Hidden Markov Models and their Extensions for Proportional Sequential Data

Samr Ali

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

PhD (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

May 2021

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Samr Ali**

Entitled: **Hidden Markov Models and their Extensions for Proportional Sequential Data**

and submitted in partial fulfillment of the requirements for the degree of

Doctor Of Philosophy (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality. Signed by the Final Examining Committee:

_____ Chair
Dr. Alex De Visscher

_____ External Examiner (to University)
Dr. Faïcal Chamroukhi

_____ External Examiner (to Program)
Dr. Tristan Glatard

_____ Examiner
Dr. Jamal Bentahar

_____ Supervisor
Dr. Nizar Bouguila

Approved by _____
Dr. Yousef R. Shayan, Chair
Department of Electrical and Computer Engineering

13th April 2021

_____
Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Hidden Markov Models and their Extensions for Proportional Sequential Data

Samr Ali, Ph.D.

Concordia University, 2021

We are facing an all-time high in the worldwide generation of data. Machine learning techniques have proven successful in unveiling patterns within data to further human knowledge. This includes building systems with overall better prediction and accuracy levels. Nonetheless, many areas have yet to be studied which warrants further exploitation of these techniques. Hence, data modeling is one of the topics at the forefront of scientific research. A particularly interesting field of research is the appropriate choice of distribution that corresponds to the nature of the data.

In this thesis, we focus on tackling challenges in the approximation of proportional Hidden Markov Models (HMM). We review the main concepts behind HMM; one of the cornerstone probabilistic graphical models for time series or sequential data. We also discuss various modern challenges that exist when training or using HMMs. Nonetheless, we primarily focus on the notorious model estimation process of HMMs as well as the appropriate choice of emission distribution based on the nature of the data. We have tackled these problems using variational inference and Maximum A Posteriori (MAP) approximation with the Dirichlet, the Generalized Dirichlet, and the Beta-Liouville (BL) distributions-based HMMs for proportional data. In this thesis, we develop frameworks for learning these proportional HMMs that have been proposed recently as an efficient way for modeling sequential proportional data. In contrast to the conventional Baum Welch algorithm, commonly used for learning HMMs, the proposed algorithms place priors for the learning of the desired parameters; hence, regularizing the estimation process. We also extend these models into infinity for a data-driven dynamically chosen structure of HMMs. Such a setup enables flexibility in the model structure with a lower computational cost for model selection. We also investigate

the fusion of the trained classifiers and witness a consequent improved performance. Moreover, we incorporate a simultaneous feature selection paradigm as well as investigate online deployment. We present our recently proposed methodologies that address the aforementioned problems and discuss the achieved results across a variety of computer vision applications. We also present how a simple novel experimental setup can drastically improve the performance of HMMs in occupancy detection, and estimation by extension, in smart building for an applied research contribution. Finally, we conclude and recommend potential future work.

Acknowledgments

First, I would like to thank Allah Al Mighty the Most Merciful for all His blessings and the many graces that enabled me to finalize my doctorate.

I am very grateful to my supervisor, Prof. Nizar Bouguila, who believed in me from day 1. It has been an interesting journey and I have certainly learned a lot. Thank you for always giving me the time to discuss with you the many questions that I had and the advice that I needed. Thank you for your mentorship and support. Thank you for the lessons. I hope that we continue to collaborate together.

My utmost thanks goes to many institutes for the various scholarships, opportunities, and support. I would like to sincerely thank the Fonds de recherche du Québec – Nature et technologies (FRQNT) for my doctoral scholarship, Ericsson Global Artificial Intelligence Accelerator (GAIA) for my internship funding along with Mitacs for its accelerate fellowship. Finally, thank you Concordia University for my tuition award of excellence.

I would like to also greatly thank my committee members for all their efforts and time. Their notes, comments, and our discussions have been invaluable for the improvement of my work. This thesis could not be the way it currently is without your input and feedback. I thank you a lot for your continuous support and much appreciated excellent input.

They say that some people come in your life for an episode to teach you a lesson and I have certainly met my share during my studies. Thank you for showing me how strong I am and how limitless I can be. To my friends who have supported me, I extend my utmost thanks and wish you all the best in your endeavours. I also appreciate the inspiring voices that reached out to me with their scientific writing.

Words can not describe how grateful and thankful I am for my family's continuous support and endless understanding. I hope that you know how much you mean to me and I dedicate this thesis to you. To my parents who always listened to me and sincerely advised, to my sister for her 24/7 availability and her light-heartedness, and to my brother who joined me for a period here in Montreal and his thoughtfulness. May Allah protect and bless you always!

To future female doctorates: I urge to soar as you ought to. Throughout my journey thus far, I have met many who tried to label me according to their own limited perceptions and past experiences. You will face many of them, those who will try to clip your wings; who will induce you to think that if you are good in one thing, you can not possibly have the skills for others. Do not listen to them and do not let your doubts rule you. Whatever you dream you can achieve, you can! You can excel in many fields even if others have not before. As a matter of fact, I encourage you to. Dare to create a new path and to inspire others to follow.

Finally, though I can not be happier for successfully completing one of my life goals. A sobering saying comes to mind, "This too shall pass." It is one that has been my companion throughout this PhD journey, a consolation at times and a warning at others, and I would be remiss not to remind myself of it now. Tomorrow brings a new challenge and with it I will rise to the occasion. Here is to hope, joy, and happiness!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> "Begin at the beginning," the King said, gravely, " and go on till you come to the end; then stop."
>
> Lewis Carroll, *Alice in Wonderland*

Hidden Markov models (HMM) have drawn research interest in the past decade. This is due to its now perceived capability in a variety of applications that extend beyond the originally investigated speech-related tasks [3]. Indeed, examples include recognition of handwritten characters, musicology, stock market forecasting, predicting earthquakes, video classification, surveillance systems, and network analysis.

HMMs are probabilistic models that fall under the generative machine learning algorithms category. Generally, data modeling techniques in machine learning classically fall under two main categories: discriminative or generative. Generally, discriminative models are trained to infer a mapping between data inputs x to class labels y, while generative models first learn the distribution of the classes before predictions are made [4]. Mathematically, the former represents the posterior probability $p(y \mid x)$ with the latter denoting the joint probability $p(x, y)$ that is used to calculate the posterior accordingly for the classification. Each of the models have their own properties and advantages which we summarize some of shortly.

Discriminative models usually achieve superior classification accuracy results due to their primary learning objective of the boundary between classes [5]. These include the famous Support Vector Machines (SVM) and decision tree classifiers. On the other hand, generative models require

less training data, can be used for outlier detection, and provide the ability to generate more training data with the same input distribution upon completion of the training of the model. Mixture models are another example of generative data models. An interested reader is referred to [4, 6] for further discussions. Hybrid models with HMMs are also possible such as in [7, 8]; however, this falls outside of our discussion.

In a manner of speaking, HMMs may be considered as an extension of mixture models along the temporal axis. That is they are capable of spatio-temporal modeling whereby both the space and time features may be taken into consideration. As expected, this leads to better performances as well as an explainable machine learning pipeline in applicable fields.

A HMM is a powerful machine learning model due to its inherent ability to capture spatio-temporal patterns in data. In contrast to time series models, such as the famous Autoregressive Integrated Moving Average (ARIMA) [9], both spatial and temporal dimensions are simultaneously taken into account in the modeling process. This enforces its wide applicability and motivates further research into its various properties and fields.

In this thesis, we concentrate on the following sub-areas of research for the HMM:

(1) Efficient learning of the parameters of the model. Traditionally, this is carried out by the Baum Welch algorithm. Instead, we employ the variational inference and the Maximum A Posteriori (MAP) approximation techniques.

(2) Compactly supported data modeling through the utilization of statistically compatible distributions. In particular, we address the problem of proportional sequential data modeling. That is data which is positive and sums to one across its total dimensions. Hence, we use the Diricihlet, the Beta-Liouville, and the Generalized Dirichlet distributions.

(3) Whereas the underneath patterns in data may be consistent, the length might not. Modeling of such a phenomenon may be performed through constructing a generative model representation for a uniform length data formulation to be input to a discriminative model. This is the generative/discriminative HMM/Support Vector Machine (SVM) technique that we apply for the proposed models in this thesis.

(4) Infinite models are an incarnation of the solution to the setting a unique number of states

as a variable parameter for the structure of the model. We base the proposed flexible structure on the Hierachical Dirichlet Process (HDP) though effectively it is a truncation of an appropriately chosen long chain of states.

(5) Feature selection refers to the process of filtering representative features for the most efficient modeling of the data without the additional noise of redundant or uninformative ones.

(6) Modeling an oncoming stream of data may be performed with an online version of the proposed model. Maintaining the initial setting of the parameters is also incorporated in our proposed contribution that we refer to as incremental learning. It is noteworthy to mention that given the required computational time complexity, this is not a real-time system. It is however capable of handling dynamic data.

These connect to each other through the development of the various parameters that define the HMM. Moreover, its abilities to flexibly model the addressed dynamic data and features also are addressed. Finally, and more broadly, all of the aforementioned themes address these challenges for the proportional data modeling, whereas we also briefly show an improvement in the setup for the experimental employment of traditional based HMMs in the area of occupancy detection as will be discussed shortly. It is also noteworthy to mention that the identifiability problem of HMMs (in regards to the uniqueness of the states) constraints our assumption of an exact experimental setup across the classes.

In the following sections, we focus on presenting a brief description of HMMs before detailing the contributions that we add to the literature of this interesting topic. Finally, we overview the organization of the remainder of the thesis.

1.1 Hidden Markov models

In this section, we introduce the HMM and present its various aspects. We begin with an overview of the model in Section 1.1.1 and discuss its origin and assumptions. We then evolve our description to divulge the topologies of HMMs in Section 1.1.2. Next, we examine the Gaussian mixture model (GMM) and its famous Expectation Maximization (EM) algorithm in Section

1.1.3 as a building block for the upcoming analysis of HMMs. In Section 1.1.4, we disclose the mathematical formulations for the learning of its parameters. Then, in Section 1.1.5, we finalize our mathematical discussions of HMMs with the final solution (the Viterbi algorithm) to the infamous three problems that are well-posed for HMMs (introduced in Section 1.1.1). Finally, we also briefly explore applications of HMMs in Section 1.1.6. It is our aspiration that we present HMMs in an easy, accessible, and intuitive manner for future generations of researchers and further motivate the progression of this interesting area of probabilistic graphical modeling.

1.1.1 Overview

HMMs are one of the most popular statistical methods used in sequential and time series probabilistic modeling [10, 11]. A HMM is a well-received double stochastic model that uses a compact set of features to extract underlying statistics [3]. Its structure is formed primarily from a Markov chain of latent variables with each corresponding to the conditioned observation. A Markov chain is one of the least complicated ways to model sequential patterns in time series data. It was first introduced by *Andrey Markov* in the early 20$^{\text{th}}$ century. Late 1960s and early 1970s then saw a boom of papers by *Leonard E. Baum* and other researchers that introduced and addressed its statistical techniques and modeling [10]. It allows us to maintain generality while relaxing the independent identically distributed assumption [12].

Mathematically, a HMM is characterized by an underlying stochastic process with K hidden states that form a Markov chain. A graphical representation can be seen in Fig. 1.1. It is also noteworthy to mention that the aforementioned latent variable must be discrete in nature. This demonstrates the distinction between the HMMs and another state space model known as the linear dynamical system [13] whose description is out of the scope of this thesis. Each of the states is governed by an initial probability π, and the transition between the states at time t can be visualized with a transition matrix $B = \{b_{ii'} = P(s_t = i'|s_{t-1} = i)\}$. In each state s_t, an observation is emitted corresponding to its distribution which may be discrete or continuous. This is the observable stochastic process set.

