provides a list of abbreviations.

[4] We note that our model can also be easily extended to a hierarchy of arbitrary depth.

[5] We note that the flat HSMM cannot be used for high-level segmentation.



(a) HMM

(b) HSMM

Fig. 1. DBN representation for a standard HMM and a standard HSMM. Shaded nodes represent observation.

scheme. We present a comprehensive set of experiments to demonstrate the performance of the model with abnormal data.

The layout of this paper is organized as follows. Section 2 introduces the readers to the duration and hierarchical extensions of the HMM. Section 3 provides a detailed discussion of the CxHSMM, including its formulation, inference and learning in its dynamic Bayesian network (DBN) structure. Section 4 develops the hierarchical model CxSHSMM including its definition, algorithms for inference and learning in DBN form. Section 5 presents the experimental results using the CxHSMM and the CxSHSMM for activity recognition and duration abnormality detection. Finally, our conclusions are presented in section 6.

2. Related Background

2.1. *The Hidden semi-Markov Model (HSMM)*

In a standard hidden Markov model [34], the (random) duration for a state can be viewed as a geometric random variable parameterized by the corresponding diagonal entry in the transition matrix. This model is often too limited in many practical applications. The semi-Markov extension overcomes this limitation by allowing more flexible duration distributions. Suppose a state i remains unchanged during time t to t' and emits an observation segment $y_{t:t'}$, if the probability of observing this segment can be factorized as $\Pr(y_{t:t'} \mid i) = \prod_{\tau=t}^{t'} \Pr(y_\tau \mid i)$, then the model is known as the explicit HSMM [34,21]. If the factorization also depends on the mean of the segment, then the model is called a segmental model [11,33]. This paper considers the former, and unless otherwise stated, the term HSMM should be understood as such. We also note that the term 'explicit' HSMM has a different meaning than in 'explicit' duration modeling, wherein the duration is modeled explicitly by a multinomial distribution.

A standard HSMM can be completely described by a state space Q, an observation alphabet set V, and a parameter set $\theta \triangleq \{\pi, A, D, B\}$. While the initial state distribution π and the observation matrix B are the same as in the standard HMM, the transition matrix A no longer allows self-transitions. In addition, the duration parameter D is explicitly introduced to specify state duration probabilities. Note that in the HMM, the self-transition probability A_{ii} for the state i defines its duration distribution: the probability that it will remain unchanged for a duration d is: $D_i^d \sim \text{Geom}(d; A_{ii}) = (A_{ii})^{d-1}(1 - A_{ii})$ where $\text{Geom}(\cdot;\cdot)$ is the geometric probability mass function. In the HSMM, this self-transition probability is set to zero at the expense of introducing a separate distribution to model the state duration D_i. Clearly, if D_i is a geometric random variable (or exponential as in continuous time case), the HSMM reduces to an HMM. Traditionally, D_i is usually modeled as the multinomial, or more generally, a member of the exponential family.

Both the HMM and HSMM can also be presented as a form of dynamic Bayesian network (DBN) [7,6] shown in Figure 1. On the right is the DBN graphical structure for HSMM with generic state duration distribution and on the left is the DBN structure for a normal HMM for comparison. At each time slice, a set of variables $\mathcal{V}_t = \{x_t, m_t, y_t\}$ is maintained where x_t is the current state, m_t is duration variable of the current state, and y_t is the current observation. The duration m_t is a counting-down variable, which not only specifies how long the current state will last, but also acts like a context influencing how the next time slice $t+1$ will be generated from the current time slice t. When $m_t > 1$, the same state x_t carries on to the next time slice; whereas when $m_t = 1$, the next state x_{t+1} 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 drawn from the distribution D_{x_t}. The variable m_{t+1} then counts down until it reaches 1.

The inference tasks for the HSMM include computing the smoothing distributions $\Pr(S_t \mid y_{1:T})$, and $\Pr(S_t, S_{t+1} \mid y_{1:T})$, where S_t is the amalgamated hidden variable: $S_t \triangleq \{x_t, m_t\}$. The inference, including scaling, is conducted using the familiar (scaled) backward/forward procedures of the HMM described in [34]. Similar to the HMM case, 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 can be estimated using the EM algorithm. Both the inference and learning tasks for the HSMM are again similar to the HMM and have been discussed in various papers [34,20,21,17,43] for different state duration probabilistic models.

The most common choice for modeling the state duration is the multinomial [34,21,43,18] due to its simplicity. Previously [34], the multinomial HSMM was extensively used in the area of speech recognition. However, there have been several recent applications in other fields. In [43], Yu *el al.* modeled and then learned the underlying process associated with the Web access traffic patterns as an explicit HSMM. Luhr et al. [18] applied the explicit HSMM to model and recognize high-level behavioral patterns in human activities. More thorough review can be found in [9].

The first drawback in using the multinomial distribution is the substantial increase in computational load. As mentioned before, the original HMM, whose state space is $|Q|$, has an inference/learning complexity of $O(|Q|^2T)$, where T is the observation length. The general approach in inference and learning in the HSMM is to treat all hidden variables as an amalgamated variable S, whose state space is $|Q|M$, where M is the maximum duration length. Thus, the theoretical complexity for the HSMM is $O(|Q|^2M^2T)$. By taking advantage of the determinism of m_t (i.e. conditionally on a given state, $m_{t+1} = m_t - 1$), the complexity can be reduced to $O(|Q|^2MT)$; or even better to $O((|Q|M + |Q|^2)T)$ by explicitly considering if x_t is in the middle of its duration or at the begin or end of its duration [43]. Nevertheless, the computational complexity for the HSMM is still significantly high, especially for large M which unfortunately could be as large as the maximum observation length T in practice.

The second drawback of the multinomial durations is the large number (i.e. $M-1$) of additional parameters required for each state. This could lead to *overfitting* when only small amount of data is available for training. In addition, M must be determined in advance. If M is set to the observation length T, the problem is then to predetermine the maximum value for T. More compact parametric distributions (e.g., the Poisson [38], the gamma [16], or more generally the exponential family [20]) have also been proposed to model the state occupancy. However, it turns out that while keeping the number of free parameters low, both discrete and continuous exponential family distributions suffer from the same computational drawback as the multinomial. This is because inference still has computational complexity that scales linearly with the maximum duration length M as these models have the same DBN representation as the multinomial HSMM (Figure 1(b)). In addition, whereas the discrete distribution parameterization (e.g., Poisson) can be estimated in a closed-form, the continuous distribution (e.g., the Gamma) requires numerical approximation during learning. Hence, the problem of effective modeling of duration is still left unresolved.

2.2. *The Hierarchical HMM (HHMM)*

Another extension to the HMM is the incorporation of hierarchical knowledge such as the hierarchical HMM (HHMM) [10], the abstract HMM (AHMM) [4], and the layered HMMs [30]. Fine *el al.* [10] were the first to introduce the HHMM, generalizing the HMM by viewing each state as an autonomous probabilistic HMM model itself. The authors apply the HHMM to the problem of learning multi-level structure in text and detect stroke patterns in handwriting. Luhr *el al.* [17] were 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. It does not allow the sharing of lower-level states by states at higher levels. Bui *et al.* [3] introduced the concept of structure sharing to allow the overlapping of common substructures in the HHMM topology, thus providing more flexibility in the model. The authors later applied it to learn movement trajectories using simulated data in [3] and real surveillance scenarios in [25].

The AHMM [4] is similarly a multi-scale probabilistic model. The original AHMM consists 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 process. The higher level policy selects the lower level ones, and the execution continues to the bottom level, where the bottom level policy does not select another policy but is modeled by a Markov chain. The observations are then generated directly from this Markov chain. At first look, the AHMM seems to act in the same manner as the HHMM. However, it extends the HHMM by allowing 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. The AHMM was first applied to activity tracking and recognition [26], and used to model movements in an indoor environments [31].

The layered HMMs in [30] can be viewed as a cascade of HMMs, where each layer is trained independently. The results of the lower layer are used as inputs to train the higher layer. The layered HMMs can be useful in reducing training and tuning requirements via re-training the lowest layer, which is the most sensitive to any changes in the environment, and keeping the higher-level layers unchanged.

The hierarchical HMM variants have been reported to successfully exploit the hierarchical structures in human activities. Nonetheless, one of their weaknesses is the lack of explicit duration models. The introduction of the SHSMM in this paper overcomes this weakness. It merges the two key extensions (hierarchy and duration) of the original HMM. The SHSMM satisfies the need of exploiting both the hierarchical decompositions and the embedded duration characteristics of human daily activities.

2.3. *Other Related Work*

Human activity recognition is a central task in video-based surveillance systems. At first, object segmentation and tracking are usually performed to extract and label human objects from the background, which are then tracked over time[6]. At a higher level, activity recognition uses tracking information to recognize behaviours, which can range from *atomic actions* such as person-walking or opening-the-door, to *higher-level activities* such as washing cloth, or cooking a meal. We distinguish the term ‘action’ and ‘activity’ to represent different levels of human behaviors; the former to denote atomic human motions (e.g, movements of the hand, head); while the latter represents higher-level tasks comprising of a sequence of combined actions, such as those activities considered in this paper. Early work in action recognition can be traced back to [42] which attempts to recognize different strokes in tennis game using the HMM. The HMM and its variants has then become popular for action recognition in several works: recognizing American Sign Language [40], action recognition and interaction [30], gesture recognition [15], body shape and gait tracking for silhouette-based human recognition [36,12].

Detecting unusual/abnormal activities in video is another important issue in surveillance systems and has been investigated in some recent work [41,5,45]. Zhong et al. [45] view normal activities as patterns that are repeated over time and develop a similarity-based framework to detect unusual activities in an unsupervised manner. The work of [41,5] uses statistical shape theory to model the shape of the object and examine its mean and dynamic deviation to spot abnormal behaviours from tracked object.

The semantics of our proposed switching HSMM is somewhat similar to the switching linear dynamic system (SLDS) proposed by [28] for the bee-dance tracking problem. While both having two layers and their top layers switch in a similar manner, they are at least different in two fundamental ways: our state spaces are discrete, whilst the SLDS is continuous at the lower level, and thus SLDS cannot model duration information; inference in ours can be done exactly, whilst that in the SLDS is intractable, and needs to be approximated. This work has recently been extended to incorporate duration at the top level [29]. However, duration is modeled explicitly as a multinomial which leads to the same complexity problems as we have outlined previously.

[6] Object segmentation and tracking, in general, is a difficult problem and is not a focus of this paper. The difficulties usually arise from camera noise, occlusion and environmental conditions and we refer to two survey papers [1,13] for further discussions on these problems.

Coxian phase-type distributions have also been used elsewhere such as in social study [19], network traffic modeling [37] or continuous time BN [27]. In [19], the authors used a Coxian to model the duration of stay of the elderly in the hospital. Based on the data collected from the patients, the model is fitted with different number of phases using a series of likelihood ratios testing to find the best fit model. The resulting best number of phases is small (equals 3) and it is consistent with the conclusion in this paper. The work in [37] considers the problem of fitting web server traffic data using the Coxian phase-type distributions. The model training method presented in that paper can be viewed as a special case of the CxSHSMM when the starting and ending indices at the top level are known. Efforts to achieve more expressive duration distribution using state-tying have also been reported [2]. Typically in such a scenario a state is 'duplicated' into K sub-states whose observation matrices are 'tied' together (i.e., share the same emission probability matrix). In particular, [2] made use of the nonnegative binomial distributions or mixtures of these. The Coxian duration model can also be viewed as a special state-tying mechanism where a state is split into K sub-states each controlling a separate Coxian phase. However, the Coxian distribution is very different from the mixture of nonnegative binomial distributions presented in [2] since the parameters for the individual geometric components are generally not identical. In addition, [2] did not provide any empirical evaluation, neither did it address the issue of model selection.

3. The Coxian Hidden Semi-Markov Model

3.1. *The Coxian Duration Model*

Recall from section 2.1 that a hidden semi-Markov model is parameterized by $\theta \triangleq \{\pi, A, D, B\}$, where π is the initial probabilities, A is the state transition probabilities, B is the emission probabilities, and D is the state duration probabilities. The duration distribution D_i of a state i is often chosen as a multinomial distribution [34,21,43], or less commonly, the exponential family [38,16,20]. However, as discussed, these modeling choices become problematic when M, the maximum duration length, is large (cf. section 2.1). Thus, we propose the use of the discrete Coxian distribution [24].

A discrete K-phase Coxian distribution [7] $\text{Cox}(\mu, \lambda)$ is defined as a mixture over sums of independent geometric random variables:

$$\text{Cox}(\mu, \lambda) = \sum_{m=1}^{K} \mu_m S_m \text{ where } \mu_m \text{ is the mixing coefficients,} \tag{1}$$

$$S_m = \sum_{i=1}^{m} X_i \text{ and } X_{i \in [1,K]} \sim \text{Geom}(\lambda_i) \tag{2}$$

The parameter μ_m specifies the prior probability of entering phase m and satisfies the constraint $0 \leq \mu_m \leq 1, \sum \mu_{m=1}^{K} = 1$. The parameter λ_m defines the probability that the phase m terminates its execution and thus $0 < \lambda_m \leq 1, \forall 1 \leq m \leq K$. The Coxian is a mixture distribution over the sums of geometric variables $S_m = X_1 + \ldots + X_m$ where X_i are independent and distributed according to a geometric distribution parameterized by λ_i, i.e., $X_i \sim \text{Geom}(\lambda_i)$.

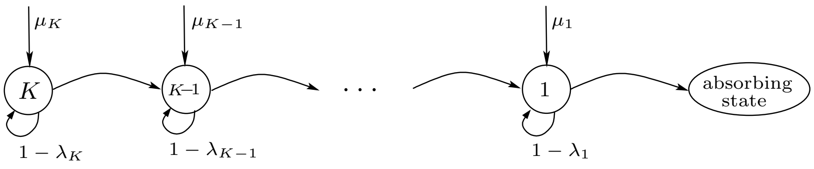


Fig. 2. The phase diagram of a discrete K-phase Coxian distribution.

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

The discrete Coxian distribution is a member of the phase-type distribution family [24] and has the following appealing interpretation. Figure 2 shows a left-to-right Markov chain with $K+1$ states numbered from K down to 1, with the self transition parameter $A_{ii} = 1-\lambda_i$ and an absorbing state. The first K states represent the K phases, while the last state is absorbing and acts like an end state. The duration of the state (phase) m is geometric: $\Pr(X_m = d) \sim \text{Geom}(d; \lambda_m) = \lambda_m(1-\lambda_m)^{d-1}$. If we start from state m, $S_m = X_m + \ldots + X_1$ is the duration of the Markov chain before the end state is reached. Thus, $\text{Cox}(\mu, \lambda)$ is in fact the distribution of the duration of this constructed Markov chain when μ is the initial state distribution.

Alternatively, the probability cumulative and probability mass functions for the Coxian can be constructed explicitly as:

$$F_{\text{Cox}}(d) = 1 - \mu^{\mathsf{T}} A^d \mathbf{I} \tag{3}$$

$$f_{\text{Cox}}(d) = \mu^{\mathsf{T}} A^{d-1} \mathbf{e} \tag{4}$$

where A is the transition matrix of the Markov chain (Figure 2) and $\mathbf{e}$ is the terminating probabilities of its phases:

$$A = \begin{bmatrix} 1-\lambda_{\mathcal{M}} & \lambda_{\mathcal{M}} & 0 & 0 & 0 \\ 0 & 1-\lambda_{\mathcal{M}-1} & \lambda_{\mathcal{M}-1} & 0 & 0 \\ 0 & 0 & \ldots & \ldots & 0 \\ 0 & 0 & 0 & 1-\lambda_2 & \lambda_2 \\ 0 & 0 & 0 & 0 & 1-\lambda_1 \end{bmatrix}, \qquad \mathbf{e} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \lambda_1 \end{bmatrix}$$

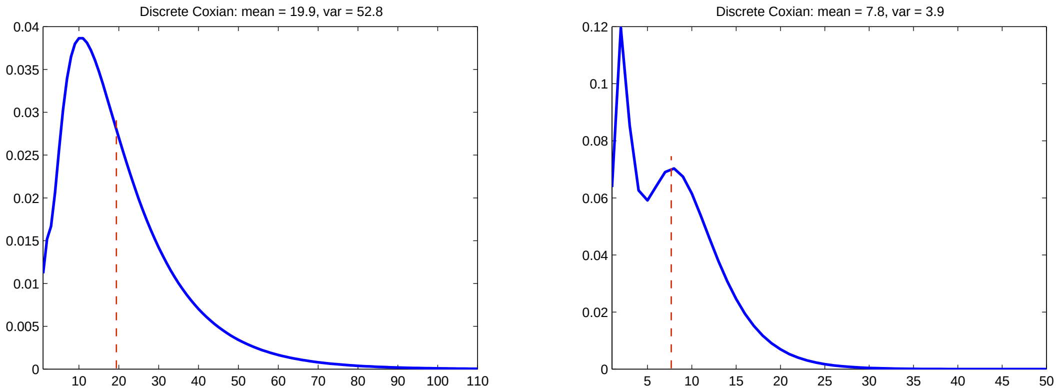
The discrete Coxian is much more flexible than the geometric distribution as its probability mass function is no longer monotonically decreasing. It is also more expressive than the nonnegative binomial distribution since it can weakly model multi-modal data. In addition the Coxian does not require a state to execute in a sequence of phases but allows entry into any arbitrary phase via the prior phase probability μ_m. Thus, it can be effective at modeling arbitrary durations. A very long duration would ideally require more phases while a short one can have as small as one phase (which, in this case, reduces to a single geometric). Figure 3 plots an example of a unimodal and a bimodal 5-phase Coxian where in the first case $\mu = (0.16\ 0.11\ 0.04\ 0.32\ 0.36), \lambda = (0.07\ 0.62\ 0.43\ 0.64\ 0.18)$ and in the second case $\mu = (0.11\ 0.25\ 0.01\ 0.31\ 0.32), \lambda = (0.58\ 0.64\ 0.46\ 0.25\ 0.41)$. The mean and variance of a Coxian distribution can also be derived in closed-form expressions [24,9]:

$$\mu_{\text{Cox}} = \sum_{m=1}^{K} \mu_m \sum_{k=1}^{m} \frac{1}{\lambda_k} \qquad \sigma^2_{\text{Cox}} = \sum_{m=1}^{K} \mu^2_m \sum_{k=1}^{m} \frac{1-\lambda_k}{\lambda^2_k} \tag{5}$$

Using the discrete Coxian distribution, we define the duration distribution for state $i \in Q$ as $D_i = \text{Cox}(\mu^i, \lambda^i)$. The parameters μ^i and λ^i are K-dimensional vectors. Finally, we term this hidden semi-Markov model as a Coxian duration HSMM (CxHSMM). We note that when $K = 1$ the model is equivalent to a HMM. The K-multinomial distribution is also a special case if all λ_i is set to 1 (in that case $\Pr(X_i = 1) = 1$, thus μ serves as the multinomial parameter).

3.2. *Dynamic Bayesian Network representation*

Figure 4(a) shows a DBN representation of the CxHSMM, in which shaded nodes are the observed variables, while clear nodes are the hidden ones. At each time slice t, a set of variables $\mathcal{V}_t = \{x_t, m_t, e_t, y_t\}$ is maintained, where x_t is the current state variable, m_t is an K-valued variable representing the current phase of x_t, e_t is a boolean-valued variable representing the ending status of x_t (i.e., $e_t = 1$ when x_t finishes

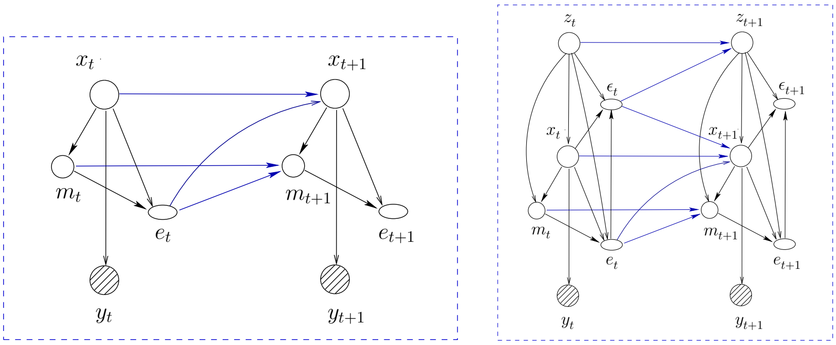


(a) unimodal

(b) bimodal

Fig. 3. Example of Coxian distributions.

its cycle or equivalently m_t leaves the last phase (i.e. phase 1); otherwise $e_t = 0$), and finally y_t is the observation returned by the system at time t. [8]



(a) CxHSMM

(b) CxSHSMM

Fig. 4. A 2-slice DBN representation for the CxHSMM and CxSHSMM.

The ending variable e_t specifies how the next time slice $t+1$ can be derived from the current time slice t given the model θ. When $e_t = 0$, the same state x_t carries on to the next time slice, whereas when $e_t = 1$, the next state x_{t+1} is drawn from the transition matrix A. In addition, the transition of the phase variables m_t follows 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 the probability of staying in the same phase is:

$$\Pr(m^1_{t+1} \,|\, m^1_t, x^i_t, e^0_t) = 1 \;\text{ for } m = 1 \tag{6}$$

$$\Pr(m^m_{t+1} \,|\, m^m_t, x^i_{t+1}, e^0_t) = 1 - \lambda^i_m \;\text{ for } m > 1 \tag{7}$$

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

$$\Pr(m^m_{t+1} \,|\, x^i_{t+1}, e^1_t) = \mu^i_m$$

Finally, $e_t = 1$ only when the m_t is in the last phase (phase 1), i.e., $\Pr(e^1_t \,|\, m^m_t, x^i_t) = 0$ if $m > 1$, and $= \lambda^i_1$ if $m = 1$. The full set of the CxHSMM 's parameters interpreted as probabilities in the DBN is is given in Table A.2 in the appendix.

[8] In general, $\{x_t, m_t, e_t\}$ are hidden and y_t is observed. In the setting of missing observation, i.e. the system fails to return its tracked data, y_t will be treated as hidden, and the framework here can be easily extended to handle this case.

3.3. *Inference and learning*

When applying the CxHSMM to modeling ADLs, we would like to learn the parameters of the CxHSMM from training data and then use the learned model for classifying *unseen* activities. Since the CxHSMM can be represented as a DBN, existing learning and inference methods for DBNs can be readily applied to our problem.

In the inference task, at time t, let $S_t \triangleq \{x_t, e_t, m_t\}$ be the amalgamated hidden state, and its realization will be written shortly as $s \triangleq \{i, k, m\}$. We then employ the familiar forward and backward procedures to compute the forward variable $\alpha_t(s) = \Pr(S_t^s, y_{1:t})$, and the backward variable $\beta_t(s) = \Pr(y_{t+1:T} \mid S_t^s)$, respectively. From α and β we compute one- and two-slice smoothing distributions, i.e. $\Pr(S_t \mid y_{1:T})$ and $\Pr(S_t, S_{t+1} \mid y_{1:T})$, which are required during EM training to compute the expected sufficient statistics for θ.

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 *scaling scheme* similar to the technique discussed in [35] for the HMM, for example, instead of calculating $\alpha_t(s)$, we calculate a *scaled* version: $\tilde{\alpha}_t(s) = \alpha_t(s)/\Pr(y_{1:t}) \triangleq \Pr(S_t^s \mid y_{1:t})$. The recursive calculation of $\tilde{\alpha}_t(s)$ is performed efficiently via dynamic programming, and in an identical fashion to that of the HMM. That results in an inference complexity of $O(|Q|^2K^2T)$, or $O(|Q|^2K^2)$ 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 $O(K)$ space instead of $O(K^2)$. This reduces the overall complexity to $O(|Q|^2KT)$ (or $O(|Q|^2K)$ 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 $K \ll M$ we can achieve significant speedup and at the same time avoided the problem of determining M in advance.

For the task of parameter learning, we use the Expectation-Maximization (EM) algorithm to learn the maximum-likelihood estimation for θ from the training data as in the HMM case. The EM estimation for a parameter τ reduces to first calculating its expected sufficient statistics (ESS), denoted as $\langle\tau\rangle$, by marginalizing out the unnecessary variables from the one- and two-slice smoothing distributions $\Pr(S_t|Y)$ and $\Pr(S_t, S_{t+1}|Y)$, and then setting the re-estimated parameter $\hat{\tau}$ to the normalized value of $\langle\tau\rangle$. We discuss here only the estimation for the Coxian duration model and leave the full set of ML-estimated formulas in Table A.4 in the appendix. Let us first look at the initial phase parameter μ_m^i in detail. The sufficient statistics (SS) of μ_m^i, denoted as $\left(\mu_m^i\right)$, is collected every time the system enters phase m right after a transition to state i, and thus: $\left(\mu_m^i\right) = \sum_{t=0}^{T-1} \mathbb{I}_{m_{t+1}}^{m} \mathbb{I}_{x_{t+1}}^{i} \mathbb{I}_{e_t}^{1}$, where the identity function $\mathbb{I}_a^b = 1$ for $a = b$, and $= 0$ for $a \neq b$. Taking the expectation of the SS over $\Pr(x, m, e \mid y_{1:T})$ results in the ESS

$$\langle\mu_m^i\rangle = E\left[\left(\mu_m^i\right)\right]_{\Pr(x,m,e \mid y_{1:T})} = \sum_{t=0}^{T-1} \Pr(x_{t+1} = i, m_{t+1} = m, e_t = 1 \mid y_{1:T})$$

which is easily obtained by marginalizing the smoothing distribution $\Pr(S_t, S_{t+1} \mid y_{1:T})$. The re-estimated formula then follows as $\hat{\mu}_m^i = \frac{\langle\mu_m^i\rangle}{\sum_{m=1}^{K}\langle\mu_m^i\rangle}$.