The emission matrix of the discrete observations can be denoted by $\Xi = \{\Xi_{it}(m) = P(X_t =$

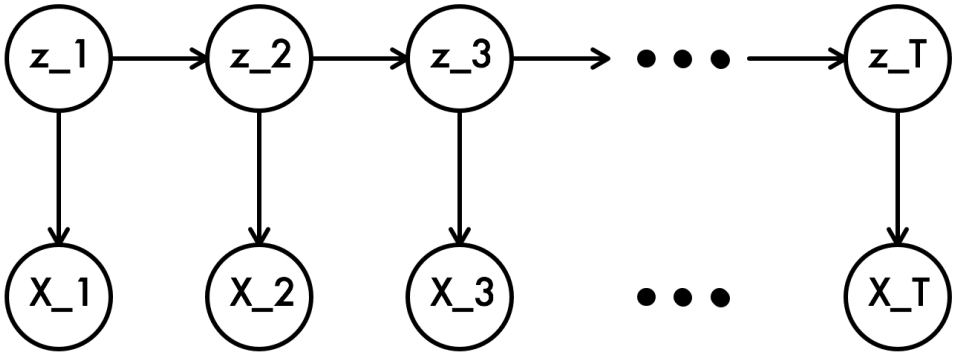


Figure 1.1: A typical hidden markov chain structure representation of a time series where z_1 denotes the first hidden state z_1 and X_1 denotes the corresponding observed state X_1. This is shown accordingly for a time series of length T.

$\xi_m \mid s_t = i)\}$ where $[m, t, i] \in [1, M] \times [1, T] \times [1, K]$, and the set of all possible discrete observations $\xi = \{\xi_1, ..., \xi_m, ..., \xi_M\}$. On the other hand, the respective parameters of a probability distribution define the observation emission for a continuous observed symbol sequence. The Gaussian distribution is most commonly used which is defined by its mean and covariance matrix $\varkappa = (\mu, \Sigma)$ [10, 14, 15]. Consequently, a mixing matrix must be defined $C = \{c_{ij} = P(m_t = j \mid s_t = i)\}$ in the case of continuous HMM emission probability distribution where $j \in [1, M]$ such that M is the number of mixture components in set $L = \{m_1, ..., m_M\}$. Hence, a discrete or continuous HMM may be defined with the following respective parameters $\Lambda = \{B, \Xi, \pi\}$ or $\{B, C, \varkappa, \pi\}$.

We next briefly recall the two conditional independence assumptions that allow for the tractability of the HMM algorithms [16]:

Assumption 1:

Given the $(t-1)$-st hidden variables, the t-th hidden variable is independent of all other previous variables such that:

$$P(s_t \mid s_{t-1}, X_{t-1}, ..., s_1, X_1) = P(s_t \mid s_{t-1}) \tag{1}$$

This is known as the *Limited Horizon* assumption such that state s_{t-1} has a sufficient representative summary of the past in order to predict the future.

Assumption 2:

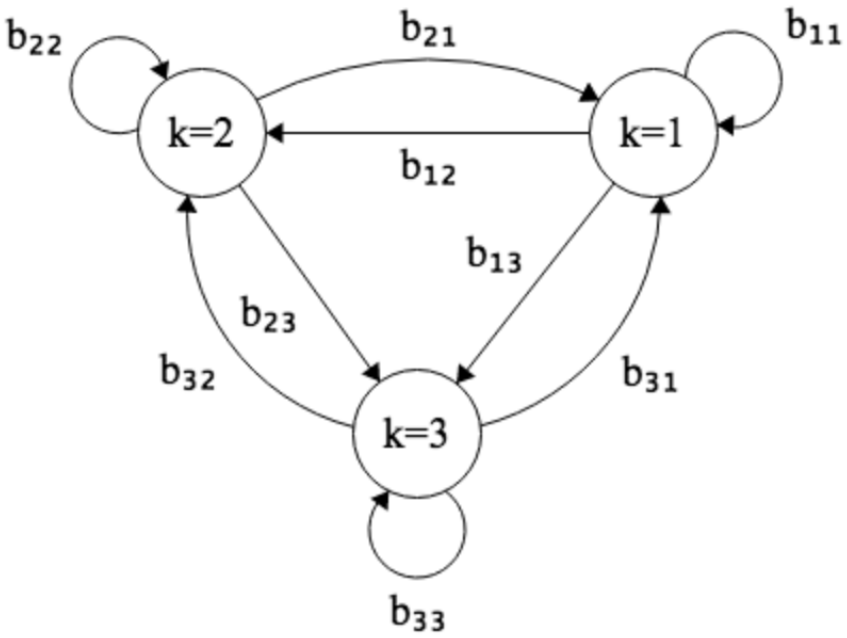


Figure 1.2: A HMM transition diagram with three states.

Given the t-th hidden variable, the t-th observation is independent of other variables such that:

$$P(X_t \mid s_T, X_T, s_{T-1}, X_{T-1}, ..., s_{t+1}, X_{t+1}, s_t, s_{t-1}, X_{t-1}, ..., s_1, X_1) = P(X_t \mid s_t) \quad (2)$$

This is known as the *Stationary Process* assumption such that the conditional distribution of a state does not change over time and is independent of other variables.

Now, we present the three classical problems of HMMs first introduced by Rabiner in [10]: evaluation or likelihood, estimation or decoding, and training or learning. These are described as follows:

(1) **Evaluation problem:** is mainly concerned with computing the probability that a particular sequential or time series data was generated by the HMM model, given both the observation sequence and the model. Mathematically, the primary objective is computing the probability $P(X \mid \Lambda)$ of the observation sequence $X = X_1, X_2, ..., X_T$ with length T given a HMM model Λ.

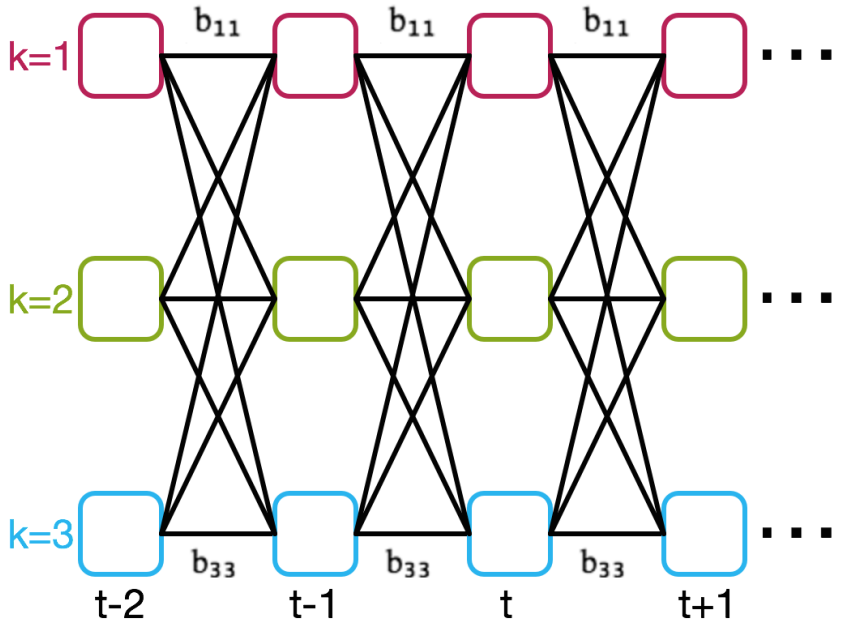


Figure 1.3: Lattice or trellis HMM structure which is a representation of the hidden states.

(2) **Decoding problem:** finds the optimum state sequence path $I = i_1, i_2, ..., i_T$ for an observation sequence X. This is mathematically $\vec{s}^* = \text{argmax}_{\vec{s}}\, P(\vec{s} \mid X, \Lambda)$.

(3) **Learning problem:** refers to building a HMM model through finding or "learning" the right parameters to describe a particular set of observations. Formally, this is performed with maximizing the probability $P(X \mid \Lambda)$ of the set of observation sequence X given the set of parameters determined Λ. Mathematically, this is $\Lambda^* = \text{argmax}_{\Lambda}\, P(X \mid \Lambda)$.

For the thorough explanation of the HMM algorithms to follow, we also introduce another visualization that depicts the graphical directed HMM structure as shown in Fig. 1.2. Fig. 1.3 shows transitions then when they become trellis or lattice.

1.1.2 Topologies

Though the main principal of this chapter is to impart an introduction to HMMs in the simplest manner, we would be remiss not to bring the attention of the reader to the main variants of HMMs. These pertain to its structure as well as its functionality. Specifically, we may have a:

- **Hidden Markov Model (HMM):** introduced in Section 1.1.1, and the entire chapter is dedicated to discussing its details. This is the traditional model and is the one referred to if no other distinctions are made to the name or referral to its structure.

- **Hidden Semi Markov Model (HSMM):** explicitly deals with state duration as its hidden stochastic process is based on a semi-Markov chain, so that a hidden state is persistent for time duration t_d. This allows for an explicit definition of the duration as an independent variable, whereas the duration of HMMs is implicitly assumed to follow a geometric distribution [17].

- **Factorial Hidden Markov Model (FHMM):** is a multilayer (each of which is a HMM that works independently from other layers) state structure for modeling of multiple loosely coupled random processes.

- **Layered HMM (LHMM):** is made up of several composed HMMs at each layer that run parallel to each other, providing an output to the higher layer. Hence, each layer is connected to the next by inferential results.

- **Autoregressive HMM (ARHMM):** can explicitly model the longer-range correlations of sequential data by adding direct stochastic dependence among observations.

- **Non-Stationary HMM (NSHMM):** capture state duration behaviour by defining a set of dynamic transition probability parameters. It can model state duration probabilities explicitly as a function of time.

- **Hierarchical HMM (HHMM):** has multi-levels states that describe a sequence of input at various levels of details. In a way, this is likened to a HMM with internal states generated from a sub-HMM in a tree-like structure.

Not only does a traditional HMM fall into the first category of the earlier discussed variants, but also is of a first-order nature. First-order HMMs refer to the property that characterizes the model in terms of the current state's dependency on previous ones. When the Markovian conditional independence is held then the model may be referred to as first-order. Indeed, this is omitted in many cases as this is one of the main assumptions of HMMs. Nonetheless, other extensions exist where connections between extra past states are made and the order would then be imperative in the description of the model. Hence, an n^{th}-order HMM is simply one with a Markov chain structure in which each state depends on the prior n states.

There are various topologies of a traditional first-order HMM which would correspond to its transition matrix construction. That is the connection between the states (i.e., edges in the graph representation) can be omitted by setting the corresponding element in B to zero. The following are well-known special cases:

(1) **Ergodic HMM:** where the transition probability between any two states is nonzero. This is also known as a *fully-connected HMM*. This is the most flexible structure and is ubiquitous as it represents the traditional full fledged HMM. This allows the model to update its transition matrix with regards to the data for a data-based approach. We note a depiction of this in both Fig. 1.2 and Fig. 1.3 where any of the states can be visited from any other state.

(2) **Left-to-Right HMM:** requires that transitions can only be made from the current state to

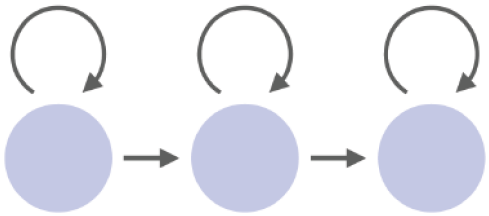


Figure 1.4: A left-to-right HMM topology with three states.

its equivalent or a larger index resulting in an upper triangular state transition matrix. This is done by simply initiating the lower triangle of the state transition matrix to zeros so that any consequent updates leave it as such. In effect, we have imposed a temporal order to the HMMs. These are typically used in speech and word recognition applications. A graphical depiction is shown in Fig. 1.4.

The structure of the HMM may also vary in regards to its emission distribution. Even in the case of assuming a continuous distribution, we may have a single distribution in each state or a mixture.

1.1.3 Gaussian Mixture Models and the Expectation Maximization Algorithm

The maximum likelihood is a general problem in the computational pattern recognition and machine learning community. It pertains to estimating the parameters of density functions given a set of data. The latter is assumed to be static for simplicity. Concluding remarks in Section 1.1.6 address non-static (dynamic) data.

Assuming independent and identically distributed (i.i.d.) data $\mathcal{X}$, a density function of its distribution p or the likelihood of the parameters given the data $\mathcal{L}(\Theta \mid \mathcal{X})$; i.e., the incomplete data-likelihood function may be denoted with the following:

$$p(\mathcal{X} \mid \Theta) = \prod_{i=1}^{N} p(\mathbf{x}_i \mid \Theta) = \mathcal{L}(\Theta \mid \mathcal{X}) \tag{3}$$

The goal then as is evident from the name of the problem is to maximize this function. Mostly this maximization is performed with the log of the function for ease of analytical purposes. This

in turn results in finding the optimum set of parameters, Θ^*, that best fits the distribution to $\mathcal{X}$. Mathematically, that is:

$$\Theta^* = \text{argmax}_{\Theta}\, \mathcal{L}(\Theta \mid \mathcal{X}) \tag{4}$$

Consequently, the derivative of the function is found and solved when set to zero. Indeed, it is noteworthy to mention that when $p(\mathbf{x} \mid \Theta)$ is a Gaussian distribution where $\Theta = (\mu, \sigma^2)$, the solution forms the equations that are commonly used for the mean and variance of a data set. However, in many cases, solving the derivative of the likelihood function is not analytically possible and hence the employment of the Expectation Maximization (EM) algorithm becomes necessary.

A question might then be raised here as to why we need mixtures. The answer lies in its better ability to capture the underlying pattern of the data. For instance, assume that the mean data point lies in between two subgroups (clusters) of the data. Using a single component for its modeling will render sub-optimal results compared to a mixture where the optimum solution would be to use two components.

The EM algorithm [18, 19, 20, 21, 22] is a general methodology for finding the maximum likelihood estimate of the parameters. Effectively, these learned parameters best model the underlying pattern of the data (or a particular dataset) when the latter is incomplete. Indeed, assumption of such hidden parameters and their values simplifies the process as we will discuss shortly.

We next introduce the general probabilistic formulation of mixture models of M components:

$$p(\mathbf{x} \mid \Theta) = \sum_{i=1}^{M} \zeta_i p_i(\mathbf{x} \mid \theta_i) \tag{5}$$

where $\Theta = (\zeta_i, \ldots, \zeta_M, \theta_i, \ldots, \theta_M)$ such that $\sum_{i=1}^{M} \zeta_i = 1$ which represents the weights of each of the distributions' density function $p_i(\mathbf{x} \mid \theta_i)$ with its respective set of characterizing parameters θ_i. Note that $p_i(\mathbf{x} \mid \theta_i)$ will be considered to be a Gaussian distribution for the remainder of this section, such that $\Theta = \Theta^g$.

Then,

$$\begin{aligned}\log(\mathcal{L}(\Theta^g \mid \mathcal{X})) &= \log \prod_{i=1}^{N} p(x_i \mid \Theta^g) \\ &= \sum_{i=1}^{N} \log \left(\sum_{j=1}^{M} \zeta_j p_j(x_i \mid \theta_j) \right)\end{aligned} \tag{6}$$

This is difficult to solve as it contains the log of the sum. This may be simplified with the assumption that this is incomplete data with mixture component labels $\mathcal{Y} = \{y_i\}_{i=1}^{N}$. That is, $y_i \in 1, \ldots, M$ for each data point i with $y_i = k$ to signify the mixture component k that the sample was generated by. It is noteworthy to mention that another, and arguably better, scheme to also achieve this is to denote this as a latent indicator variable that becomes 1 at the position of the mixture component for a sample, and 0 otherwise. Nevertheless, the likelihood now may be denoted by:

$$\begin{aligned}\log(\mathcal{L}(\Theta^g \mid \mathcal{X}, \mathcal{Y})) &= \log(p(\mathcal{X}, \mathcal{Y} \mid \Theta^g)) \\ &= \sum_{i=1}^{N} \log(p(x_i \mid y_i) p(y_i)) \\ &= \sum_{i=1}^{N} \log(\zeta_{y_i} p_{y_i}(x_i \mid \theta_{y_i}))\end{aligned} \tag{7}$$

$\mathcal{Y}$ is assumed to be a random vector with the Gaussian distribution (or any desired distribution) to be computationally feasible. Then, applying Bayes's rule:

$$\begin{aligned}p_{y_i}(x_i, \Theta^g) &= \frac{\zeta_{y_i}^g p_{y_i}(x_i \mid \theta_{y_i}^g)}{p_{y_i}(x_i \mid \Theta^g)} \\ &= \frac{\zeta_{y_i}^g p_{y_i}(x_i \mid \theta_{y_i}^g)}{\sum_{k=1}^{M} \zeta_k^g p_k(x_i \mid \theta_k)}\end{aligned} \tag{8}$$

and $\mathbf{y} = (y_1, \ldots, y_N)$ for an independent data sample in:

$$p(\mathbf{y} \mid \mathcal{X}, \Theta^g) = \prod_{i=1}^{N} p(y_i \mid x_i, \theta) \tag{9}$$

Consequently,

we may now compute the first step in the EM algorithm which depends on computing the expected value of the complete-data log-likelihood $p(\mathcal{X}, \mathcal{Y} \mid \Theta)$ with respect to $\mathcal{Y}$ given $\mathcal{X}$ and the current parameter estimates $\Theta^{(t-1)}$. This is also referred to as the E-step. Generally, this is denoted as:

$$Q\left(\Theta, \Theta^{(t-1)}\right) = \mathbb{E}\left[\log p(\mathcal{X}, \mathcal{Y} \mid \Theta) \mid \mathcal{X}, \Theta^{(t-1)}\right] \tag{10}$$

Then,

$$\begin{aligned}
Q\left(\Theta, \Theta^{g}\right) &= \sum_{\mathbf{y} \in \Upsilon} \log(\mathcal{L}(\Theta \mid \mathcal{X}, \mathbf{y})) p\left(\mathbf{y} \mid \mathcal{X}, \Theta^{g}\right) \\
&= \sum_{\mathbf{y} \in \Upsilon} \sum_{i=1}^{N} \log\left(\zeta_{y_i} p_{y_i}\left(x_i \mid \theta_{y_i}\right)\right) \prod_{j=1}^{N} p\left(y_j \mid x_j, \Theta^{g}\right) \\
&= \sum_{y_1=1}^{M} \sum_{y_i=1}^{M} \cdots \sum_{y_N=1}^{M} \sum_{i=1}^{N} \log\left(\zeta_{y_i} p_{y_i}\left(x_i \mid \theta_{y_i}\right)\right) \prod_{j=1}^{N} p\left(y_j \mid x_j, \Theta^{g}\right) \\
&= \sum_{y_i=1}^{M} \sum_{\ell_i}^{M} \cdots \sum_{\ell_\ell}^{M} \sum_{i=1}^{M} \sum_{\ell=1}^{M} \delta_{\ell, y_i} \log\left(\zeta_{\ell} p_{\ell}\left(x_i \mid \theta_{\ell}\right)\right) \prod_{j=1}^{N} p\left(y_j \mid x_j, \Theta^{g}\right) \\
&= \sum_{\ell=1}^{M} \sum_{i=1}^{N} \log\left(\zeta_{\ell} p_{\ell}\left(x_i \mid \theta_{\ell}\right)\right) \sum_{y_1=1}^{M} \sum_{y_2=1}^{M} \cdots \sum_{y_N=1}^{M} \delta_{\ell, y_i} \prod_{j=1}^{N} p\left(y_j \mid x_j, \Theta^{g}\right)
\end{aligned} \tag{11}$$

This may be simplified further. First, for $\ell \in 1, \ldots, M$:

$$\begin{aligned}
&\textstyle\sum_{y_1=1}^{M} \sum_{y_2=1}^{M} \cdots \sum_{y_N=1}^{M} \delta_{\ell, y_i} \prod_{j=1}^{N} p\left(y_j \mid x_j, \Theta^{g}\right) \\
&= \left(\textstyle\sum_{y_1=1}^{M} \cdots \sum_{y_{i-1}=1}^{M} \sum_{y_{i+1}=1}^{M} \cdots \sum_{y_N=1}^{M} \prod_{j=1, j \neq i}^{N} p\left(y_j \mid x_j, \Theta^{g}\right)\right) p\left(\ell \mid x_i, \Theta^{g}\right) \\
&= \textstyle\prod_{j=1, j \neq i}^{N} \left(\sum_{y_j=1}^{M} p\left(y_j \mid x_j, \Theta^{g}\right)\right) p\left(\ell \mid x_i, \Theta^{g}\right) = p\left(\ell \mid x_i, \Theta^{g}\right)
\end{aligned} \tag{12}$$

as $\sum_{i=1}^{M} p\left(i \mid x_j, \Theta^{g}\right) = 1$. Then, replacing Eq. (12) into Eq. (11), we get:

$$\begin{aligned}
Q\left(\Theta, \Theta^{g}\right) &= \sum_{\ell=1}^{M} \sum_{i=1}^{N} \log\left(\zeta_{\ell} p_{\ell}\left(x_i \mid \theta_{\ell}\right)\right) p\left(\ell \mid x_i, \Theta^{g}\right) \\
&= \sum_{\ell=1}^{M} \sum_{i=1}^{N} \log\left(\zeta_{\ell}\right) p\left(\ell \mid x_i, \Theta^{g}\right) + \sum_{\ell=1}^{M} \sum_{i=1}^{N} \log\left(p_{\ell}\left(x_i \mid \theta_{\ell}\right)\right) p\left(\ell \mid x_i, \Theta^{g}\right)
\end{aligned} \tag{13}$$

This allows us to move into the next major stage that is part of the EM algorithm, which is the maximization step (M-step).

In the M-step, the goal is to maximize the expectation computed through:

$$\Theta^{(t)} = \text{argmax}_{\Theta}\, Q(\Theta, \Theta^{(t-1)}) \tag{14}$$

This is repeated together with the E-step with a guarantee to converge to a local maximum as the log-likelihood is increased.

ζ_ℓ and θ_ℓ may be maximized independently due to the non-existence of a relationship between them. We begin with the ζ_ℓ and use the Lagrange multiplier λ with the constraint $\sum_\ell \zeta_\ell = 1$. This is due to the role that ζ_ℓ undertakes as the weight of each of the mixture components. Then, we need to solve the following:

$$\frac{\partial}{\partial \zeta_\ell}\left[\sum_{\ell=1}^{M}\sum_{i=1}^{N}\log(\zeta_\ell)p(\ell \mid x_i, \Theta^g) + \lambda\left(\sum_{\ell}\zeta_\ell - 1\right)\right] = 0 \tag{15}$$

or

$$\sum_{i=1}^{N}\frac{1}{\zeta_\ell}p(\ell \mid x_i, \Theta^g) + \lambda = 0 \tag{16}$$

When both sides are summed, we end up with $\ell\ \lambda = -N$, so that:

$$\zeta_\ell = \frac{1}{N}\sum_{i=1}^{N}p(\ell \mid x_i, \Theta^g) \tag{17}$$

This is a general result that holds for all mixture models, regardless of the distribution at hand. As to the θ_ℓ, that is entirely dependent on the distribution assumed. For us, that is $\theta = (\mu, \Sigma)$ denoting the mean and the covariance matrix of a D-dimensional Gaussian distribution (or component in this instance) respectively. This is formulated by:

$$p_\ell(x \mid \mu_\ell, \Sigma_\ell) = \frac{1}{(2\pi)^{D/2}|\Sigma_\ell|^{1/2}}\text{exp}^{-\frac{1}{2}(x-\mu_\ell)^T|\Sigma_\ell|^{-1}(x-\mu_\ell)} \tag{18}$$

Next, we compute the log of Eq. (18) and ignore any constants as they are zeroed out when we will

compute the derivatives. Then, substitute into Eq. (13):

$$\begin{aligned}&\sum_{\ell=1}^{M}\sum_{i=1}^{N}\log(p_\ell(x_i \mid \mu_\ell, \Sigma_\ell))p(\ell \mid x_i, \Theta^g)\\&\quad=\sum_{\ell=1}^{M}\sum_{i=1}^{N}\left(-\frac{1}{2}\log(|\Sigma|)-\frac{1}{2}(x-\mu_\ell)^T|\Sigma_\ell|^{-1}(x-\mu_\ell)\right)p(\ell \mid x_i, \Theta^g)\end{aligned} \tag{19}$$

We now derive Eq. (19) with respect to μ and solve for zero:

$$\sum_{i=1}^{N}|\Sigma_\ell|^{-1}(x-\mu_\ell)p(\ell \mid x_i, \Theta^g)=0 \tag{20}$$

The result is:

$$\mu_\ell=\frac{\sum_{i=1}^{N}x_i p(\ell \mid x_i, \Theta^g)}{\sum_{i=1}^{N}p(\ell \mid x_i, \Theta^g)} \tag{21}$$

For Σ, first we rewrite Eq. (19) as:

$$\begin{aligned}&\textstyle\sum_{\ell=1}^{M}\left[\frac{1}{2}\log\left(\left|\Sigma_\ell^{-1}\right|\right)\sum_{i=1}^{N}p\left(\ell \mid x_i, \Theta^g\right)-\frac{1}{2}\sum_{i=1}^{N}p\left(\ell \mid x_i, \Theta^g\right)\mathrm{tr}\left(\Sigma_\ell^{-1}\left(x_i-\mu_\ell\right)\right.\right.\\&\quad\left.\left.\left(x_i-\mu_\ell\right)^T\right)\right]\\&=\textstyle\sum_{\ell=1}^{M}\left[\frac{1}{2}\log\left(\left|\Sigma_\ell^{-1}\right|\right)\sum_{i=1}^{N}p\left(\ell \mid x_i, \Theta^g\right)-\frac{1}{2}\sum_{i=1}^{N}p\left(\ell \mid x_i, \Theta^g\right)\mathrm{tr}\left(\Sigma_\ell^{-1}\mathfrak{N}_{\ell,i}\right)\right]\end{aligned} \tag{22}$$

where $\mathfrak{N}=\left(x_i-\mu_\ell\right)\left(x_i-\mu_\ell\right)^T$.