The individual phase's terminating probability λ_m^i needs to be treated with more care. For $m > 1$, the sufficient statistics $\left(\lambda_m^i\right)$ is counted every time the phase m is terminated within the given state i:

$$\left(\lambda_m^i\right) = \sum_{t=1}^{T-1} \mathbb{I}_{m_{t+1}}^{m-1} \mathbb{I}_{m_t}^{m} \mathbb{I}_{x_{t+1}}^{i} \mathbb{I}_{e_t}^{0}$$

Its expected sufficient statistics (ESS) follows as:

$$\langle\lambda_m^i\rangle = E\left[\left(\lambda_m^i\right)\right]_{\Pr(x,m,e \mid y_{1:T})} = \sum_{t=1}^{T-1} \Pr(m_{t+1}^{m-1}, m_t^m, x_{t+1}^i, e_t^0 \mid y_{1:T})$$

The normalization factor is obtained by marginalizing all possible values of the following phase:

$$\text{normalization} = \sum_{m' \in \{m, m-1\}} \sum_{t=1}^{T-1} \Pr(m_{t+1}^{m'}, m_t^m, x_{t+1}^i, e_t^0 \mid y_{1:T}) = \sum_{t=1}^{T-1} \Pr(m_t^m, x_{t+1}^i, e_t^0 \mid y_{1:T}) \tag{8}$$

$$\text{therefore } \hat{\lambda}_m^i = \frac{\langle \lambda_m^i \rangle}{\sum_{t=1}^{T-1} \Pr(m_t^m, x_{t+1}^i, e_t^0 \mid y_{1:T})} \tag{9}$$

For $m = 1$, λ_1^i becomes the probability that the state i has finished its duration and of course the Coxian is at its last phase. Therefore, by using the same counting and expectation procedures, we obtain: $\langle \lambda_1^i \rangle = \sum_{t=1}^{T} \Pr(e_t^1, m_t^1, x_t^i \mid y_{1:T})$. The normalized factor now is equivalent to the probability that the Coxian is at its last phase (regardless whether state i has or has not finished its duration):

$$\text{normalization} = \langle \lambda_1^i \rangle + \sum_{t=1}^{T} \Pr(e_t^0, m_t^1, x_t^i \mid y_{1:T}) = \sum_{t=1}^{T} \Pr(m_t^1, x_t^i \mid y_{1:T})$$

The re-estimated equation thus becomes:

$$\hat{\lambda}_1^i = \frac{\langle \lambda_1^i \rangle}{\sum_{t=1}^{T} \Pr(m_t^1, x_t^i \mid y_{1:T})}$$

Finally, note that the number of free parameters for the Coxian duration model is $|Q|(2K - 1)$ which is usually much smaller than $|Q|(M - 1)$ for the explicit duration model, where M can be potentially as large as T.

4. The Coxian Switching Hidden Semi-Markov Model

We now move to merge both durational and hierarchical extensions to form a novel stochastic model, termed *the Coxian Switching Hidden semi-Markov Model* (CxSHSMM). We start with a two-layer hierarchical HMM, and then describe how the Coxian duration distribution can be integrated into this model. By viewing the model as a dynamic Bayesian network, methods for inference and parameter estimation can be easily extended from the CxHSMM.

4.1. *Model definitions and parameters*

Let us consider a two-layer hierarchical HMM [10,3] defined as follows. The state space is divided into the set of states at the top level $Q^* = \{1, \ldots, |Q^*|\}$ and 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. The parameters $\pi_p^* \in [0, 1]$ and $A_{pq}^* \in [0, 1]$ are the initial and transition probabilities of a Markov chain defined over the states in Q^*. For each top-level state p, $\text{ch}(p) \subset Q$ is the set of children of p. It is possible that different parent states may share common children [3]. A transition to p at the top-level Markov chain will initiate a Markov chain at the bottom level over the states in $\text{ch}(p)$. The parameters of this p-initiated chain are given by $\{\pi_i^p, A_{ij}^p, A_{i,end}^p\}$, where $\pi_i^p \in [0, 1]$, $A_{ij}^p \in [0, 1]$ are the initial and transition probabilities as usual, and $A_{i,end}^p \in [0, 1]$ is the probability that this chain will terminate after a transition to i. Note that the stochastic constraint requires $\sum_{j \in Q} A_{ij}^p + A_{i,\text{end}}^p = 1$. At each time, an alphabet v from (discrete) observation space V is generated with a probability of $B_{v|i} \in [0, 1]$, where i is the current state at the bottom level.

In this two-layer HHMM, the duration of a bottom-level state $i \in \text{ch}(p)$, denoted as $D_{p,i}$, follows a geometric distribution. This however is too restrictive to model realistic data. We thus adapt the semi-Markov extension to allow the state duration $D_{p,i}$ to model any general distributions. More precisely, the p-initiated chain at the bottom level is now a semi-Markov sequence with $\pi_i^p, A_{ij}^p, D_{p,i}$ being the initial, transition and duration probabilities, respectively (A_{ii}^p must be zero). The termination and observation probabilities, $A_{i,\text{end}}^p$ and $B_{v|i}$, remain the same as in the two-layer HHMM. We term this two-layer structure the Switching Hidden Semi-Markov Model (SHSMM)[9] since it can be viewed as the concatenation of many HSMMs, each initiated by a different "switching" state p.

Given the disadvantages of existing duration models (multinomial and exponential family distributions), as described in section 3, we propose the use of the Coxian distribution to model state durations at the bottom level in the SHSMM, and term the new model as the Coxian Switching Hidden semi-Markov Model (CxSHSMM). For each p-initiated semi-Markov sequence, the duration distribution of a child state i is $D_{p,i} = \text{Cox}(\boldsymbol{\mu}^{p,i}, \boldsymbol{\lambda}^{p,i})$. Again, the parameters $\boldsymbol{\mu}^{p,i}$ and $\boldsymbol{\lambda}^{p,i}$ are K-dimensional vectors where K is a fixed constant representing the number of geometric phases in the discrete Coxian. Finally, note that for $K = 1$, the CxSHSMM is equivalent to a HHMM.

4.2. *Dynamic Bayesian Network representation*

Figure 4(b) shows the graphical DBN representation of the CxSHSMM over two time-slices. 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; 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[10]. The K-valued variable m_t then represents the current phase of x_t. Lastly, y_t is the observed alphabet.

The parameters of this DBN are constructed from the parameters of the CxSHSMM similar to the HHMM [3,22]. Intuitively, 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. When $e_t = 1$, there are two possibilities: if $\epsilon_t = 0$, the same top-level state carries on to the next time-slice, but the semi-Markov sequence at the bottom level transits to a new child state; if $\epsilon_t = 1$, the top-level state "switches" to the next state, and a new semi-Markov sequence is initiated at the bottom level. When $e_t = 0$, since the top state cannot switch if its current child has not ended yet, ϵ_t must be set to 0, and the same states at the top and bottom levels carry on to the next time-slice.

The state duration is modeled by a discrete Coxian, thus the transition of the phase variable m_t follows the parameters of a Coxian model as in the CxHSMM case (section 3). When $e_t = 0$ (ϵ_t must be zero), we have $m_{t+1} \in \{m_t, m_t - 1\}$, and the probability of staying in the same phase is :

$$\begin{aligned}\Pr(m^m_{t+1} \,|\, m^m_t, x^i_{t+1}, z^p_{t+1}, e^0_t) &= 1 - \lambda^{p,i}_m \;\text{ for } m > 1\\ \Pr(m^1_{t+1} \,|\, m^1_t, x^i_{t+1}, z^p_{t+1}, e^0_t) &= 1\end{aligned}$$

When $e_t = 1$, the starting phase of a new state within the same p-initialized semi-Markov sequence (if $\epsilon_t = 0$) or of a newly p-initialized semi-Markov sequence (if $\epsilon_t = 1$) is:

$$\Pr(m^m_{t+1} \,|\, x^i_{t+1}, z^p_{t+1}, e^1_t) = \mu^{p,i}_m$$

Note that a state x_t can only finish its duration ($e_t = 1$) to transit to a new state when m_t is in its last phase:

$$\Pr(e_t = 1 \,|\, m^m_t, x^i_t, z^p_t) = \begin{cases} 0, & m > 1 \\ \lambda^{p,i}_1, & m = 1 \end{cases}$$

Finally, the full set of the CxSHSMM's parameters when mapped into DBN is presented in Table A.3 in the appendix.

[9] We preliminarily introduce this model in our previous work in [8].

[10] In an HSMM, t is the end of duration of the state x_t iff $x_t \neq x_{t+1}$. However, in an CxSHSMM, 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 $e_t = \epsilon_t = 1$.

4.3. *Inference and parameter estimation*

When applying the CxSHSMM to activity modeling, we learn the parameters of the CxSHSMM from training data and then use the learned model for classifying and segmenting activities, and detecting abnormality. In the inference task, let $S_t \triangleq \{z_t, \epsilon_t, x_t, e_t, m_t\}$ be the amalgamated hidden state, and we are interested in computing the filtering distribution $\Pr(S_t \,|\, y_{1:t})$ and the smoothing distributions $\Pr(S_t \,|\, y_{1:T})$ and $\Pr(S_t, S_{t+1} \,|\, y_{1:T})$. A range of queries regarding the current high-level activity (z_t), the current atomic activity (x_t) and the remaining duration of the current activity can be answered from the marginals of these distributions. The inference including scaling is done in a similar fashion to that of the CxHSMM; 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 $O(|Q^*||Q|K)$, therefore, the recursive complexities of the smoothing distribution is $O(|Q^*|^2|Q|^2KT)$.[11] Again, if the duration is modeled by the multinomial or exponential family distributions, the complexity will be $O(|Q^*|^2|Q|^2MT)$, where M is the maximum duration length and typically $M >> K$. Thus, when the model becomes more complex (i.e. hierarchical), a greater computational factor is saved by using the Coxian duration model.

Similar to the HMM case, given a sequence of training data of the form $y_{1:T}$, the maximum likelihood parameter $\theta^* = \text{argmax}_\theta \Pr(y_{1:T} \,|\, \theta)$ can be estimated iteratively using the EM algorithm. Within each p-initiated semi-Markov chain, the re-estimation process is equivalent to that of a CxHSMM except that the explicit information about the current parent state is carried along. For example, the solution for Coxian initial phase parameter is: $\hat{\mu}^{p,i}_m = \frac{\langle \mu^{p,i}_m \rangle}{\sum_m \langle \mu^{p,i}_m \rangle}$, where $\langle \mu^{p,i}_m \rangle = \sum_{t=1}^{T-1} \Pr(m^m_{t+1}, x^i_{t+1}, z^p_{t+1}, e^1_t \mid y_{1:T}, \theta)$. The full set of re-estimated formulas is presented in the appendix.

5. Experiments

The smart environment used in our experiments is a laboratory kitchen set up as shown in figure 5. The scene is captured by two cameras mounted at two opposite ceiling corners, and a multiple-camera tracking module is used to detect movements, returning the list of positions of the *single* occupant in *x*-*y* coordinates. For modeling convenience, the kitchen is quantized into 28 square cells of $1m^2$ (shown by the crosses on the floor) and the returned *x*-*y* readings are converted into cell numbers. The low-level vision tracking module employed in this work is the same as that of [26]. This tracking module, however, sometimes returns a neighboring position instead of the actual position occupied by the person, so an observation model is estimated offline with manually labeled ground truth [26]. This corresponds to estimating the observation model B separately.

The remainder of this section is organized as follows. First, in section 5.1 we apply the CxHSMM to automatic learning and recognition of ADLs and compare its performance with other existing HSMMs and the standard HMM. The next experiment (section 5.2) aims to explore both the inherent temporal complexity and hierarchical decomposition. We employ the CxSHSMM for this task and compare it with the MuSHSMM, a 2-layer HHMM (without duration model) and a HSMM (without hierarchical model). In section 5.3 we use the learned models in section 5.2 to construct a new scheme to detect any deviation in the durations of unseen ADLs. The final set of experiment in section 5.4 reports the performance of the CxSHSMM under a more difficult scenario with missing observations and partially labeled data.

5.1. *Recognition of Activities of the Same Category*

We observe that there are several common categories of ADLs in the house (e.g., *cooking-meal, washing-dishes, ironing-clothes, leisure-reading*), in which activities of the same category generally follow the same standard procedures. For example, the *cooking-meal* category would include: *taking-food-from-fridge* → *washing-vegies/cutting-meat* → *seasoning-food* → *cooking*; or the *ironing-clothes* category would consists of: *bringing-clothes-*

[11] Note that the full joint probability of m_t and m_{t+1} is just $O(M)$ instead of $O(M^2)$ (cf. section 3.3).

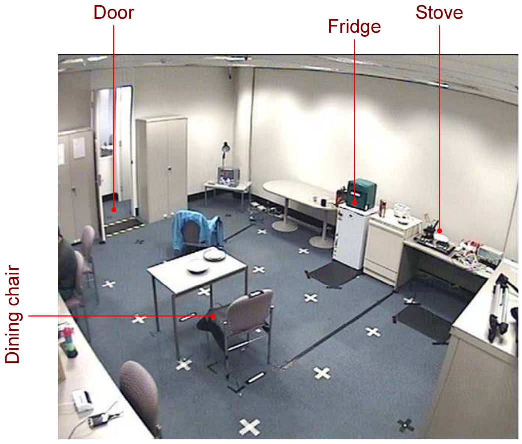


Fig. 5. The environment setup when viewed from the first camera.