Now, we can compute the derivative with respect to Σ_ℓ:

$$\begin{aligned}&\frac{1}{2}\sum_{i=1}^{N}p(\ell \mid x_i, \Theta^g)(2\Sigma_\ell-\mathrm{diag}(\Sigma_\ell))-\frac{1}{2}\sum_{i=1}^{N}p(\ell \mid x_i, \Theta^g)(2\mathfrak{N}_{\ell,i}-\mathrm{diag}(\mathfrak{N}_{\ell,i}))\\&\quad=\frac{1}{2}\sum_{i=1}^{N}p(\ell \mid x_i, \Theta^g)(2\mathfrak{J}_{\ell,i}-\mathrm{diag}(\mathfrak{J}_{\ell,i}))\\&\quad=2\mathfrak{R}-\mathrm{diag}(\mathfrak{R})\end{aligned} \tag{23}$$

where $\mathfrak{J}_{\ell,i}=\Sigma_\ell-\mathfrak{N}_{\ell,i}$ and $\mathfrak{R}=\frac{1}{2}\sum_{i=1}^{N}p(\ell \mid x_i, \Theta^g)\mathfrak{J}_{\ell,i}$. Setting derivative to zero through $2\mathfrak{R}-\mathrm{diag}(\mathfrak{R})=0$ or $\mathfrak{R}=0$, then:

$$\sum_{i=1}^{N}p(\ell \mid x_i, \Theta^g)(\Sigma_\ell-\mathfrak{N}_{\ell,i})=0 \tag{24}$$

or

$$\begin{aligned}\Sigma_\ell &= \frac{\sum_{i=1}^{N} p(\ell \mid x_i, \Theta^g)\mathfrak{N}_{\ell,i}}{\sum_{i=1}^{N} p(\ell \mid x_i, \Theta^g)} \\ &= \frac{\sum_{i=1}^{N} p(\ell \mid x_i, \Theta^g)\,(x_i - \mu_\ell)\,(x_i - \mu_\ell)^T}{\sum_{i=1}^{N} p(\ell \mid x_i, \Theta^g)}\end{aligned} \tag{25}$$

Consequently, these are the final update equations for the parameters of GMM with the EM algorithm:

$$\zeta_\ell^{new} = \frac{1}{N}\sum_{i=1}^{N} p(\ell \mid x_i, \Theta^g) \tag{26}$$

$$\mu_\ell^{new} = \frac{\sum_{i=1}^{N} x_i p(\ell \mid x_i, \Theta^g)}{\sum_{i=1}^{N} p(\ell \mid x_i, \Theta^g)} \tag{27}$$

$$\Sigma_\ell^{new} = \frac{\sum_{i=1}^{N} p(\ell \mid x_i, \Theta^g)\,(x_i - \mu_\ell^{new})\,(x_i - \mu_\ell^{new})^T}{\sum_{i=1}^{N} p(\ell \mid x_i, \Theta^g)} \tag{28}$$

1.1.4 Baum Welch Algorithm

The *Baum Welch algorithm* is a special case of the EM algorithm whereby we can efficiently calculate the parameters of the HMM [23, 24]. In the context of HMMs, this algorithm is of extreme importance [10]. The Baum Welch algorithm is traditionally used to solve the estimation problem of HMMs. As a matter of fact, this remains an active area of research with interesting recent results such as in [25].

This may be applied to the discrete as well as the continuous case. In this chapter, we focus on the latter and further develop Section 1.1.3 for the computation of such continuous emission distributions. The discrete case is a simplification of the continuous due to its limited parameters and hence can be induced in a straightforward manner from our discussions.

The Baum Welch algorithm is also known as the *forward-backward algorithm*. This is due to its composition of two approaches that when repeated recursively form the complete algorithm. As you might have concluded, these algorithms are named the *forward algorithm* and the *backward algorithm*. This iterative algorithm requires an initial random clustering of the data, is guaranteed to converge to more compact clusters at every step, and stops when the log-likelihood ratios no longer show significant changes [26].

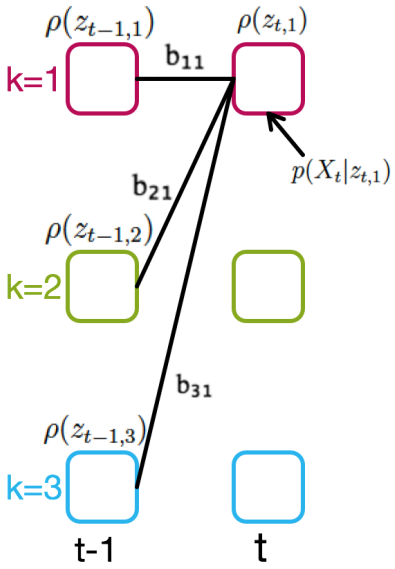


Figure 1.5: Graphical representation of the evaluation of the ρ variable of the forward algorithm in a HMM lattice fragment.

The forward algorithm solves the first problem that is posed for HMM as discussed in Section 1.1.1; i.e., the evaluation problem. The forward algorithm calculates the probability of being in state state s_i at time t after the corresponding partial observation sequence given the HMM model Λ. This defines the forward variable $\rho_t(i) = P(X_1, X_2, ..., X_t, i_t = s_i \mid \Lambda)$ which is solved recursively as follows:

(1) Initiate the forward probabilities with the joint probability of state s_i and the initial observation X_1:

$$\rho_1(i) = \pi_i \Xi_i(X_1), \quad 1 \leqslant i \leqslant K \tag{29}$$

(2) Calculate how state $q_{i'}$ is reached at time $t+1$ from the K possible states s_i, $i = 1, 2, ..., K$ at time t and sum the product over all the K possible states:

$$\rho_{t+1}(j) = \left[\sum_{i=1}^{K} \rho_t(i) b_{ij}\right] \Xi_j(X_{t+1}), \quad t = 1, 2, ..., T-1; 1 \leqslant j \leqslant K \tag{30}$$

(3) Finally, compute:

$$P(X \mid \Lambda) = \sum_{i=1}^{K} \rho_T(i) \tag{31}$$

The forward algorithm has a computational complexity of K^2T which is considerably less than a naive direct calculation approach. A graphical depiction of the forward algorithm can be observed in Fig. 1.5.

Fig. 1.6 depicts the computation process of the backward algorithm in a HMM lattice structure. It is similar to the forward algorithm, but now computing the tail probability of the partial observation from $t+1$ to the end, given that we are starting at state s_i at time t and model Λ. This has the variable $\beta_t(i) = P(X_{t+1}, X_{t+2}, ..., X_T, i_t = s_i \mid \Lambda)$ and is solved as follows:

(1) Compute an arbitrary initialization:

$$\beta_T(i) = 1, \quad 1 \leqslant i \leqslant K \tag{32}$$

(2) Compute the remainder of the variable with the update:

$$\beta_t(i) = \sum_{i'=1}^{K} b_{ii'} \Xi_{i'}(X_{t+1}) \beta_{t+1}(i'), \quad t = T-1, T-2, ..., 1; 1 \leqslant i \leqslant K \tag{33}$$

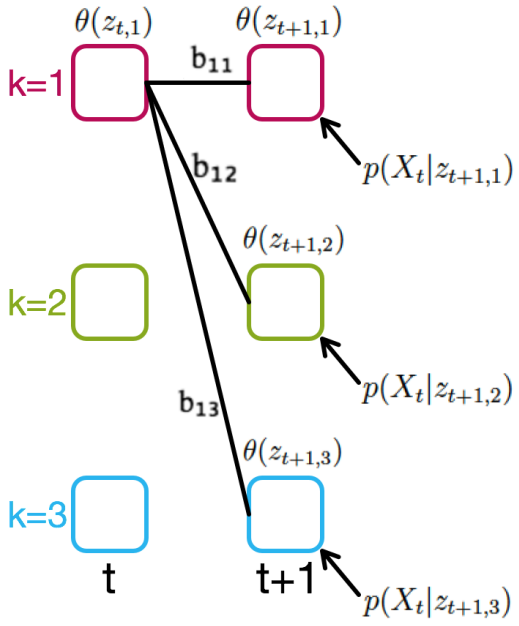


Figure 1.6: Graphical representation of the evaluation of the β variable of the backward algorithm in a HMM lattice fragment.

In order to apply the Baum Welch algorithm, we must also define:

$$\begin{aligned}\gamma_t(i) &= P(i_t = s_i \mid X, \Lambda) \\ &= \frac{P(X, i_t = s_i \mid \Lambda)}{P(X \mid \Lambda)} \\ &= \frac{P(X, i_t = s_i \mid \Lambda)}{\sum_{i=1}^{K} P(X, i_t = s_i \mid \Lambda)}\end{aligned} \tag{34}$$

where $\gamma_t(i)$ is the probability of being is state s_i at time t, given Λ and X. Also, because of the Markovian conditional assumption, we can denote the following:

$$\begin{aligned}\rho_t(i)\beta_t(i) &= P(X_1, X_2, ..., X_t, i_t = s_i \mid \Lambda) P(X_{t+1}, X_{t+2}, ..., X_T, i_t = s_i \mid \Lambda) \\ &= P(X, i_t = s_i \mid \Lambda)\end{aligned} \tag{35}$$

Then, we may also formulate the following:

$$\gamma_t(i) = \frac{\rho_t(i)\beta_t(i)}{\sum_{i'=1}^{K} \rho_t(i')\beta_t(i')} \tag{36}$$

Further, another important variable needs to be defined. That is the probability of path being in state s_i at time t and then transitioning at time $t+1$ with $b_{ii'}$ to state s'_i, given Λ and X. We denote this by $\varphi_t(i, i')$ and formulate it as:

$$\begin{aligned}\varphi_t(i, i') &= P(i_t = s_i, i_{t+1} = s'_i \mid X, \Lambda) \\ &= \frac{P(i_t = s_i, i_{t+1} = s'_i, X \mid \Lambda)}{p(X \mid \Lambda)} \\ &= \frac{\rho_t(i) b_{ii'} \Xi_{i'}(X_{t+1}) \beta_{t+1}(i')}{\sum_{i=1}^{K} \sum_{i'=1}^{K} \rho_t(i) b_{ii'} \Xi_{i'}(X_{t+1}) \beta_{t+1}(i')} \\ &= \frac{\gamma_t(i) b_{ii'} \Xi_{i'}(X_{t+1}) \beta_{t+1}(i')}{\beta_t(i)}\end{aligned} \tag{37}$$

$\rho_t(i)$ then considers the first observations ending at state s_i at time t, $\beta_{t+1}(i')$ the rest of the observation sequence, and $b_{ii'}\Xi_{i'}(X_{t+1})$ the transition to state $s_{i'}$ with observation X_{t+1} at time

$t+1$. Hence, $\gamma_t(i)$ may also be expressed as:

$$\gamma_t(i) = \sum_{i'=1}^{K} \varphi_t(i, i') \tag{38}$$

whereby $\sum_{t=1}^{T-1} \varphi_t(i, i')$ is the expected number of transitions made from s_i to $s_{i'}$ and $\sum_{t=1}^{T} \gamma_t(i)$ is the expected number of transitions made from s_i.

The general re-estimation formulas for the HMM parameters π and B are then:

$$\bar{\pi}_i = \gamma_1(i), 1 \leqslant i \leqslant K \tag{39}$$

which is the relative frequency spent in state s_i at time $T = 1$ and

$$\bar{b}_{ii'} = \frac{\sum_{t=1}^{T-1} \varphi_t(i, i')}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{40}$$

which is the expected number of transitions from state s_i to $s_{i'}$ relative to the expected total number of transitions away from state i.