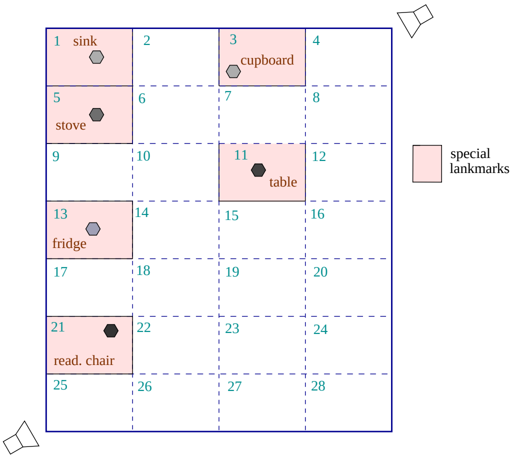


Fig. 6. The environment when mapped to a grid of 1m^2 cells.

to-laundry → *taking-out-the-iron* → *setting-up-the-iron-board* → *ironing* → *tidying-up-the-hot-iron-&-the-iron-board* → *putting-the-ironed-clothes-away*. However, the sub-activities within a given category may possess different duration characteristics. 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 morning would be much less than for *ironing-the-whole-set-of-clothes* at weekends. A *challenging problem* is to learn and distinguish ADLs of the same category mainly based on the differences in the durations of their sub-activities.

We experiment with the HSMM variants in learning and recognizing three routines of the meal preparation and consumption, and compare them with the standard HMM. In these HSMM variants, different kinds of distribution are used for modeling state durations, including the proposed Coxian, the Multinomial, the Poisson, and the Inverse Gaussian. The Multinomial was selected as it was the most popular distribution used in the HSMM, e.g., [34,21,43]. The (discrete) Poisson was chosen because of its simplicity and its good results in modeling state durations for the HSMM in speech recognition, e.g., [38]. The Inverse Gaussian was selected as an example of continuous distributions for duration modeling because it is restricted to the positive domain and has been used to model patients' staying time in hospital with successful results [39].

5.1.1. *Data descriptions*

We collect a total of 48 sequences for three activities: (a.1) *a-tea-cake-newspaper-breakfast*, (a.2) *a-scrambled-egg-on-toast-lunch*, and (a.3) *a-lasagna-salad-lunch*. We consider the 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 in duration patterns, and not in trajectories makes our task of activity classification more challenging. All the three activities follow the following twelve fixed sequential steps: *1. take-food-from-fridge* → *2. bring-food-to-stove* → *3. wash-vegetable/fill-water-at-sink* → *4. come-back-to-stove-for-cooking* → *5. take-plates/cup-from-cupboard* → *6. return-to-stove-for-food* → *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*. To give an idea about the activity lengths, Table 1 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 stove for cooking scrambled eggs on toast, which is generally longer than for reheating the lasagna ($8 - 10$(s)), or making a cup of tea ($7 - 9$(s)); having breakfast while reading the morning newspaper, $28 - 32$(s), usually requires more time at the table than simply having lunch alone, $14 - 16$(s) or $19 - 21$(s). In addition, Table 1 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. [12]). In this experiment, we consider the possibility

[12] 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) (after steeping tea) 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.

Table 1
Typical durations spent (in seconds) at the landmarks obtained from empirical data.

	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

that an occupant may visit some landmarks several times within an activity, and different activities may occasionally share the same typical durations at the same places.

5.1.2. *Training*

To ensure an objective result, we employ a *leave-one-out* cross validation 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. For model specification, we let the number of states $|Q| = 28$, equal to the number of quantized cells in the kitchen environment (figure 5), and the observation model B is obtained offline [26]. For the MuHSMM, the PsHSMM, and the IgHSMM, we equate the maximum duration M to the maximum activity length ($\sim 100 - 120$(s)), otherwise all other parameters are randomly initialized.

Model selection on the CxHSMM variants: When modeling the state duration by a Coxian distribution, we have to face the problem of choosing 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 CxHSMM, we train six different variants by varying K from 2 to 7 (note that for $K = 1$, the CxHSMM reduces to a HMM). We measure the model's cross-validated performance in terms of *classification accuracy* and *early detection rate* (defined in the next section) on *unseen* data to select the most suitable K.

5.1.3. *Experimental Results*

We compare the performance of all models (CxHSMMs, MuHSMM, PsHSMM, IgHSMM, and HMM) in Table 2 and Figure 8 based on three criteria: *classification accuracy, early detection rates (EDR),* and *running time.* For each sequence $y_{1:T}$ left out in the *leave-one-out* training selection, the likelihood $\Pr(y_{1:t} \mid \theta_i)$, for $i = 1, 2, 3$, where θ_i is the model trained with the set of activity ($\mathsf{a}.i$), is computed at each time t and used to label the most likely activity. Classification accuracy is the ratio of activities correctly labeled at $t = T$ to the total activities tested, while early detection rate is the ratio $t_0/\mathit{activityLength}$ with t_0 is the earliest time from which the activity label remains accurate.

The result shows that the HMM performs worst with only 68% in average classification accuracy; the performance of the PsHSMM is almost equally poor (69%). The IgHSMM performs comparably to a 2-phase CxHSMM with 76% and 78% accuracy respectively. Further analysis, discussed later on, shows evidences of underfitting in these cases. Starting from $K = 3$, Coxian-based models begin to increase their performances and outperform these baseline models quickly. With an additional step of parameter smoothing to avoid overfitting in the multinomial duration distribution, the MuHSMM achieves the best recognition rate of 95.56% in this experiment. The Coxian comes second at 91.39% when $K = 5$, however, it was achieved with a significant speedup (about 10 times faster than the MuHSMM in this case).

Among the Coxian models, performance varies as the number of phases increases. We observe a good performance when $K = 5$ in terms of both recognition and early detection rates as shown in Table 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 Table 1. To give an idea of how the recognition was performed, Figure 9 plots a specific example of online recognition performed by the 5-phase CxHSMM for a randomly chosen sequence of activity (a.2).

It is also interesting to note that, on comparison between the HMM and the CxHSMM, *by simply adding one more geometric phase*, i.e., extending from HMM to 2-phase CxHSMM, the model can be improved its recognition significantly (68.02% to 78.61%). *By adding a few more geometric phases* (e.g., increase K to 5), we can achieve much better performance (91.39%). The model performance slightly decreases when $K = 6$

Table 2
Classification accuracy and early detection rate (EDR) results for the CxHSMM with different number of phases versus other baseline models. EDR is measured as the percentage of the earliest detected time to the whole sequence length.

	Classification Accuracy (%)				Early Detection Rate (EDR)			
	(a.1)	(a.2)	(a.3)	Avg.	(a.1)	(a.2)	(a.3)	Avg.
HMM	88.24	62.50	53.33	68.02	9.12	37.28	42.57	29.66
PsHSMM	58.82	75.00	73.33	69.05	31.54	13.89	43.96	29.80
IgHSMM	100	56.25	73.33	76.53	7.99	47.72	31.96	29.22
MuHSMM	100	100	86.67	**95.56**	8.97	11.77	26.03	**15.59**
CxHSMM								
$K = 2$	100	62.50	73.34	78.61	7.12	31.28	41.76	26.72
$K = 3$	100	93.75	73.33	89.03	6.47	11.41	39.93	19.27
$K = 4$	94.12	75.00	80.00	85.00	8.35	31.39	56.23	31.99
$K = 5$	100	87.50	86.67	**91.39**	7.26	20.31	27.56	**18.38**
$K = 6$	100	75.00	93.00	89.44	7.70	25.99	34.47	22.72
$K = 7$	100	87.50	80.00	89.17	7.84	17.72	52.29	25.95
	(a)				(b)			

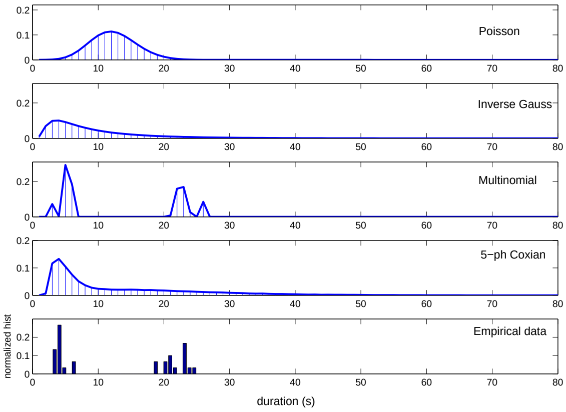


Fig. 7. Duration distributions for state “at-table” in activity (a.3) learned by different types of distribution (bottom: normalized histogram from empirical data).

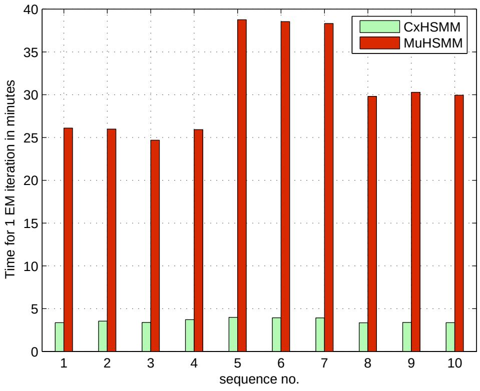


Fig. 8. EM running time comparison between a 5−phase CxHSMM and a MuHSMM.

and 7, a sign of starting to overfit the training data. Thus, $K = 5$ is the optimal number of phases selected for this experiment. Further results on the recognition performance among the activities are provided in Table A.5 in the appendix.

Regarding time complexity, the CxHSMM, as mentioned earlier, scales linearly with the standard HMM multiplied by its number of phases K, whereas the MuHSMM and the exponential family duration distribution HSMM (including PsHSMM and IgHSMM) scale by the maximum duration length M. In this experiment, K is optimal at 5, whereas M varies from 100 to 120 depending on each activity. Thus, the Coxian is faster than other baseline models by a theoretical factor of 20 to 24 times. Figure 8 shows our MATLAB computation time for one EM iteration run on ten sequences randomly chosen from activities (a.1) to (a.3). The empirical speedup 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 generally the longest[13]. It is important to note that

[13] In our Matlab implementation, the MuHSMM is coded using standard forward-backward inference algorithm where the code

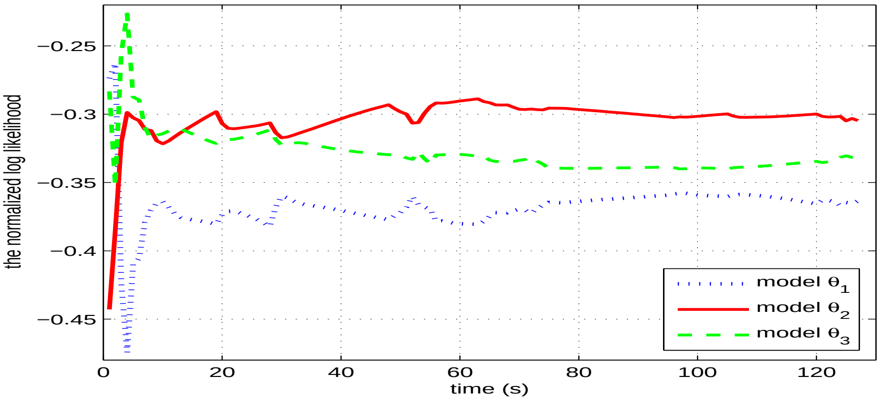


Fig. 9. Example of online recognition for an unseen sequence of activity (a.2) obtained from the 5-phase CxHSMMs trained on sets of activities (a.1) (model θ_1), activities (a.2) (model θ_2), and activities (a.3) (model θ_3). As can be seen, at at about 15s, this activity was correctly recognized by model θ_2 onward.

while the CxHSMM computation time does not increase noticeably with the activity length ((a.1) vs. (a.2)), the MuHSMM runs much slower as it moves from activities (a.1) to (a.2), taking more than 35 minutes per EM iteration. Therefore, in comparison with the PsHSMM and the IgHSMM, the CxHSMM is better not only in performance but also in running time; whereas in comparison with the MuHSMM, the CxHSMM retains a slightly worse performance but at a small fraction of the computational time. We believe that the computational speedup achieved is a very important factor for semi-Markov models to have their real-world applications as activity lengths can be arbitrarily long.

To provide some further insights on the performance of the various models, we investigate how these models have learned the state duration distributions in comparison with the empirical distribution found in the training data. Figure 7 shows the duration spent at the table in activity (a.3) learned by the PsHSMM the IgHSMM, the MuHSMM, and the 5-phase CxHSMM. Intuitively from this figure, Poisson and InvGaussian duration models have slightly underfitted the data. Being weakly multi-modal, the Coxian has learned the first dominant mode in the data well and smoothed out the less dominant one. The Multinomial was able to learn both dominant modes in the data and fitted best in this example. However, since we are comparing the fitted model with the empirical duration distribution in the training data, a good fit here does not translate to good generalization.

To illustrate this matter further, Figure 10 plots another example where the Multinomial has learned a rather 'spiky' distribution, showing a potential cause of concern for overfitting, whereas the Coxian seems to have the right fit, being able to pick up the most dominant mode and provide a smoother distribution. We also note that, in this case, multinomial parameterization would requires over 100 parameters whilst it is less than 10 for the Coxian. The result for the InvGaussian is also included as an example of underfitting.

To further illustrate the behavior of the Coxian when the number of phases changes, Figure 11 plots the learned Coxian with K ranges from 2 to 7 with the same setting as in Figure 10. For comparison, a normalized histogram is also plotted at the bottom of the figure. It can seen that as the number of phases increases, the mode of the learned Coxian gradually shifts to the right, showing sign of going from underfitting to good fitting and overfitting. Starting from $K = 5$, it matches reasonably well with the dominant mode from the empirical distribution (bottom chart, marked with $*$). As the result has shown earlier, among these Coxians, the recognition performance is also achieved best at $K = 5$.

has been optimized, taking advantage of Matlab vectorization for speedup and deterministic counting down of duration variable between two consecutive time-slices for minimizing memory allocation (cf. section 2.1)

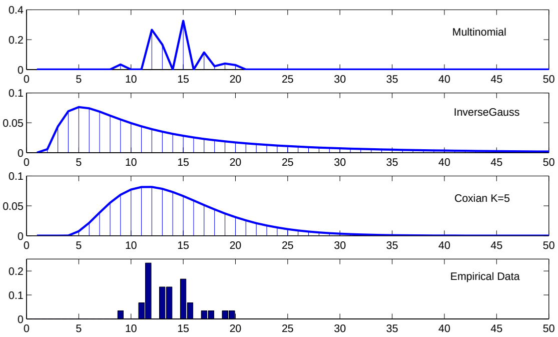


Fig. 10. Learned duration distributions for state “at-sink” in activity sequence (a.3) (bottom: normalized histogram from empirical data).

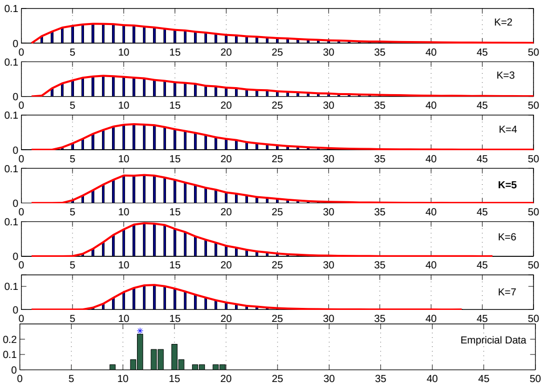


Fig. 11. (continue from Fig. 10) Learned Coxian with increasing number of phases from 2 to 7 for “at-sink” in activity (a.3).

5.2. *Recognition and Segmentation of Activities in Sequence*

In the previous section, we have experimented with flat-structured data and models. In this section, we move to tackle more complex and hierarchical data aiming to recognize and segment complex ADLs at multiple levels. Given a morning routine, consisting of sequential, but unlabeled and unsegmented ADLs (e.g., *reading-morning-newspaper, preparing-breakfast, having-breakfast*) our objective is to be able to query what the occupant is doing and when s/he changes activity. We present the results of applying the CxSHSMM and a cross-validated model selection experiment to pick the best number of phases for the CxSHSMM. The CxSHSMM's performance will be compared with a MuSHSMM and a two-layer HHMM as baseline methods.

5.2.1. *Data descriptions*

We consider a typical morning routine consisting of six high-level activities: (a.1) *entering-the-room and making-breakfast*, (a.2) *eating-breakfast*, (a.3) *washing-dishes*, (a.4) *making-coffee*, (a.5) *reading-morning-newspaper and having-coffee*, and (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 [14] in Figure 12.The shaded regular polygons 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.

The above typical 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-coffee* 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 roughly $28, 26, 16$ 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 around $5-7$(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 unlabeled, unsegmented sequences of cells are returned from the tracking module [26]. Each consists of six activities with total length of around 135 sample points. To ensure an objective evaluation, we construct three different data sets ($\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$), each consisting of 40 training and 22 testing sequences randomly partitioned from the 62 sequences.

[14] Note that the environment in Figure 12 is a quantized version of that in Figure 5.

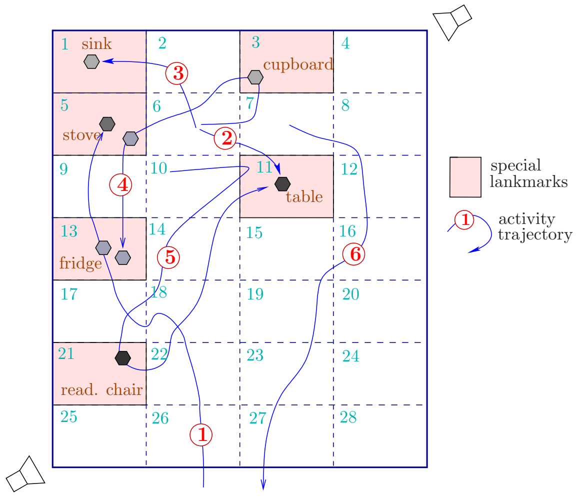


Fig. 12. The morning routine consists of activities (a.1)→(a.6). The darker the polygons the more time spent at landmarks.

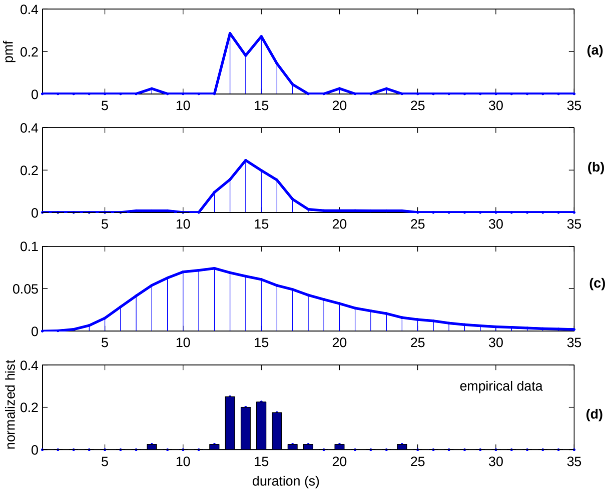


Fig. 13. Duration "at-stove" learned by a MuSHSMM: **(a)** before smoothed, **(b)** after smoothed; and **(c)** by a 5-phase CxSHSMM. Groundtruth is plotted in (d).

5.2.2. *Training*

We train three different kinds of models: various CxSHSMMs (for $K = 2, 3, \ldots, 7$), a MuSHSMM, and a two-layer 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: $|Q| = 28$. We use the estimated spatial extent of each 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 $\mathrm{ch}\mathrm{S}(p)$, or end with $\mathrm{ch}\mathrm{E}(p)$. This is done manually using the prior knowledge on the activity and environment. For example, activity (a.1) *entering-the-room* and *making-breakfast* (illustrated in Figure 12) presumably start in the door region consisting of cell 26 and any of its immediate neighbors, therefore its starting children set is $\mathrm{ch}\mathrm{S}(1) = \{21, 22, \ldots, 27\}$; activity ($a$.2) *eating-breakfast* is supposedly carried in the stove and dinning table areas, thus its set of children states is $\mathrm{ch}(2) = \{1, 2, \ldots, 15, 16\}$; and activity ($a$.3) *washing-dishes* is assumed to end when the occupant leaves the sink area, accordingly its ending children set is $\mathrm{ch}\mathrm{E}(3) = \{1, 2, 5, 6\}$. The atomic activity carried within a cell, e.g., *cooking-at-the-stove* in cell 5, is represented by a bottom-level state $i \in Q$. For the MuSHSMM, the maximum duration M is set to 35, which is the maximum time span of any individual activity (assumed to be known in advance). The same observation model as in section 5.1 is used. Except for the constraints outlined, all other parameters of these models are initialized randomly or otherwise stated, uniformly, during training.

Smoothing the multinomial duration: A simple moving-average can roughly smooth out the learned multinomial intendedly to avoid overfitting and improve the classification accuracy on unseen data for baseline methods. In addition to the learned (unsmoothed) MuSHSMM, we also report the performance of a smoothed duration version for comparison.

5.2.3. *Experimental Results*

We compare performances of the trained models (CxSHSMMs with increasing number of phases, a MuSHSMM, and a two-layer HHMM) in terms of *segmentation accuracy, early detection* and *running time* on unseen and unsegmented sequences from three data sets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. We use the learned models for segmenting and classifying segments of the test sequences into the six high-level activities. The filtering distributions $\Pr(z_t \mid y_{1:t})$ 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.

Table 3 presents the average segmentation and early detection results obtained from the three data sets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. Our first observation is that, for small number of phases ($K = 2, 3, 4$) the CxSHSMM was having trouble in distinguishing between first two activities, showing sign of underfitting, but still delivering good segmentation performance on the remaining set of activities (Table 3(a)). With $K = 2$, the segmentation

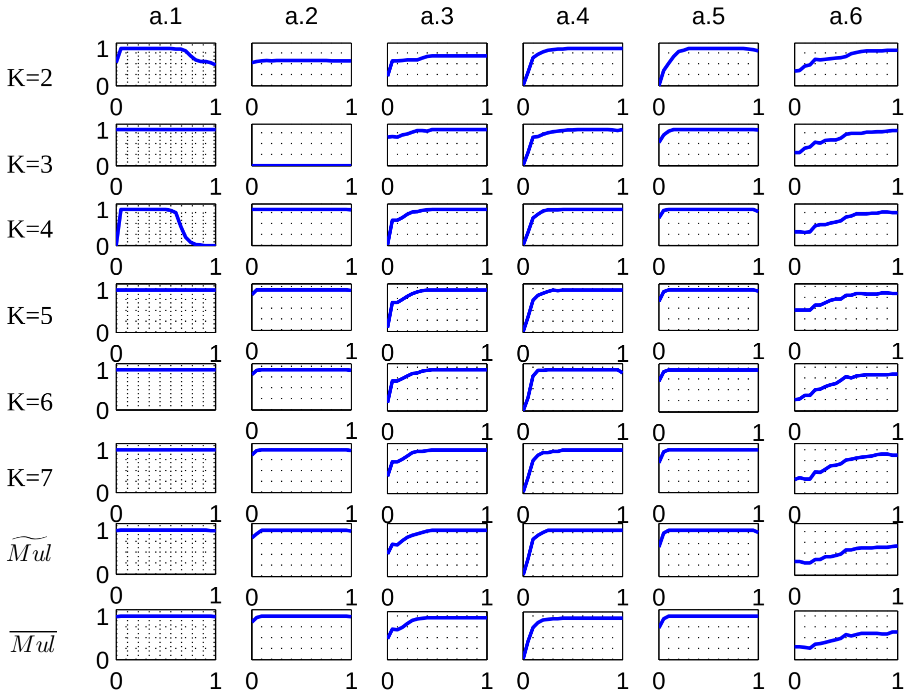


Fig. 14. Recognition accuracy averaged over three data sets obtained from the CxSHSMMs ($K = 2, \ldots, 7$), the UnS-mul ($\widetilde{Mul}$) and the S-mul ($\overline{Mul}$). The x axis shows the true segmentation of each activity from the start → the end (i.e., $0 \rightarrow 1$). The y axis shows the accuracy rate.

result was less than 70% accuracy; activity (a.1) could not be recognized when $K = 4$ and so was activity (a.2) when $K = 3$.

To illustrate this situation further, Figure 14 plots a sequence of online recognition results for different models. It shows that when $K = 2$ the CxSHSMM has occasionally segmented activity (a.1) earlier than its true ending time while the 4-phase always does so, leading to the poor performance on this activity for $K = 2$ and 4. This can be attributed by the fact that the last two states of activity (a.1) (corresponding to cells 9 and 5) are also 'shared' in the starting children set chs(2) of activity (a.2). Consequently, confusion arises between these two activities. For $K = 3$, our close examination shows that the CxSHSMM has mistakenly classified the majority of activity (a.2) as activity (a.3). One possible explanation is that, these two activities share many common children states, in addition to the fact that their starting children sets are identical.

However, starting from $K = 5$ onwards, the CxSHSMM has successfully resolved this problem and produced consistent segmentation accuracy across all activities, achieving more than 96% accuracy on average. The optimal performance is again marked at $K = 5$ in terms of segmentation accuracy (97.73%) and early detection rate (7.56%) (Table 3).

The best performance among the baseline methods is the MuSHSMM with smoothing. The segmentation accuracy is comparable to the Coxian model for the first five activities. However, it performs much poorly on the last activity, making its average performance approximately 3% lower the optimal performance of the Coxian. Finally, as expected, the two-layer HHMM, without duration knowledge, has learned a poor transition model at the high level, resulting a low performance (i.e., it occasionally detecting some activities such as (a.2) or (a.3) correctly, while generally failing to detect the others).

With respect to the running time, the filtering computations per time slice for $K = 5$ is 0.73s, improved by four times per time slice compared with the multinomial (about 3s). The theoretical time saving factor [15]

[15] In this experiment the Coxian should have been $\frac{M}{K} = \frac{35}{5} = 7$ times faster, however more coding optimization have been used to improve the speed of the MuSHSMM.

Table 3
Activity segmentation accuracy on *unseen data* with the K-phase CxSHSMMs, unsmoothed MuSHSMM (UnS-mul), smoothed MuSHSMM (S-mul), and a 2-layer HHMM.

	Segmentation accuracy of each activity (%)						
	(a.1)	(a.2)	(a.3)	(a.4)	(a.5)	(a.6)	**Avg.**
$K = 2$	56.06	66.67	80.30	100	93.94	95.45	82.07
$K = 3$	100	0	100	100	98.48	96.97	82.58
$K = 4$	0	98.48	100	100	93.94	90.91	80.56
$K = 5$	100	98.48	100	100	96.97	90.91	**97.73**
$K = 6$	100	98.48	100	92.42	100	89.39	96.72
$K = 7$	100	98.48	100	100	100	87.88	97.73
UnS-mul	98.48	98.48	100	100	95.45	65.15	92.93
S-mul	98.48	98.48	100	100	100	65.15	93.69
HHMM	19.69	100	100	19.69	77.27	68.18	64.14

(a)

	Early Detection Rate for each activity (%)						
	(a.1)	(a.2)	(a.3)	(a.4)	(a.5)	(a.6)	**Avg.**
$K = 2$	0	0.84	13.44	10.68	14.89	21.39	10.21
$K = 3$	0	NA	6.97	14.36	4.18	25.93	10.29
$K = 4$	NA	0	13.95	9.98	1.09	28.66	10.74
$K=5$	0	0.41	12.29	10.19	1.23	21.22	**7.56**
$K = 6$	0	0.46	12.94	8.88	2.68	29.76	9.12
$K = 7$	0	0.46	10.84	10.41	2.78	31.77	9.38
UnS-mul	0	0.91	11.88	9.86	2.99	36.04	10.28
S-mul	0	0.60	9.77	9.54	2.86	37.77	10.09

(b)

is given as the ratio of the maximum duration length M to the number of phases K.