For Ξ, it is hereby defined as a GMM. Then, the definition of another probability for the generation of X_t from the the ℓ^{th} component of the i^{th} GMM is required and may be expressed as:

$$\begin{aligned} \gamma_t(i\ell) &= P(i_t = s_i, Y_{it} = \ell \mid X, \Lambda) \\ &= \gamma_t(i) \frac{c_{i\ell} \Xi_{i\ell}(X_t)}{\Xi_i(X_t)} \end{aligned} \tag{41}$$

where Y_{it} is an indicator random variable for the mixture component at t for s_i. Our earlier treatment of GMMs in Section 1.1.3 enables us to easily derive the update equations needed. These are:

$$c_{i\ell} = \frac{\sum_{t=1}^{T} \gamma_t(i\ell)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{42}$$

$$\mu_{i\ell} = \frac{\sum_{t=1}^{T} \gamma_t(i\ell) X_t}{\sum_{t=1}^{T} \gamma_t(i\ell)} \tag{43}$$

$$\Sigma_{i\ell} = \frac{\sum_{t=1}^{T} \gamma_t(i\ell)(X_t - \mu_{i\ell})(X_t - \mu_{i\ell})^T}{\sum_{t=1}^{T} \gamma_t(i\ell)} \tag{44}$$

In case we have O sequences with each o^{th} sequence of length T_o, then the update equations are the summation across all sequences. This may be denoted by the following:

$$\pi_i = \frac{\sum_{o=1}^{O} \gamma_1^o(i)}{O} \tag{45}$$

$$b_{ii'} = \frac{\sum_{o=1}^{O} \sum_{t=1}^{T_o} \varphi_t^o(i,i')}{\sum_{o=1}^{O} \sum_{t=1}^{T_o} \gamma_t^o(i)} \tag{46}$$

$$c_{i\ell} = \frac{\sum_{o=1}^{O} \sum_{t=1}^{T_o} \gamma_t^o(i\ell)}{\sum_{o=1}^{O} \sum_{t=1}^{T_o} \gamma_t^o(i)} \tag{47}$$

$$\mu_{i\ell} = \frac{\sum_{o=1}^{O} \sum_{t=1}^{T_o} \gamma_t^o(i\ell) X_t^o}{\sum_{o=1}^{O} \sum_{t=1}^{T_o} \gamma_t^o(i\ell)} \tag{48}$$

$$\Sigma_{i\ell} = \frac{\sum_{o=1}^{O} \sum_{t=1}^{T_o} \gamma_t^o(i\ell)(X_t^o - \mu_{i\ell})(X_t^o - \mu_{i\ell})^T}{\sum_{o=1}^{O} \sum_{t=1}^{T_o} \gamma_t^o(i\ell)} \tag{49}$$

1.1.5 Viterbi Algorithm

Next, the *Viterbi algorithm* aims to find the most likely progression of states that generated a given observation sequence in a certain HMM. Hence, it offers the solution to the decoding problem. This involves choosing the most likely states at each time t individually. Hence, the expected number of correct separate states is maximized. This is illustrated in Fig. 1.7.

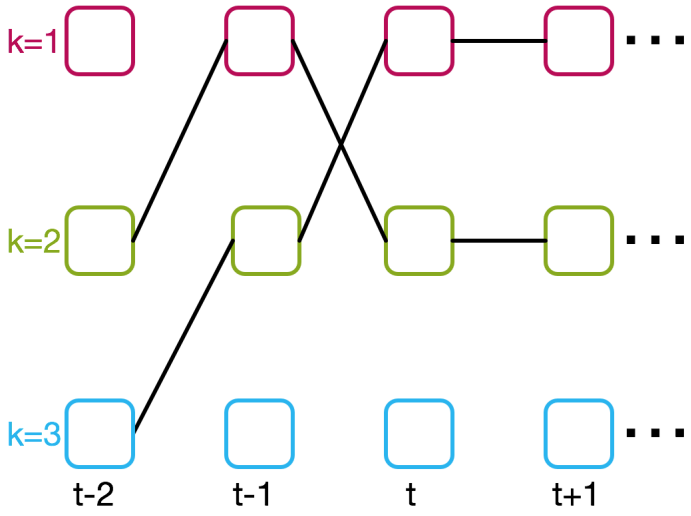


Figure 1.7: Graphical representation of two probable pathways in a HMM lattice fragment. The objective of the Viterbi algorithm is to find the most likely one.

The main steps of the Viterbi algorithm can then be summarized as:

(1) Initialization

$$\delta_1(i) = \pi_i \Xi_i(X_1), 1 \leqslant i \leqslant K \tag{50}$$

$$\psi_1(i) = 0 \tag{51}$$

(2) Recursion

$$\text{For } 2 \leqslant t \leqslant T, 1 \leqslant i' \leqslant K$$

$$\delta_t(i') = \max_{1 \leqslant i \leqslant K} \left[\delta_{t-1}(i) b_{ii'}\right] \Xi_{i'}(X_t) \tag{52}$$

$$\psi_t(i') = \operatorname{argmax}_{1 \leqslant i \leqslant K} \left[\delta_{t-1}(i) b_{ii'}\right] \tag{53}$$

(3) Termination

$$P^* = \max_{1 \leqslant i \leqslant K} \left[\delta_T(i)\right] i_T^* = \operatorname{argmax}_{1 \leqslant i \leqslant K} \left[\delta_T(i)\right] \tag{54}$$

(4) State sequence path backtracking

$$i_t^* = \psi_{t+1}(i_{t+1}^*), \text{for } t = T-1, T-2, ..., 1 \tag{55}$$

This finalizes our mathematical discussions of the background of the famous HMMs.

1.1.6 Applications

Early applications of this powerful model were in speech-related tasks and this has remained predominantly true. Indeed, it is an integral model in the musicology field. However, to motivate the reader to further explore the horizons in applying the acquired knowledge, we briefly touch upon a diversity of applications where HMMs are used in this section.

Bioinformatics is a field where HMMs are ubiquitous. For instance, it is increasingly used in genomics, gene sequencing, and protein classification. An interested reader is referred to [27] for a study of HMMs in a variety of biological applications. Forecasting weather may also be performed

utilizing HMMs such as in [28].

Security applications are another field where the application of HMMs is imperative. For instance, they may be deployed in video surveillance systems for automatic detection of security threats as well as anomaly detection [29, 30] or even to detect fraud in bank transactions [31]. HMMs are also applicable in gesture recognition. An example is artificially intelligent cockpit control in [32]. You may then infer that HMMs would also shine whenever spatio-temporal analysis is carried out due to the nature of its composition.

HMMs are also highly influential in the area of occupancy estimation. The latter is also dependent on Internet of Things (IoT) technologies. A closely related area is activity recognition in which HMMs may be used to classify such activities within a smart building environment [33]. A method for efficient power usage is also proposed in [34] and another for power signature anomaly detection in [35].

Similar to speech recognition, HMMs are highly preferred in natural language processing and its subfields. Examples include recognition of handwritten characters [36], writer identification and verification systems [37, 38], and speech synthesis for the English language [39] and recently for Tamil [40]. We also refer an interested reader to [41] for a systematic survey of the applications of HMMs.

1.2 Contributions of the Thesis

We contribute to a recent research direction that has focused on proposing new HMMs for a data-driven approach. In particular, emission distributions of the model are traditionally chosen to be a GMM. However, this is an assumption that does not hold for all cases. That is when the nature of the data can be inferred to be nonsymmetric and its range does not expand across $(-\infty, \infty)$. Indeed, other distributions have proven to be performing better in terms of fitted models in these instances [42, 43, 44, 45].

It naturally follows that that would be the circumstance in time-based probabilistic modeling using HMMs. In this thesis, we focus on proposing and investigating proportional-based HMMs;

in particular, Dirichlet, Generalized Dirichlet, and Beta Liouville-based HMMs. Another important aspect of this interesting work is the investigation of other learning techniques that improve on the traditional Baum Welch algorithm. This is because the latter suffers from a risk to over-fit or under-fit as well as vulnerability to initialization conditions. Namely, we derive variational inference and Maximum A Posteriori frameworks for proportional-based HMMs. One of the drawbacks of discriminative models is its inability to handle data of different sizes. We present a hybrid generative/discriminative approach based on the proposed models for a best-of-both-worlds framework. Another research venue that we also tackle in this thesis is online deployment.

The overall contributions of the thesis are then as follows:

- This chapter that is considered as a roadmap to HMMs for beginners and practitioners alike.
    - This has been accepted as a part of the book chapter in the upcoming volume entitled *Hidden Markov Models and Applications* to be published in the book series *Unsupervised and Semi-Supervised Learning*, Springer, under the title "A Roadmap to Hidden Markov Models and A Review in its Application in Occupancy Estimation" by Samr Ali and Nizar Bouguila.
- An experimental advance in the deployment of HMMs in the sector of Energy and Sustainability with a study case in occupancy detection.
    - This work is under review in the *Energy and Buildings* journal under the title "Towards Scalable Deployment of Hidden Markov Models in Occupancy Estimation: A Novel Methodology Applied to the Study Case of Occupancy Detection" by Samr Ali and Nizar Bouguila.
- Propose a novel variational inference approach for Beta-Liouville (BL) HMM that is capable of modeling proportional sequential data. Variational inference mitigates hindrances in the parameter estimation whereas the traditional method is prone to sensitivity in initialization as a point estimate, and computationally expensive sampling-based techniques are not guaranteed to converge.

- This work was published and presented as: *S. Ali and N. Bouguila, "Variational learning of beta-liouville hidden markov models for infrared action recognition," in 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2019* [46].
- We also evaluate the proposed model approach for action recognition in both the infrared and the visible spectra and implement a fusion scheme to improve accuracy results. This work is published as in *S. Ali and N. Bouguila, "Multimodal Action Recognition Using Variational-based Beta-Liouville Hidden Markov Models" in IET Image Processing, 2020, 14, (17), p. 4785-4794* [47].

- Investigations of hybrid generative discriminative approaches for proportional HMMs and their validation on the categorization of dynamic textures task. Our research has been published as:
  - Dirichlet and Beta-Liouville HMM-based hybrid approach: *S. Ali and N. Bouguila, "Dynamic texture recognition using a hybrid generative discriminative approach with hidden markov models and support vector machines," in 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP). IEEE, 2019, pp. 1–5* [48].
  - Generalized-Dirichlet HMM-based approach: *S. Ali and N. Bouguila, "Hybrid generative-discriminative generalized dirichlet-based hidden markov models with support vector machines," in 2019 IEEE International Symposium on Multimedia (ISM). IEEE, 2019, pp. 231–2311* [49].

- MAP estimation of proportional HMMs. This is because while both the latter and variational inference are approximation approaches, the MAP method has a lower computational cost than variational inference. Moreover, they both share the same fundamental principle of placing appropriate priors over the parameters to be estimated for improving the performance of the evaluation. The priors that are chosen over the parameters in the MAP technique smooth the likelihood function; hence, reducing its multimodal nature. In turn, this improves the approximation of the desired global maximum. The investigation carried out in [50]

also motivates further exploration of the MAP methodology in the estimation of the HMM parameters.

  - The MAP approximation of the Dirichlet and the BL HMMs was accepted and presented as:

    S. Ali and N. Bouguila, "Maximum A Posteriori Approximation of Dirichlet and Beta-Liouville Hidden Markov Models for Proportional Sequential Data Modeling," Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2020), Toronto, October 2020 [2].

  - The MAP approximation of the Generalized Dirichlet (GD) HMMs was accepted and presented as:

    S. Ali and N. Bouguila, "On Maximum A Posteriori Approximation of Hidden Markov Models for Proportional Data," Proceedings of the IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP 2020), Tampere, October 2020 [51].

- Incorporation of a simultaneous feature selection paradigm in proportional HMMs.

  - This work is accepted in the *IEEE Transactions of Neural Networks and Learning Systems (TNNLS)* under the title "Maximum A Posteriori Approximation of Hidden Markov Models for Proportional Sequential Data Modeling with Simultaneous Feature Selection" by Samr Ali and Nizar Bouguila.

- Infinite extension of Dirichlet and Beta-Liouville HMMs. The inference is based on variational learning. The validation is carried out on video anomaly detection.

  - This work is to be submitted to a journal with the title "Towards An Efficient Anomaly Detection in Videos: An Infinite Hidden Markov Model Approach" by Samr Ali and Nizar Bouguila.

- Infinite extension of the Generalized Dirichlet HMM as well as the incorporation of simultaneous feature selection. The validation is carried out on video anomaly detection.

- This work is under review in the *IEEE Transactions on Multimedia* with the title "Towards Efficient Anomaly Detection in Videos: An Infinite Hidden Markov Model with Simultaneous Feature Selection" by Samr Ali and Nizar Bouguila.

- Online learning for variational inferred Beta-Liouville HMMs with the online Dirichlet HMMs utilized for benchmarking. Thus far all the HMMs discussed have assumed an offline deployment. That is the model does not adapt to new data as it becomes available since the training is performed once for a static model. Online models incorporate such new data. Furthermore, incremental ones (a subcategory of them) do not forget the original parameters as dynamic training is performed. This work has been published and presented as:

  - *S. Ali and N. Bouguila, "Online learning for beta-liouville hidden markov models: Incremental variational learning for video surveillance and action recognition," in 27th IEEE International Conference on Image Processing (ICIP 2020), 2020* [52].

1.3 Thesis Overview

The thesis mainly follows a manuscript-based organization. In terms of the contributions, the sole other author in all the manuscripts is my supervisor. The remainder of the thesis is organized as follows:

- Experimental innovation in HMM deployment in occupancy detection is presented in Chapter 2.

- Chapter 3 presents the variational inference of the Beta Liouville-based HMMs and multimodal fusion in action recogntion.

- Chapter 4 discusses the hybrid generative discriminative approach based on proportional HMMs.

- Chapter 5 details MAP approximation of proportional HMMs.

- Chapter 6 introduces simultaneous feature selection for proportional HMMs.

- Chapter 7 extends proportional HMMs to infinity.
- Chapter 8 incorporates simultaneous feature selection for infinite proportional HMMs.
- Chapter 9 presents the setting of online deployment and incremental learning.
- The conclusion and future works are discussed in Chapter 10.

Chapter 2

Towards Scalable Deployment of Hidden Markov Models in Occupancy Estimation: A Novel Methodology Applied to the Study Case of Occupancy Detection

> Information is the oil of the 21st century, and analytics is the combustion engine
>
> Peter Sondergaard, Gartner Research

To start off, we present an innovation in the experimental deployment of hidden Markov models (HMMs). This pertains to the occupancy estimation and detection applications and impacts the energy and sustainability sector. On the other hand, one of the modern world's major issues is the conservation of energy and sustainable development.

Buildings are a major component of society and are integral in such efforts. A report released on building energy efficiency by the World Business Council for Sustainable Development states that buildings are responsible for at least 40% of energy use in many countries, mainly from fossil fuels [53, 54].

HVACL (Heating, Ventilation, Air Conditioning and Lighting) systems utilize about half of this amount in industrialized countries [55, 56]. Improving energy efficiency through better control strategies is a highly researched area. Such HVACL strategies already in place rely heavily on predetermined occupancy times as well the number of occupants [57]. Due to such presumptions, a large amount of energy consumed is actually wasted. This can be overcome by relying on the actual occupancy of the building [58].

For highest control efficiency, a real time input of occupancy information to the systems is required [59]. Real-time occupancy estimation is essential in evacuation of buildings and other emergencies [60]. Furthermore, on the long run, these monitored buildings may be used for the prediction of future usage of the occupied space with such occupancy estimation information [61, 62]. This chapter presents a thorough experimentation and analysis of the proposed setup.

2.1 Introduction

Internet of Things (IoT) represents an integral block in the future of data science and artificial intelligence. This is due to its ability to formally connect automation systems at all levels and pool data to the cloud. This data may be used for analysis of results as well as investigating the various underlying patterns to better assist users.

Smart cities may be categorized under this umbrella of which our topic of interest is occupancy detection and estimation. Occupancy detection refers to identifying whether the space that is usually monitored with IoT sensors has people at any given time. On the other hand, occupancy estimation refers to the process of finding an approximation of the exact number of people that are occupying a monitored space at a time. Naturally, the latter case also considers zero number of occupants.

The objective behind identifying whether there are occupants or not is for profiling energy in smart buildings [1]. Indeed, it has been established that energy consumption in smart buildings can be reduced by 40% by only performing occupancy detection [63], [64], [65]. These are significant findings and motivate further analysis into methods for superior modeling in this domain.

Machine learning has recently revolutionized various fields. Indeed, its application in occupancy detection is ubiquitous. For instance, the use of random forests was investigated in [66] and [1] as

well as support vector machines (SVMs) and K-nearest neighbour (KNN) in [67]. Models such as random forests, SVMs, and KNN fall under the category of discriminative machine learning models.

Such models are efficient in learning the boundary between the differently labelled data as they deduce the conditional probability of a class given the data [8]. This allows them to usually render higher accuracy levels than generative models; i.e., the other category. However, generative models deduce the joint probability and hence uncover the underlying distribution and pattern of the data. This also renders these models to be less susceptible to over-fitting.

In this chapter, we focus on the application of HMMs and their efficient deployment in the occupancy detection problem. HMMs fall under the generative machine learning models. They are known as one of the most prominent sequential modeling techniques in machine learning [10]. Indeed, their impact in speech recognition and its applications is well-known [3]. Moreover, it has been recently successfully applied in other fields such as infrared action recognition [46] and dynamic texture classification [52].

HMMs are also commonly supported models by researchers in the area of occupancy detection due to its structural suitability with the data and its nature [68],[69],[70]. Nonetheless, the influence of the mathematical behavior of HMMs as well as their understanding remain inconspicuous in the majority of smart buildings' studies. Indeed, some of its parameters; i.e., the transition matrix, are usually set with manual probability computation which may compromise the integrity of the model. In contrast, in this paper, the transition matrix values are determined with the update equations of the Baum Welch algorithm; i.e., the parameter learning algorithm of HMMs [3, 10].

Further, this chapter aims to explain the behavior of HMMs as well as present a consequent novel setup to this field. The proposed approach constitutes of the independent training of a HMM for each of the classes. To assign a label for a testing observation, the likelihoods are computed then the label is assigned according to the maximum. Furthermore, we also present a summary of multiple future work venues. This is in order to address the advancement of the understanding and integration of machine learning models, especially HMMs, in the field of occupancy detection and potential better results in occupancy estimation in smart buildings.

Hence, the contributions of this chapter can be listed as follows:

- Novel scalable methodology for occupany detection using HMMs. The approach may also

be generalized easily for occupancy estimation of any number.

- First comprehensive study with an in-depth understanding of the influence of the behavior of continuous Gaussian-based HMMs on the results of occupancy detection in smart buildings.
- An investigation of the optimal number of states as well as a study of the best independence criteria for the representation of the data and how that finally influences the accuracy.
- Discussion of multiple areas of future study for the application of HMMs in smart buildings. This aims to open up this topic for better understanding of the implementation of machine learning models in IoT applications; especially in the case of smart buildings.

2.2 Materials and Methods

2.2.1 Hidden Markov models

The latent variables forming the Markov chain are often referred to as the hidden states of the HMM. A graphical representation of the HMM can be seen in Figure 1.1. We note that such a diagram depicts a first-order HMM since each state in the Markov chain depends only on the previous one. Hence, an n^{th}-order HMM is simply one with a Markov chain structure in which each state depends on the prior n states.

Mathematically, a HMM is characterized by an underlying stochastic process with K hidden states that form a Markov chain. Each of the states is governed by an initial probability π, and the transition between the states at time t can be visualized with a transition matrix $B = \{b_{ii'} = P(s_t = i'|s_{t-1} = i)\}$. In each state s_t, an observation is emitted corresponding to its distribution which may be discrete or continuous. This is the observable stochastic process set.

The emission matrix of the discrete observations can be denoted by $\Xi = \{\Xi_{it}(m) = P(X_t = \xi_m|s_t = i)\}$ where $[m, t, i] \in [1, M]\times[1, T]\times[1, K]$, and the set of all possible discrete observations $\xi = \{\xi_1, ..., \xi_m, ..., \xi_M\}$. On the other hand, the respective parameters of a probability distribution define the observation emission for a continuous observed symbol sequence. The Gaussian distribution is most commonly used which is defined by its mean and covariance matrix $\varkappa = (\mu, \Sigma)$ [10, 14, 15]. Consequently, a mixing matrix must be defined $C = \{c_{ij} = P(m_t = j|s_t = i)\}$ in

the case of continuous HMM emission probability distribution where $j \in [1, M]$ such that M is the number of mixture components in set $L = \{m_1, ..., m_M\}$. Hence, a discrete or continuous HMM may be defined with the following respective parameters $\lambda = \{B, \Xi, \pi\}$ or $\{B, C, \varkappa, \pi\}$. In this chapter, our approach is based on a continuous HMM with a single Gaussian distribution in each of its states. The parameters are approximated using the well-known Baum Welch algorithm.

Figure 1.3 shows transitions then when they become trellis or lattice. We note in both figures that any of the states can be visited from any other state. This type of HMMs is known as ergodic or fully connected whereby the transition probability between any two states is nonzero [10] and is the structure that we use throughout this chapter.

2.2.2 Estimation of the Parameters

There are three famous problems for HMMs; namely, the decoding, the evaluation, and the learning. In this chapter, we require the solution of two latter problems in order to address the task at hand. Particularly, we use the *Baum Welch* algorithm to address the learning problem and the *Forward* algorithm to solve the evaluation problem. Though they are both prevalent in the literature and well-defined, we aim to provide the reader with a summary of each in this section.

The Baum Welch algorithm is the traditional method to compute the parameters of HMMs. It may be viewed as the expectation maximization algorithm for HMMs. It is also referred to as the Forward-Backward algorithm. This is due to its dependency on the two algorithms; the Forward and the Backward algorithms. Accordingly, we first begin by the definition of the Forward algorithm and draw relationships between the various dependencies of the algorithms and their parameters.

The Forward algorithm is responsible for the calculation of the probability of being in state $s_t = i$ in a given HMM λ after a partial observation sequence. The intermediate forward variable $\rho_t(i) = P(X_1, X_2, \ldots, X_t, s_t = i|\lambda)$ can then be computed recursively. The process starts with initiating $\rho_1(i) = \pi_i \varkappa_G^i(X_1)$ where $\varkappa_G^i(.)$ refers to the Gaussian distribution that models the emission probability for the observation in state $s_t = i$. This is carried out for all states $1 \leq i \leq K$ given K states.

Next, computation of reaching the next state $s_{t+1} = j$ from all possible K states is carried out with a sum over their product. This may be mathematically denoted by $\rho_{t+1}(j) = \sum_{i=1}^{K} \rho_t(i) b_{ij} \varkappa_G^j(X_{t+1})$

where $t = 1, 2, \ldots, T-1$ and $1 \leq j \leq K$. This finally allows us to compute the likelihood that is the result of the Forward algorithm $P(X|\lambda) = \sum_{i=1}^{K} \rho_T(i)$.

On the other hand, the Backward algorithm computes the tail probability of partial observation from $t+1$ to the end or T. This is performed in a similar fashion to the forward algorithm with the intermediate backward variable $\theta_t(i)$ initialized to 1. Together, they are used to compute the Baum Welch algorithm with a guarantee to converge into more compact clusters with a requirement of an initial clustering. Two important intermediate variables are defined for the execution of the Baum Welch algorithm. The first is $\omega_t(i,j) = P(s_t = i, s_{t+1} = j|X, \lambda)$. This is the probability of starting at $s_t = i$ and transitioning at $t+1$ to state j with b_{ij} given λ and X. The other intermediate variable is $\varpi_t(i)$ which is the expected number of transitions from i to j. This is mathematically denoted by $\sum_{j=1}^{K} \omega_t(i,j)$.

It is also noteworthy to recall the two conditional independence assumptions that allow for the tractability of these algorithms [3]. The first is that given the previous hidden state, the current hidden state is independent of all other variables such that:

$$P(s_t|s_{t-1}, X_{t-1}, \ldots, s_1, X_1) = P(s_t|s_{t-1}) \tag{56}$$

The second assumption is that given the current hidden state, the current observation is independent of all other variables such that:

$$P(X_t|s_T, X_T, s_{T-1}, X_{T-1}, \ldots, s_{t+1}, X_{t+1}, s_t, s_{t-1}, X_{t-1}, \ldots, s_1, X_1) = P(X_t|s_t) \tag{57}$$

2.2.3 Proposed approach

We adopt a train by class model for HMMs in this chapter. This proposed approach has not been investigated in the smart buildings domain to the best of our knowledge. However, it has been proven to be efficient in computer vision applications such as in [46, 48, 52]. These results have motivated us to propose and investigate such a setup in this chapter.