We provide further insights on the performance of the different models by examining the learned parameters of the models and compare with the corresponding statistics in the training data. We found that while both the Coxian and the multinomial SHSMMs can capture the patterns in the training data adequately, the two-layer HHMM has failed to do so (Table 4). In particular, there is no significant difference between the Coxian and the multinomial. They both have learned reasonable transitions: 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). On the contrary, the HHMM has failed to capture these transitions. As a specific example, Figure 13 plots duration spent at stove in activity (a.1) (whose "true" duration is usually centered at 14(s)) learned by a 5-phase 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 seems to offer a better fit, being smoother than the multinomial model. For comparison, we have also smoothed the multinomial duration distribution using a simple moving-window averaging method.

Table 4
The learned transition matrices.

	5-phase CxSHSMM						MuSHSMM						HHMM					
Act.	(a.1)	(a.2)	(a.3)	(a.4)	(a.5)	(a.6)	(a.1)	(a.2)	(a.3)	(a.4)	(a.5)	(a.6)	(a.1)	(a.2)	(a.3)	(a.4)	(a.5)	(a.6)
(a.1)	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
(a.2)	0	0	0.8	0.2	0	0	0	0	0.8	0.2	0	0	0	0.88	0.01	0.01	0.1	0
(a.3)	0	0	0	0.8	0	0.2	0	0	0	0.8	0	0.2	0	0	0.91	0.07	0	0.02
(a.4)	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0
(a.5)	0	0	0.228	0	0.006	0.766	0	0	0.27	0	0	0.73	0	0.32	0.19	0.01	0.29	0.19
(a.6)	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0

5.3. *Duration Abnormality Detection*

Abnormality in the duration of activities, if detected, can provide important clues in alert system. For example, in the elder care domain, a person staying at a location for a longer duration than usual might indicate the onset of illness. Therefore, given a daily routine consisting of several activities in sequence, our aim is to be able to query if the occupant is successfully conducting his/her daily jobs or if can capture the normal patterns of durations spent at each location, it can also be used to detect abnormality in new activity sequences. For evaluation of abnormality detection, we capture 18 abnormal morning routine (section 5.2.1) sequences, which are also unlabeled and unsegmented. In the abnormal data, the activity trajectories are kept unchanged with respect to the normal data, 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 5.2.2 to serve as models for normal data in our abnormality detection scheme.

5.3.1. *The duration abnormality detection scheme*

We implement an online abnormality 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 MuSHSMM, we intend to set the duration parameter $\bar{D}_{p,i}$ of $\bar{\theta}_p$ to be either uniform or "inverted", where the "inverted" distribution of $Mult(\mu_n)$ is $Mult(\bar{\mu}_n)$ with $\bar{\mu}_n = \frac{\max(\mu)-\mu_n}{M*\max(\mu)-1}$. For the K -phase CxSHSMM, the duration parameter $\bar{D}_{p,i}$ is a randomly generated 2-phase Coxian which satisfies $mean(\bar{D}_{p,i}) = mean(D_{p,i}) - 0.5M$, if $mean(D_{p,i}) > 0.5M$; otherwise $mean(\bar{D}_{p,i}) = mean(D_{p,i}) + 0.5M$. 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 it is known to have a high variance [32] suiting the variable characteristics of abnormality. For comparison, we also perform abnormality detection with $\bar{D}_{p,i}$, being a randomly generated K-phase Coxian (K is the number of phase of $D_{p,i}$) whose mean is equal to that of the 2-phase Coxian $\bar{D}_{p,i}$. These two detection schemes are then compared against the background scheme, where $\bar{D}_{p,i}$ is a uniform multinomial distribution.

We argue that the abnormal model $\bar{\theta}_p$, constructed by only changing the duration model, suffices to capture abnormalities since our aim is to focus on detecting a more *subtle form of abnormality*, which is the *abnormality 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 three advantages. Firstly, it does not require addition of of new abnormal models in response to unseen data. Secondly, 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. Thirdly, there is no need to train abnormal models, which is practically difficult as abnormal data are both diverse and rare. Furthermore, by deriving an abnormal model $\bar{\theta}_p$ and taking the likelihood ratios $R_p(t)$, we can avoid the unsettling problem of having to normalize the likelihood after setting a threshold because of the uneven length in observation sequences [18]. We can examine the abnormality 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. The ability to point out when the behavior has become abnormal, or returned to normal, is equally important in issuing timely alerts to caregivers. To illustrate the capability of our model in solving this nontrivial problem, some of the 18 abnormal test sequences have only one or two activities containing abnormal durations.

5.3.2. *Online Segmentation of Abnormal Activities*

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 disturbances 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 (CxSHSMMs and MuSHSMM) to perform

Table 5
Activity segmentation on *abnormal data* with the K-phase CxSHSMM, experimented with unsmoothed (UnS-mul), and smoothed (S-mul) MuSHSMMs.

	Segmentation accuracy of each activity (%)						
	(a.1)	(a.2)	(a.3)	(a.4)	(a.5)	(a.6)	**Avg.**
$K = 2$	75.93	62.96	77.78	100	100	87.04	83.95
$K = 3$	100	0	94.44	100	100	92.59	81.17
$K = 4$	29.63	94.44	87.04	100	100	87.04	83.02
$K = 5$	100	98.15	83.33	100	100	87.04	**94.75**
$K = 6$	100	100	83.33	100	100	85.19	**94.75**
$K = 7$	100	100	83.33	100	100	87.04	**95.06**
UnS-mul	100	96.30	77.78	100	100	66.67	90.12
S-mul	100	96.30	79.63	100	100	66.67	90.43

(a)

	Early Detection Rate for each activity (%)						
	(a.1)	(a.2)	(a.3)	(a.4)	(a.5)	(a.6)	**Avg.**
$K = 2$	0	30.84	28.84	3.97	14.64	29.64	17.99
$K = 3$	0	NA	23.04	10.55	6.29	34.15	14.81
$K = 4$	0	15.98	23.54	6.67	3.46	32.35	13.67
$K = 5$	0	17.96	19.83	6.74	3.42	31.45	13.23
$K = 6$	0	14.69	20.31	5.17	2.68	34.49	12.89
$K = 7$	0	14.18	17.22	7.22	2.99	37.67	13.21
UnS-mul	0	13.18	27.44	8.10	5.91	46.41	16.84
S-mul	0	12.50	22.18	7.10	4.31	45.68	15.30

(b)

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.

Table 5 shows the average segmentation results obtained when testing against the set of 18 abnormal sequences on the models (CxSHSMMs and MuSHSMM) which were trained with three normal data sets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. Similar to the case of normal data (cf. Table 3), the CxSHSMMs with a small number of phases ($K \leq 4$) has failed to segment the activities adequately. The MuSHSMM has segmented reasonably well for set of activities $\{(\mathsf{a}.1), (\mathsf{a}.2), (\mathsf{a}.4), (\mathsf{a}.5)\}$, but failed on activity (a.6) one third of the time, and occasionally failed on activity (a.3), resulting in a performance of 90.4%. With $K \geq 5$ the CxSHSMMs performs well across all six activities with more than 94% in accuracy, demonstrating its feasibility for abnormality detection. Finally, we note that even though the CxSHSMMs perform comparably for $K = 5, 6$ and 7, when $K = 5$, it seems to offer a good trade-off between accuracy and EDR (upper bound = 31% in activity (a.6) - Table 5(b)).

5.3.3. *Duration abnormality detection with CxSHSMM*

Our objective is to find the most effective abnormality 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, which are incorrectly recognized as abnormal, to the total normal activities tested.

Figure 15 presents the Receiver Operating Characteristic (ROC) curves for the 5-phase CxSHSMM ($K > 5$ gives similar results). 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. The background uniform multinomial $\bar{D}_{p,i}$ seems to be the least affective, while the 2-phase Coxian $\bar{D}_{p,i}$ produces the considerably best ROC curve. In the region of false alarm not greater than 10% (i.e. FP $\leq$ 10%), the 2$-$phase Coxian $\bar{D}_{p,i}$ scores best with TP= 84% in comparison to 82%, and 78% from the 5$-$phase Coxian $\bar{D}_{p,i}$ and the uniform multinomial $\bar{D}_{p,i}$, respectively. Given that abnormal data is not present in the training sets, the abnormality detection rate of 84.09% is a promising result.

5.3.4. *SHSMM vs. HSMM*

We also compare the use of hierarchical SHSMMs versus a flat HSMM for abnormality detection task. Since the HSMM cannot segment the sequence into the six high-level 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. Figure 16 shows an example of a sequence comprising

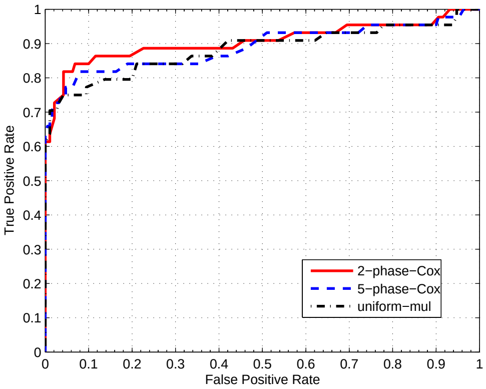


Fig. 15. ROC curves obtained from 5-phase CxSHSMM using data set $\mathcal{A}$ and its abnormal counterparts in which abnormal duration ($\bar{D}_{p,i}$) is modeled by: a 2-phase, a 5-phase Coxian, or a uniform multinomial.

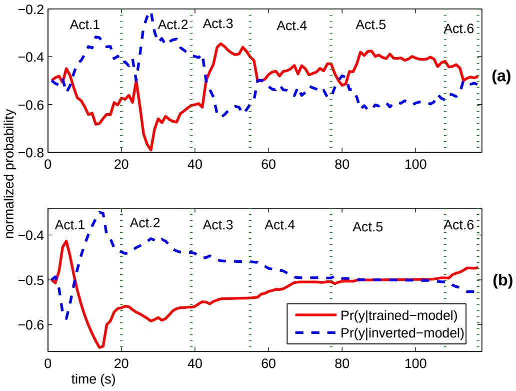


Fig. 16. Abnormality detection with: **(a)** the 5-phase CxSHSMM and its 2-phase $\bar{D}_{p,i}$ counter model, and **(b)** the flat HSMM and its "inverted" duration counter model.

activities in order (a.1) to (a.6), in which the first two activities (a.1) and (a.2) are abnormal, while the rest are normal. While the 5-phase CxSHSMM has successfully dealt with this scenario by correctly detecting only the first two activities are abnormal, the HSMM continues to label the sequence as abnormal until the sequence reaches its end. We note that the ability of the SHSMM to recognize early that activities have returned to normal is greatly important in the context of monitoring ADLs in a smart home (e.g., for the aged).

5.4. *Improvement in Activity Recognition and Segmentation with Partially Labelled Data*

In our previous experiments, we have been mindful during data capturing process so that missing trajectories are minimized. In this section, we wish to evaluate our models on a more unconstrained setting, aiming to progress towards to a more realistic setting. In this experiment, the occupant is allowed to freely move or sit wherever she or he prefers, including sitting occluded behind the table, staying still at a fixed location for longer period on the sofa, and occasionally moving fast (e.g., running) between two landmarks, or even moving out of the camera view. This setting has created a significant portion of the tracks being lost (more than 35%), and affecting every sequence recorded in the dataset. In addition to this capturing flexibility, our high-level activities share considerable overlappings in their trajectories (totally overlap in some cases), and more complicated than those considered previously in section 5.2

Our goals are, again, remaining the same as in section 5.2: classifying and segmenting ADLs in the activity sequence. In addition, under a partially supervised learning setting, a fraction of data (randomly selected) is labelled during parameter estimation phase to improve the performance. Our idea is to understand the effect of this additional labelling step in helping our models to overcome the missing trajectories. On the technical note, it can be shown that when these labels are supplied, the parameter estimation procedure presented earlier is essentially kept the same, except that the consistency over the observation is ensured by multiplying a set of identity functions. For example, if we observe the stop state $z_t = k$ in the training data, then an identity function, $\mathbb{I}^k_{z_t}$ (i.e., return 1 if $z_t = k$ and 0 otherwise) is multiplied whenever the term z_t is involved during the calculation.