The proposed approach can be observed in Fig. 2.1. As shown, a model is trained for each of the classes corresponding to no occupants and occupants detected. Then, the likelihoods of the testing

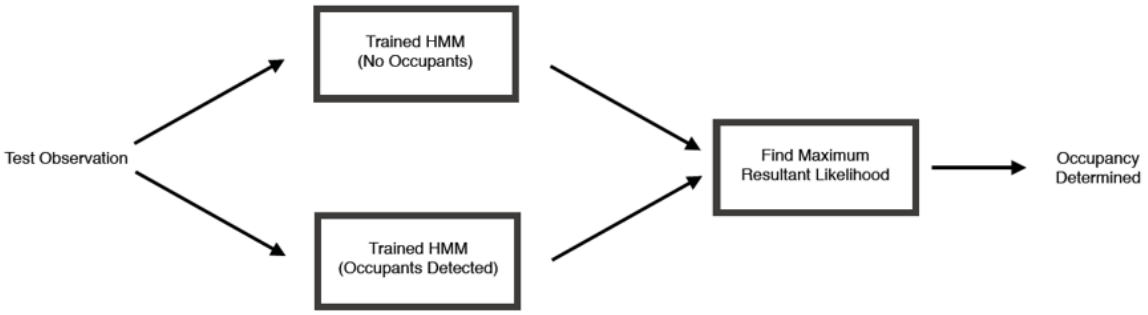


Figure 2.1: Proposed approach setup.

observations are computed using both trained models. This is executed using the forward algorithm [3]. The final label is then assigned based on the maximum likelihood. This aims to improve the modeling of the occupancy detection problem as well as provide a scalable and stable approach for its efficient deployment. In Section 2.3, we investigate the implications of employing the proposed approach in comparison to the traditionally utilized HMM approach in occupancy detection.

Furthermore, the significance of this chapter may also be highlighted through the fact that occupancy detection may be viewed as a special case of occupancy estimation. That is because occupancy estimation would then require a trained HMM for each number of occupants to be considered whereas occupancy detection only requires two. As such, this chapter represents a preliminary and promising result for a scalable HMM-based framework for occupancy estimation that is stable regardless of the range of occupants that is assumed.

2.3 Experiments

In this section, we present the details carried out for investigating the proposed approach and discuss the results. In particular, we introduce the dataset that we used in Section 2.3.1 and the HMM benchmark methodology in Section 2.3.2. Next, we define the evaluation metrics that we report in Section 2.3.3. Finally, we study the model selection problem in Section 2.3.4 and the complexity of the model parameters as well as dependency in Section 2.3.5.

2.3.1 Dataset

We evaluate our approach on a set of training and testing sets whose sizes and class distributions are summarized as shown in Fig. 2.2. It has been first published in [1] and is utilized in other prominent occupancy detection papers [71, 70]. Hence, this dataset may be considered as a benchmark for the occupancy detection task. This motivated our choice of the dataset and allows researchers to easily explore the performance of various other techniques. The collection of the data is performed with an IoT platform and an interested reader is referred to the original paper for the data collection process. It is noteworthy to mention that the data is time-series as its collection occurs with the passage of time. That is the readings of the sensors are collected in time which as a whole forms a time series data such that there are temporal relations.

Data set	Number of observations	Data Class Distribution (%)	
		0 (non-occupied)	1 (occupied)
Training	8143	0.79	0.21
Testing 1	2665	0.64	0.36
Testing 2	9752	0.79	0.21

Figure 2.2: Class distribution and size of training and testing datasets used [1].

For the experimental setup, we maintain the same training and testing data splits as used in the literature. This is performed to facilitate mass comparison with methods in the literature and potential future ones. Consequently, the training of the proposed model is performed on data that is collected with the door mostly closed during detection of occupancy duration, which matches the environmental conditions for testing set 1. In contrast, testing set 2 contains data that has been with the door mostly open. It is noteworthy to mention that this also plays part in the interpretation of the results that we detail in Section 2.3.4 and 2.3.5. It is noteworthy to mention that we chose to maintain the same split of the data as the original paper rather than use K-fold cross validation for the interpretability of the results in relation to the variability conditions of the collected data. Moreover, this allows the chance for the results to be benchmarked with others in the literature as well as future ones that maintain the same split. Finally, this setup maintains the integrity of the data, while a K-fold cross validation evaluation may be performed as a future investigation.

The datasets constitute of averaged recordings that have been collected at 14 second or 3 to

4 times per minute intervals. The number of observations and the data class distributions can be observed in Fig. 2.2. An interested reader is referred to [1] for extensive details and analysis of the features including the correlation matrix.

2.3.2 Benchmark Setup

We benchmark against the traditional methodology of applying HMMs in occupancy detection and estimation. The pipeline constitutes of assuming that each state in the model represents a class of occupancy. Hence, in the case of occupancy detection that would be two: no occupancy and occupancy detected. However, this approach relies on the assumption that the underlying relationship between the various features is best represented correspondingly to the number of labels available.

In the case of estimation, the number of states would have to increase according to the number of occupants to be detected. This renders the traditional approach to be less scalable than the proposed approach. This is due to the requirement of training a single model for the entirety of the available labels at any given time. We also study the state and parameter complexity further in Section 2.3.5.

Imbalanced data where the number of observations in each class is not the same is a significant problem to address. This is due to the fact that datasets in occupancy detection and estimation are usually characterized by a large percentage of data under the no occupants label in non-residential buildings. This logically follows the fact that such areas only contain occupants during work hours and do not have facilities for slumbering. Moreover, this phenomenon is nowadays more prevalent than ever due to the circumstances that we are facing because of COVID19. This may also be observed in the dataset that we are experimenting on in this chapter.

It is imperative to mention that the proposed approach mitigates the problem of imbalanced data in comparison to the benchmark. This is due to the process of independent training of the models for each of the data classes. Consequently, the parameters of each of the models is inferred given the data available and given that the HMM is a generative machine learning model, the underlying distribution of the data is then deduced. Furthermore, overfitting is also less of a concern in generative models which is an overall advantage for using HMMs for occupancy detection and estimation tasks.

2.3.3 Evaluation Metrics

Accuracy may be defined as the percentage of correctly classified observations. This can be computed with the following mathematical formula:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{58}$$

where TP denotes the number of true positives which represents the number of instances where occupancy detected labels are also predicted as such. On the other hand, FP denotes the number of false positives which represents the number of instances where occupancy is not detected but are predicted to have been. TN and FN follow a similar analogy whereby they denote the number of true negatives and false negatives, respectively.

We also report the following performance criteria for the chosen number of states:

$$Precision = \frac{TP}{(TP + FP)} \tag{59}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{60}$$

$$F1 - Score = \frac{2 * (Recall * Precision)}{(Recall + Precision)} \tag{61}$$

The recall allows us to measure the missed positive predictions and hence the coverage of the minority class in an imbalanced data. This translates to the positive label or the detection of occupancy in this chapter. On the other hand, precision quantifies how many of the positive class predictions actually belonged or fall under the positive class. The F1-Score or the F-measure is the harmonic mean between the two latter measures and represents a weighed measure of both.

2.3.4 Model Selection

Model selection refers to the process of identifying the appropriate size of the model; i.e., selecting the best representative model structure. In HMMs, this then translates to determining the

Table 2.1: Accuracy of the HMM for both the benchmark and novel approaches across different number of states.

		Accuracy (%)				
K (No. of States)		2	3	4	5	6
Benchmark Approach	Test set 1	**91.52**	91.52	65.14	83.79	55.80
	Test set 2	**57.56**	55.18	47.39	44.49	43.35
Novel Approach	Test set 1	92.50	**97.75**	95.72	91.52	90.43
	Test set 2	92.83	**97.01**	73.72	67.16	64.21

optimum number of states to represent the occupancy observations. We carry out an experimental investigation in this chapter to meet this end.

On the other hand, another choice that we may have taken into account is applying model selection techniques that are based on information criteria such as the famous Akaike information criterion (AIC) or Bayesian information criterion (BIC). Though these are usually utilized when a maximum likelihood approach is presented, a recent investigation [72] found that they are not suitable for HMMs in some cases dependent on the nature of the problem and the corresponding state solution.

This presents an interesting venue for future work in this application, albeit infinite HMMs are capable of presenting a dynamically flexible structure of the model without the need for an extra measure or experiments to determine the optimum number of states. Hence, the latter represents a better solution. Nonetheless, in this chapter, we have considered the complexity of the model in terms of the number of states versus the resultant accuracy for both the proposed approach as well as the benchmark method.

The results of our model selection experiments are shown in Table 2.1. Each pair of rows represents the results for an approach with each row reporting the results for a particular testing set. It can be observed that the highest accuracy is achieved for the two states-based model in the benchmark approach, while 3 states-based structure is most suitable for the proposed novel approach.

An interesting interpretation then follows for the benchmark approach whereby the HMM was indeed able to discern the labels and catalogue them into corresponding states. This best representation is formed when the number of states is assumed to be the same as the number of classes. On

the other hand, the superiority of the proposed method can be clearly shown in terms of the increase in the accuracy.

Indeed, we notice a difference of 6.23% and 39.45% increase for testing sets 1 and 2, respectively. The relatively lower increase in accuracy in testing set 1 can be explained by the match between the environmental conditions of its observations and of the training set. The boost in the accuracy of the second testing set clearly shows the generalization ability of the novel approach as well as its superiority in a better representation of the data.

This is also intuitively shown by the structure of the approaches themselves whereby an entire HMM is used to infer the distribution and process of the observations of each class in the proposed approach. In contrast, in the benchmark approach, all of the data is represented by one model with an assumption that may not always be fulfilled. That is the number of states of the optimum model is directly proportional to the number of labels and its different parameters can be inferred correctly accordingly.

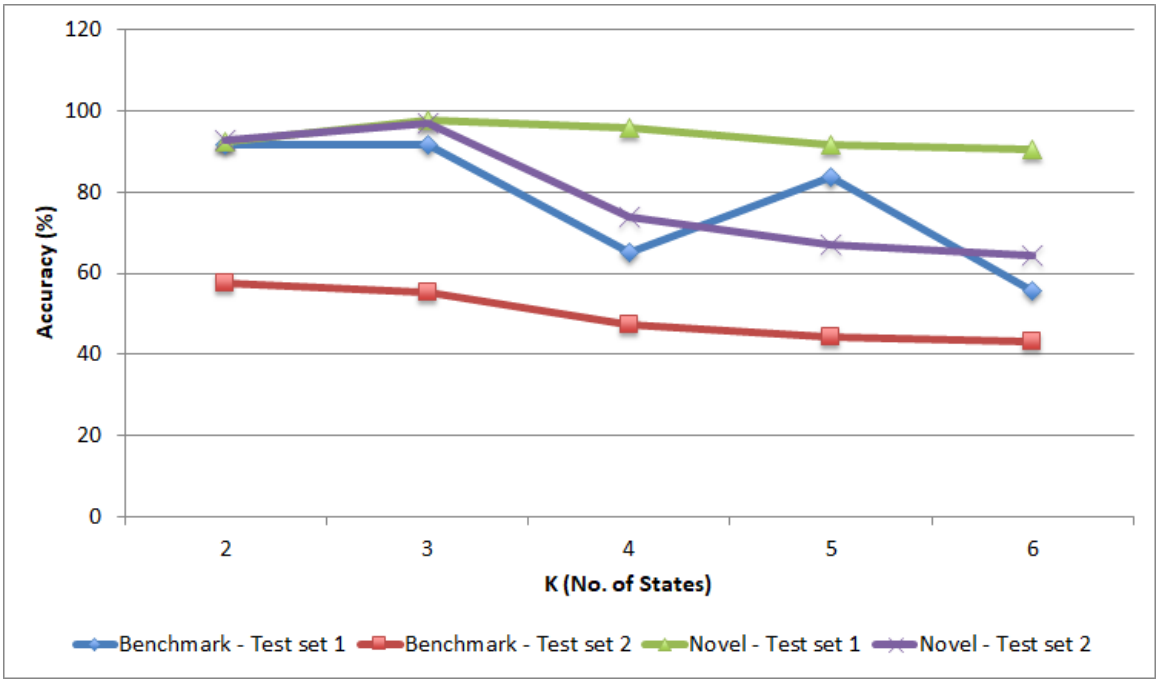


Figure 2.3: Investigation of the accuracy versus the number of states for model selection of the HMM for both the benchmark and novel approaches.