5.4.1. *Data descriptions*

We 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)*: eating-dinner*, (a.4)*: relaxing-on-sofa-&-watching-tv*, (a.5)*: cleaning-stove*, (a.6)*: sweeping-floor*, and (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

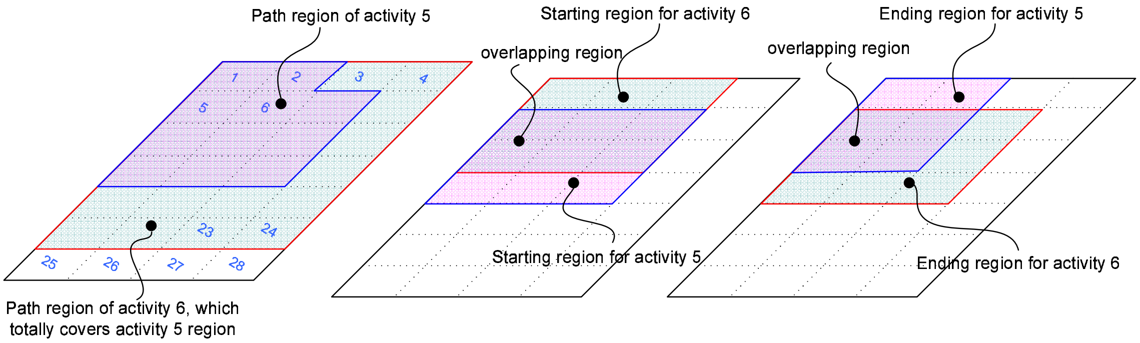

Path region of activity 5
1
2
3
4
5
6
23
24
25
26
27
28
Path region of activity 6, which totally covers activity 5 region
Starting region for activity 6
overlapping region
Starting region for activity 5
Ending region for activity 5
overlapping region
Ending region for activity 6

Fig. 17. Illustrations for path, starting, and ending regions for activity 'cleaning-stove' (a.5) and 'sweeping-floor' (a.6).

clean the stove (a.5) before/after watching television (a.4). The segmentation tasks at high-level activities is challenging, partially 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. The total evening routine is approximately 3 minutes, and the data is sampled every half of second. A total of 63 sequences are captured, in which 39 of them (accounting for about 60%) are used for training, and the remaining 24 sequences for testing. Every sequence including the unseen testing sequences has a portion of missing observations.

5.4.2. *Training*

We employ the CxSHSMM to learn data with either totally unlabelled or partially labelled (from 1% to 16% of the data), and then perform activity classification and segmentation on unseen and *unlabelled* data. Again we run the tests on different $K-$phase CxSHSMMs (for $K \in \{2, 3, \ldots, 10\}$) for model selection on the number of phases. Similar to section 5.2, 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 bottom level to the number of quantized cells in the kitchen floor $Q = 28$. The children set $\text{ch}(p)$, the starting children set $\text{ch}\textsc{s}(p)$, and the ending children set $\text{ch}\textsc{e}(p)$, for $p \in Q^*$, are then defined by our prior knowledge of the activities. There are significant overlaps between these sets for different p. For instance, Figure 17 shows the estimated spatial extents of activities (a.5)*: cleaning-stove* and (a.6)*: sweeping-floor*. We observe that $\text{ch}(5) \subset \text{ch}(6)$ as *cleaning-stove* concentrates only around the stove area while *sweeping-floor* is done on the whole floor. There are also major overlappings between $\text{ch}\textsc{s}(5)$ and $\text{ch}\textsc{s}(6)$, and between $\text{ch}\textsc{e}(5)$and $\text{ch}\textsc{e}(6)$ as sweeping starts and ends around the stove area.

5.4.3. *Experimental Results*

Similar to section 5.2, we compare the performance of different K-phase CxSHSMMs and the standard HHMM on *segmentation accuracy,* and *early detection.* Training the MuSHSMM for this experiment would take too much time: on a workstation configured with 3.2GHz CPU, 2GB memory the 5-phase Coxian took approximately 20mins per one EM iteration on one single training sequence, while the MuSHSMM took approximately 19hours (57 times slower); therefore its results are not reported. We train the CxSHSMMs and HHMM on unlabelled data, and partially labelled (with 1%, 4%, 8%, and 16%) and test them on unseen, unsegmented, and unlabelled data containing approximately 36% missing trajectories.

The results show that, even though the 3$-$phase CxSHSMM significantly perform better than the HHMM for unlabelled data, its performance was still very low and unsatisfactory (49.4%). However, when supplied with a small fraction of training labels (e.g., with just 1%), the 3$-$phase CxSHSMM dramatically increases its performance to 73% as compared with a modest rise of only 2% (from 29% to 31%) for the case of the HHMM (further results are shown in Table A.6 in the appendix). Figure 18(a) further shows that the HHMM performance remains around 60% even when supplied with up to 16% of labelled data. In contrast, with 4% labels and above, as we add in more geometric phases into the state durations ($K = 2,\ 3, \ldots$) the CxSHSMMs continue to improve their performance, stabilizing around 90% for $K \geq 4$.

In fact, with as little as 1% labels, our resuls show that the CxSHSMMs (e.g, with $K = 4, 5, 6, 9, 10$) perform reasonably well, achieving around 80% accuracy on average. Nevertheless, they have occasionally failed on some activities as illustrated by their worst performance in Figure 18(b). For example, for $K = 4$, despite of gaining 80% overall, the CxSHSMM has failed miserably on the activity ($a.5$) more than 50% of the time.

We also observe from Figure 18(a) that, with $K > 4$, there is no noticeable performance difference for the CxSHSMM when the data is labelled with 4%, 8% or 16% with an exception in segmentation accuracy when trained with 16% labelled data (Figure 18(b)). Similar conclusions are observed for comparison on early detection rate (EDR) as shown in Figure 18(c). On average, for $K \geq 4$, the CxSHSMMs can correctly identify activities around 15% to 20% of their executable time.

Finally, we again observe a consistent remark through out all of our experiments thusfar: the Coxian duration model generally requires a small number of phases to achieve its optimal performances. For this particular experiment setting, it requires a small increase in computation cost as compared with the two-layer HHMM (multiplied by factor equal to the optimal $K = 4$), but has dramatically increased the performance over all. The incorporation of both duration and hierarchical properties in our CxSHSMM model leads to reasonable results even on complicated and overlapping ADLs.

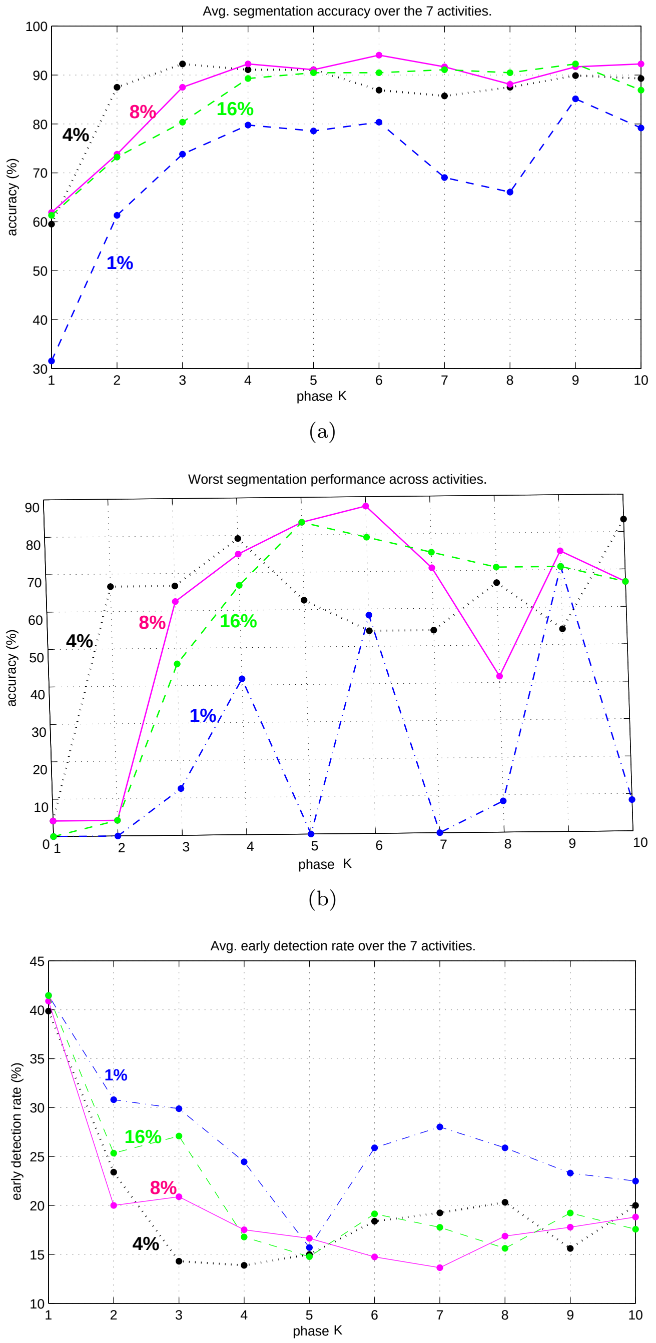
6. Conclusion

We have addressed the problem of learning and recognizing ADLs in smart homes with (hierarchical) hidden semi-Markov models. Our first main contribution is the innovative use of the Coxian distribution to efficiently model the duration information, resulting in a novel form of stochastic model, the CxHSMM, which has three significant advantages over existing models: its computational efficiency, low dimensionality of parameter space, and the existence of closed-form parameter estimation. We have then extensively applied the CxHSMM in a real-world scenario to learn and recognize a set of activities of the same category and compare its performance with various rival models. The results have shown that the CxHSMM is consistently superior to the HMM, the PsHSMM and the IgHSMM. In addition, it achieves a competitive performance close to that of the MuHSMM, whilst gaining a substantial improvement in computation time.

Our second main contribution is to combine hierarchical and duration information via a novel stochastic model, the CxSHSMM, which again uses the Coxian as the distribution for duration modeling. When applying this model to the ADLs domain, the model can learn what an occupant normally does from unsegmented training data, and then performs online activity classification and segmentation. The model is further evaluated in a difficult, noisy and unreliable tracking setting. In addition, we have also formulated abnormality detection schemes based on the trained models. We have then applied the CxSHSMM to a set of complex activities and compare its performance to various counterparts including the MuSHSMM, the two-layer HHMM (without duration knowledge), and the HSMM (without hierarchical knowledge). The improvements in both recognition rate and abnormality detection in our experiments confirm our belief that both duration and hierarchy information are crucial in the accurate modeling of ADLs; they further show that the Coxian parameterization is more robust as compared to the multinomial by having a significantly fewer number of free parameters, thus delivering more stable performances across activities. Finally, using the Coxian requires the specification of the number of phases K. To thoroughly complete our investigation, we have also experimented on a model selection setting using cross-validation. In sets of experiments with both the CxHSMM and the CxSHSMM, our results have empirically shown that high and comparable accuracy can be achieved with a relatively low number of phases ($K = 5$), thus making the Coxian an attractive model for the domain of ADLs as well as a potential model for other applications.

Acknowledgement

SRI International is supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-0185/0004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, or the Air Force Research Laboratory (AFRL).



(a)

(b)

(c)

Fig. 18. Average Segmentation and Early Detection Performance obtained from the HHMM ($K = 1$), and the CxSHSMMs for $K = 2, \ldots, 10$ trained with 1%, 4%, 8%, and 16% labelled data.

Appendix A. Summary of parameter mappings and ML estimation solutions

The inference and learning in both the CxHSMM and the CxSHSMM are formulated by viewing the models as DBN networks. Tables A.2 and A.3 list the formal definitions of the models' parameters in DBN framework; Table A.4 presents the set of their ML (maximum likelihood) estimation solutions; Table A.1

presents a list of acronyms used in the paper; finally Table A.5 and A.6 provide further results on the recognition confusion among the activities in section 5.1.3 and 5.4.

Table A.1
Summary of acronyms used in this paper.

ADLs	Activities of daily living.
HMM	Hidden Markov Model.
HSMM	Hidden semi-Markov Model.
CxHSMM	Coxian duration Hidden semi-Markov Model.
PsHSMM	Poisson duration Hidden semi-Markov Model.
SHSMM	Switching Hidden semi-Markov Model.
CxSHSMM	Coxian duration Switching Hidden semi-Markov Model.
MuSHSMM	Multinomial duration Switching Hidden semi-Markov Model.
MuHSMM	Multinomial duration Hidden semi-Markov Model.
IgHSMM	Inverse Gaussian duration Hidden semi-Markov Model.

Table A.2
CxHSMM parameter.

π_i	$= \Pr(x_1^i)$
A_{ij}	$= \Pr(x_{t+1}^j \mid x_t^i, e_t^1)$
D_i	$= \mathrm{Cox}(\boldsymbol{\mu}^i, \boldsymbol{\lambda}^i)$
μ_m^i	$= \Pr(m_{t+1}^m \mid x_{t+1}^i, e_t^1)$
$\lambda_{m>1}^i$	$= \Pr(m_{t+1}^{m-1} \mid m_t^m, x_{t+1}^i, e_t^0)$
λ_1^i	$= \Pr(e_t^1 \mid m_t^1, x_t^i), \quad m = 1$
$B_{v\mid i}$	$= \Pr(y_t^v \mid x_t^i)$

Table A.3
CxHSMM parameter.

$\pi_p^* = \Pr(z_1^p), \quad A_{pq}^* = \Pr(z_{t+1}^q \mid z_t^p, \epsilon_t^1)$
$\pi_i^p = \Pr(x_{t+1}^i \mid z_{t+1}^p, \epsilon_t^1, e_t^1)$
$A_{ij}^p = \Pr(x_{t+1}^j, \epsilon_t^0 \mid z_{t+1}^p, x_t^i, e_t^1), \quad A_{i,\mathrm{end}}^p = \Pr(\epsilon_t^1 \mid z_t^p, x_t^i, e_t^1)$
$D_{p,i} = \mathrm{Cox}(\boldsymbol{\mu}^{p,i}, \boldsymbol{\lambda}^{p,i}), \quad \mu_m^{p,i} = \Pr(m_{t+1}^m \mid x_{t+1}^i, z_{t+1}^p, e_t^1)$
$\lambda_{m>1}^{p,i} = \Pr(m_{t+1}^{m-1} \mid m_t^m, x_{t+1}^i, z_{t+1}^p, e_t^0)$
$\lambda_1^{p,i} = \Pr(e_t^1 \mid m_t^1, x_t^i, z_t^p), \quad m = 1$
$B_{v\mid i} = \Pr(y_t^v \mid x_t^i)$

References

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, 1999.

[2] J.A. Bilmes. What HMMs Can Do. *IEICE Transactions on Information and Systems*, pages 869–891, 2006.

[3] 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.

[4] 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.

[5] R. Chellappa, N. Vaswani, and A. Roy Chowdhury. Activity modeling and recognition using shape theory. In *Behavior Representation in Modeling and Simulation*, 2003.

[6] P. Dagum and A. Galper. Time series prediction using belief network models. *International Journal of Human-Computer Studies*, 42:617–632, 1995.

[7] Thomas Dean and Jeiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(3):142 – 150, 1989.

[8] 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, 2005. IEEE Computer Society.

[9] T.V. Thi Duong. *Efficient Duration Modelling in the Hierarchical Hidden Semi-Markov Models and Their Applications.* PhD thesis, Department of Computing, Curtin University of Technology, 2008.

[10] Shai Fine, Yoran Singer, and Nftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.

Table A.4
Maximum Likelihood (ML) estimation solutions.

ML estimation for the CxHSMM	
Re-estimation	Expected Sufficient Statistics
$\hat{\pi}_i = \left\langle \pi_i \right\rangle / \sum_i \left\langle \pi_i \right\rangle = \left\langle \pi_i \right\rangle$	$\left\langle \pi_i \right\rangle = \Pr(x_1^i \mid y_{1:T})$
$\hat{A}_{ij} = \left\langle A_{ij} \right\rangle \Big/ \sum_j \left\langle A_{ij} \right\rangle$	$\left\langle A_{ij} \right\rangle = \sum_{t=1}^{T-1} \Pr(x_{t+1}^j, x_t^i, e_t^1 \mid y_{1:T})$
$\hat{\mu}_m^i = \left\langle \mu_m^i \right\rangle / \sum_{m=1}^K \left\langle \mu_m^i \right\rangle$	$\left\langle \mu_m^i \right\rangle = \sum_{t=0}^{T-1} \Pr(m_{t+1}^m, x_{t+1}^i, e_t^i \mid y_{1:T})$
$\hat{\lambda}_m^i = \begin{cases} \left\langle \lambda_m^i \right\rangle / \sum_{t=1}^{T-1} \Pr(m_t^m, x_{t+1}^i, e_t^0 \mid y_{1:T}) & m > 1 \\ \left\langle \lambda_m^i \right\rangle / \sum_{t=1}^{T} \Pr(m_t^m, x_t^i \mid y_{1:T}) & m = 1 \end{cases}$	$\left\langle \lambda_m^i \right\rangle = \begin{cases} \sum_{t=1}^{T-1} \Pr(m_{t+1}^{m-1}, m_t^m, x_{t+1}^i, e_t^0 \mid y_{1:T}) & m > 1 \\ \sum_{t=1}^{T} \Pr(e_t^1, m_t^m, x_t^i \mid y_{1:T}) & m = 1 \end{cases}$
$\hat{B}_{v\mid i} = \left\langle B_{v\mid i} \right\rangle / \sum_v \left\langle B_{v\mid i} \right\rangle$	$\left\langle B_{v\mid i} \right\rangle = \sum_{t=1}^{T} \Pr(x_t^i \mid y_{1:T}) \mathbb{I}_{y_t}^v$
ML estimation for the CxSHSMM	
$\hat{\pi}_p^* = \left\langle \pi_p^* \right\rangle \Big/ \sum_p \left\langle \pi_p^* \right\rangle$	$\left\langle \pi_p^* \right\rangle = \Pr(z_1^p \mid y_{1:T})$
$\hat{A}_{pq}^* = \left\langle A_{pq}^* \right\rangle \Big/ \sum_q \left\langle A_{pq}^* \right\rangle$	$\left\langle A_{pq}^* \right\rangle = \sum_{t=1}^{T-1} \Pr(z_{t+1}^q, z_t^p, \epsilon_t^1 \mid y_{1:T})$
$\hat{\pi}_i^p = \left\langle \pi_i^p \right\rangle / \sum_i \left\langle \pi_i^p \right\rangle$	$\left\langle \pi_i^p \right\rangle = \sum_{t=0}^{T-1} \Pr(x_{t+1}^i, z_{t+1}^p, \epsilon_t^1, e_t^1 \mid y_{1:T})$
$\hat{A}_{ij}^p = \left\langle A_{ij}^p \right\rangle \Big/ \left[\sum_j \left\langle A_{ij}^p \right\rangle + \left\langle A_{i,\text{end}}^p \right\rangle \right]$	$\left\langle A_{ij}^p \right\rangle = \sum_{t=1}^{T-1} \Pr(x_{t+1}^j, x_t^i, z_{t+1}^p, \epsilon_t^0, e_t^1 \mid y_{1:T})$
$A_{i,\text{end}}^p = \left\langle A_{i,\text{end}}^p \right\rangle \Big/ \left[\sum_j \left\langle A_{ij}^p \right\rangle + \left\langle A_{i,\text{end}}^p \right\rangle \right]$	$\left\langle A_{i,\text{end}}^p \right\rangle = \sum_{t=1}^{T-1} \Pr(\epsilon_t^1, x_t^i, z_t^p, e_t^1 \mid y_{1:T})$
$\hat{\mu}_m^{p,i} = \left\langle \mu_m^{p,i} \right\rangle / \sum_m \left\langle \mu_m^{p,i} \right\rangle$	$\left\langle \mu_m^{p,i} \right\rangle = \sum_{t=0}^{T-1} \Pr(m_{t+1}^m, x_{t+1}^i, z_{t+1}^p, e_t^1 \mid y_{1:T})$
$\hat{\lambda}_m^{p,i} = \begin{cases} \dfrac{\left\langle \lambda_m^{p,i} \right\rangle}{\sum_{t=1}^{T-1} \Pr(m_t^m, x_{t+1}^i, z_{t+1}^p, e_t^0 \mid y_{1:T})} & m > 1 \\ \dfrac{\left\langle \lambda_m^{p,i} \right\rangle}{\sum_{t=1}^{T} \Pr(m_t^m, x_t^i, z_t^p \mid y_{1:T})} & m = 1 \end{cases}$	$\left\langle \lambda_m^{p,i} \right\rangle = \begin{cases} \sum_{t=1}^{T-1} \Pr(m_{t+1}^{m-1}, m_t^m, x_{t+1}^i, z_{t+1}^p, e_t^0 \mid y_{1:T}) & m > 1 \\ \sum_{t=1}^{T} \Pr(e_t^1, m_t^m, x_t^i, z_t^p \mid y_{1:T}) & m = 1 \end{cases}$
$\hat{B}_{vi} = \left\langle B_{v\mid i} \right\rangle / \sum_v \left\langle B_{v\mid i} \right\rangle$	$\left\langle B_{v\mid i} \right\rangle = \sum_{t=1}^{T} \Pr(x_t^i \mid y_{1:T}) \mathbb{I}_{y_t}^v$

[11] 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.

[12] J. Gao and J. Shi. Multiple frame motion inference using belief propagation. In *The 6th International conference on Automatic Face and Gesture Recognition*, 2004.

[13] D. M Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[14] Henry Kautz, Oren Etzioni, Dieter Fox, and Dan Weld. Foundations of assisted cognition systems. Technical report, University of Washington, CSE, March 2003.

[15] Hyeon-Kyu Lee and Jin H. Kim. An HMM-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):961–973, 1999.

[16] Stephen E. Levinson. Continuously variable duration hidden markov models for automatic speech recognition. *Computer Speech and Language*, 1(1):2945, 1986.

[17] 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.

[18] 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.

[19] Adele H. Marshall and Sally 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.

[20] 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, 1993.

[21] Carl Mitchell, Mary Harper, and Leah Jamieson. On the complexity of explicit duration HMMs. *IEEE Transactions on Speech and Audio Processing*, 3(3), 1999.

[22] 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.

[23] Kelvin Murphy. Learning switching kalman filter models. Technical report, Campaq Cambridge Research Lab, 1998.

[24] Marchel F. Neuts. *Matrix-Geometric Solutions in Stochastic Models.* The Johns Hopkins University Press, Baltimore and London, 1981.

[25] 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.

Table A.5
Further classification confusion among the activities for different models presented in Section 5.1.3.

	$K = 2$ (avg.78.61%)			$K = 3$ (avg.89.03%)			$K = 4$ (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

	$K = 5$ **(avg.91.39%)**			$K = 6$ (avg.89.44%)			$K = 7$ (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

(a) Classification results for different K-phase 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**

(b) Classification results for other models.

Table A.6
Confusion matrices showing segmentation accuracy across the 7 activities for the HHMM and 3-phase CxSHSMM presented in section 5.4.

HHMM (Avg. 29.17%)	3−phase CxSHSMM(Avg. 49.40%)
$\begin{bmatrix} 25.0 & 0 & 0 & 0 & 75.0 & 0 & 0 \\ 0 & 12.5 & 0 & 87.5 & 0 & 0 & 0 \\ 0 & 0 & 4.2 & 95.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 12.5 & 0 & 87.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 37.5 & 0 & 0 & 62.5 \end{bmatrix}$	$\begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8.3 & 79.2 & 8.3 & 0 & 0 & 4.2 & 0 \\ 0 & 0 & 8.3 & 79.2 & 0 & 12.5 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 4.2 & 16.7 & 8.3 & 66.7 & 0 & 4.2 & 0 \\ 4.2 & 0 & 0 & 95.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 41.2 & 0 & 0 & 58.3 \end{bmatrix}$

Trained with 1% labelled data	
HHMM (Avg. 31.55%)	3−phase CxSHSMM(Avg. 73.81%)
$\begin{bmatrix} 25.0 & 0 & 0 & 0 & 75.0 & 0 & 0 \\ 0 & 37.5 & 0 & 62.5 & 0 & 0 & 0 \\ 0 & 0 & 29.2 & 70.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 16.7 & 4.2 & 79.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 4.2 & 0 & 0 & 29.2 & 0 & 37.5 & 29.2 \end{bmatrix}$	$\begin{bmatrix} 95.8 & 4.1667 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 45.8 & 54.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 95.8 & 0 & 4.2 & 0 \\ 0 & 8.3 & 58.3 & 20.8 & 12.5 & 0 & 0 \\ 0 & 0 & 0 & 29.2 & 0 & 70.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.2 & 95.8 \end{bmatrix}$

[26] 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, 2004.

[27] Uri Nodelman, Christian R. Shelton, and Daphne Koller. Expectation maximization and complex duration distributions for continuous time bayesian networks. In *Procs. of the 21st International Conference on Uncertainty in Artificial Intelligence*, pages 421–430, 2005.

[28] Sang Min Oh, James M. Rehg, Tucker Balch, and Frank Dellaert. Learning and inference in parametric switching linear dynamic systems. In *International Conference on Computer Vision (ICCV-2005)*, Beijing, China, 2005.

[29] Sang Min Oh, James M. Rehg, and Frank Dellaert. Parameterized duration modeling for switching linear dynamic systems. In *International Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, New York, USA, 2006.

[30] Nuria Oliver, Eric Horvitz, and Ashutosh Garg. Layered representations for human activity recognition. *Fourth IEEE International Conference on Multimodal Interfaces (ICMI'02)*, 2002.

[31] Sarah Osentoski, Victoria Manfredi, and Sridhar Mahadevan. Learning hierarchical models of activity. *IEEE/RSJ International Conference on Robots and Systems (IROS)*, 2004.

[32] 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, 2003.

[33] 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.

[34] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Procs. IEEE*, volume 77, pages 257–286, 1989.

[35] Lawrence R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[36] Cen Rao, Alper Yilmaz, and Mubarak Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2):203–226, 2002.

[37] Alma Riska, Mark Squillante, Shun-Zheng Yu, Zhen Liu, and Li Zhang. Matrix-analytic analysis of a map/ph/1 queue fitted to web server data. In G. Latouche and P. Taylor, editors, *Matrix-Analytic Methods: Theory and Applications*, pages 335–356. World Scientific, 2002.

[38] 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, 1985.

[39] V. Seshadri. *The Inverse Gaussian Distribution: A Case Study in Exponential Family.* Oxford Science Publications, 1993.

[40] Thad Starner and A. Pentland. Visual recognition of american sign language using hidden Markov models. pages 184–194, Int. Workshop on Automatic Face and Gesture Recognition, 1995.

[41] 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.

[42] 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, 1992.

[43] Shun-Zheng Yu and Hisashi Kobayashi. An efficient forward-backward algorithm for an explicit-duration hiden markov model. *IEEE Signal Processing Letters*, 10(1), 2003.

[44] J. M. Zacks and B. Tversky. Event structure in perception and conception. *Psychological Bulletin*, 127(1):3–21, 2001.

[45] 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.