We also visualize the trend of the accuracies across the different models as shown in Fig. 2.3. Both the benchmark and proposed HMM approaches show degradation in the results after reaching

Table 2.2: Accuracy of the HMM for both the benchmark and novel approaches across different number of states with the independence assumption.

		Accuracy (%)				
K (No. of States)		2	3	4	5	6
Benchmark Approach	Test set 1	**79.88**	62.48	53.73	45.52	65.95
	Test set 2	**53.65**	44.09	41.48	33.08	37.85
Novel Approach	Test set 1	95.20	**96.96**	92.46	79.06	85.33
	Test set 2	95.27	82.47	**95.80**	82.55	83.51

the optimum number of states. Nonetheless, instability of the model representation can be discerned in the trend of testing set 1 (shown in blue) whereby a fitted line would show degradation but the data has a jump at $K = 5$.

2.3.5 State Complexity and Parameter Dependency

It is also noteworthy to discuss the number of parameters for the model. This is mainly dependent on the number of states. As such, a tradeoff relationship exists between the increase in accuracy and the number of states with the objective of achieving optimum accuracy. The performance of the HMM then degrades as more states become unreflective of the intrinsic behavior of the modeled problem; i.e., occupancy detection in this chapter.

In order to then study the complexity of the model in terms of the parameters, we differentiate between the state-based parameters and the HMM-based one. In particular, we note that state-based parameters are dependent on the nature of the emission probability function that we choose which is a Gaussian as discussed. The latter, on the other hand, pertains to the HMM structure itself which we detailed in Section 2.2. This is computed dependent on the number of states as $K * K$ for the transition matrix parameters and K initial parameters.

As to the state-based parameters, we also include dependency on the assumption of independence in our investigation. This translates to a full covariance matrix in the case of the investigations that we carried out in Section 2.3.4 which translates to $D * D$ in each of the states where D is the number of features in the dataset; i.e., 5 and D means for each of the states. If assuming that each of the dimensions of the distribution is independent then we have a diagonal covariance matrix ($D * D$ is reduced to D for each state) whereby each of the dataset features is assumed to be independent.

Table 2.3: Accuracy fluctuation of the benchmark method across 10 runs when independence of features is assumed.

	Accuracy (%)				
Number of States (K)	2	3	4	5	6
Test Set 1					
Average	79.88	62.48	53.73	45.52	65.95
Minimum	12.65	13.25	13.06	2.74	12.83
Maximum	87.35	81.05	86.42	74.71	79.06
Test Set 2					
Average	53.65	44.09	41.48	33.08	37.85
Minimum	45.44	35.75	33.31	5.05	25.87
Maximum	54.56	48.54	53.74	57.03	42.82

Table 2.4: Precision fluctuation of the benchmark method across 10 runs when independence of features is assumed.

	Precision (%)				
Number of States (K)	2	3	4	5	6
Test Set 1					
Average	83.18	77.24	79.43	79.83	93.55
Minimum	16.35	19.12	16.88	5.57	63.53
Maximum	90.61	100.00	100.00	100.00	100.00
Test Set 2					
Average	82.45	79.56	84.72	81.83	96.71
Minimum	54.02	55.39	53.64	20.66	79.01
Maximum	85.61	86.69	99.98	99.78	100.00

Table 2.5: Recall fluctuation of the benchmark method across 10 runs when independence of features is assumed.

	Recall (%)				
Number of States (K)	2	3	4	5	6
Test Set 1					
Average	79.88	62.48	53.73	45.52	65.95
Minimum	12.65	13.25	13.06	2.74	12.83
Maximum	87.35	81.05	86.42	74.71	79.06
Test Set 2					
Average	53.65	44.09	41.48	33.08	37.85
Minimum	45.43	35.75	33.31	5.05	25.87
Maximum	54.56	48.54	53.74	57.03	42.82

Table 2.6: F1-Score fluctuation of the benchmark method across 10 runs when independence of features is assumed.

	F1-Score (%)				
Number of States (K)	2	3	4	5	6
Test Set 1					
Average	80.26	66.74	57.17	54.76	73.87
Minimum	14.26	15.65	14.72	3.67	20.29
Maximum	87.59	84.02	86.91	84.20	87.27
Test Set 2					
Average	56.41	52.83	50.20	45.08	52.00
Minimum	49.36	43.45	46.85	8.11	38.83
Maximum	57.20	55.75	56.99	71.62	57.60

This assumption may be followed for simplicity or to meet a lower computational cost requirement. It is prevalent in the literature when the dimensions or the features of a dataset is large in size.

As expected, we found out that the accuracy decreases in the latter case for both approaches as shown in Table 2.2. It is noteworthy to mention that the accuracy has not degraded significantly even with an independence assumption using the novel model. We also notice the flexibility of the model whereby more states were found to be optimum for testing set 2 in order to maintain the performance level.

Note that for the benchmark approach, we reported average accuracy results. This shows another interesting result that supports the stability and robustness of the proposed model in comparison to the benchmark. As shown in Table 2.3, the benchmark results vary greatly across ten runs whereas the novel approach was the same. The minimum, average, and maximum results of precision, recall, and F1-score evaluation metrics for both testing sets can be observed in Table 2.4, 2.5, and 2.6, respectively. The fluctuations are also visualized as boxplots for the accuracy, precision, recall, and F1-score evaluation metrics for testing sets 1 and 2 as shown in Fig. 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10, and 2.11 respectively. In contrast, the confusion matrices for the optimum number of states, under the independence assumption, of the novel approach are shown in Fig. 2.12 and 2.13.

We can observe in Fig. 2.4 that most of the runs yield the maximum accuracy in state 2. Indeed, this occurred in nine of the ten runs. Hence, the classification of the minimum accuracy as an outlier. The fluctuations of the results for the other number of states is more dispersed. This also supports the model selection conclusions that we reached in Section 2.3.4. Similar observations can be made

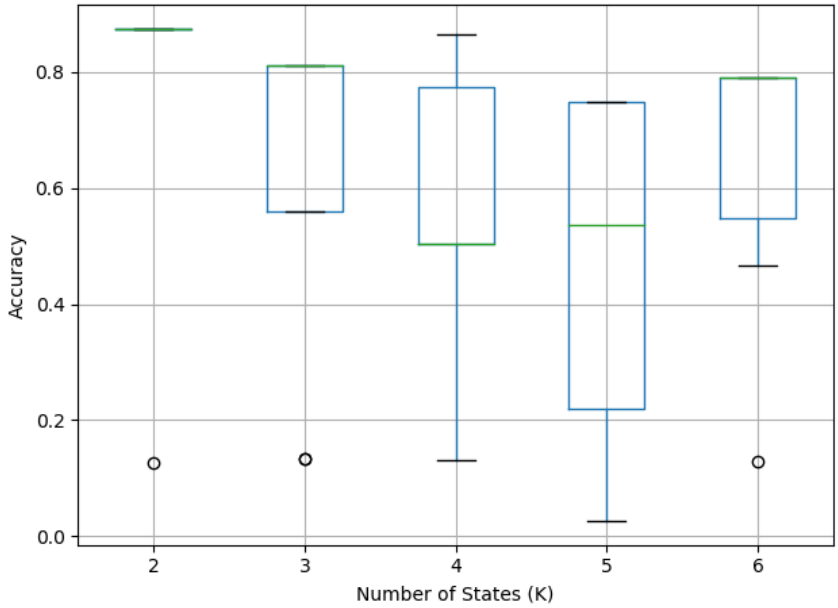


Figure 2.4: Visualization of the accuracy fluctuation of the benchmark HMM approach when independence of features is assumed for test dataset 1.

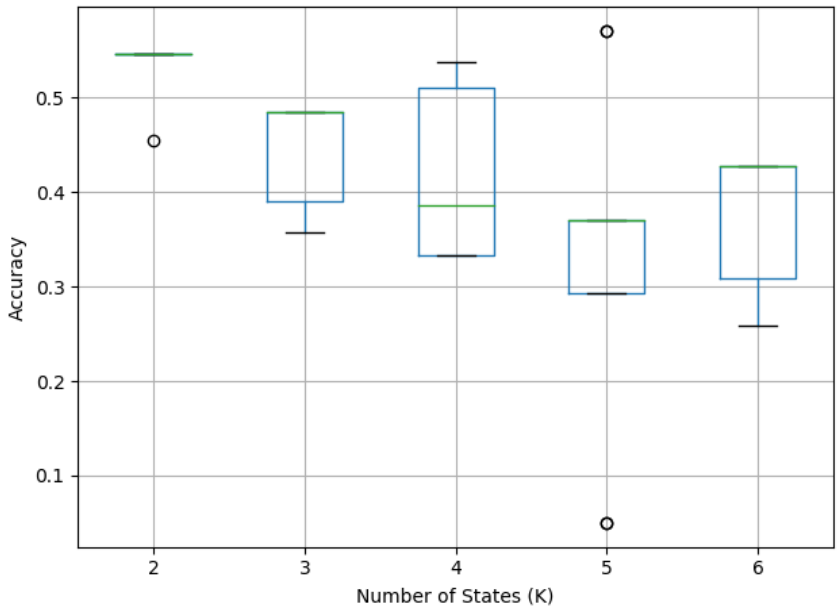


Figure 2.5: Visualization of the accuracy fluctuation of the benchmark HMM approach when independence of features is assumed for test dataset 2.

from Fig. 2.5 for the accuracy of testing set 2 with the benchmark approach under the independence assumption.

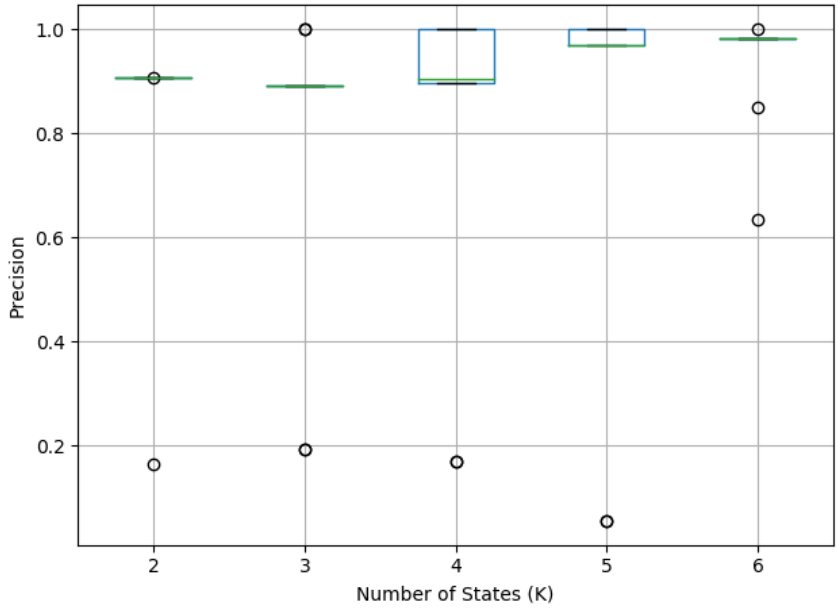


Figure 2.6: Visualization of the precision fluctuation of the benchmark HMM approach when independence of features is assumed for test dataset 1.

The precision pattern in Fig. 2.6 shows an interesting phenomenon though whereby a precision of 100% is reached for 3 states and above, albeit only at a maximum and as an outlier occasionally. This is closely the case in Fig. 2.7 for testing set 2. It is important here to notice that this comes at the cost of affecting the recall as expected and can also be supported by Fig. 2.8 and 2.9.

Hence, even when taking into account all of the metrics in terms of the F1-score, or the harmonic mean between the precision and the recall, as well as the accuracy, the model selection criteria would still venture towards the selection of 2 states as the best choice for the model structure of the benchmark approach. Furthermore, the lowest dispersion of results of F1-score can be reported for the 2 states model, as shown in Fig. 2.10 and 2.11, which further supports this conclusion.

Overall, we reach two important conclusions from the independence assumption experiments. First, that the novel approach is capable of performing comparably well which shows its scalability. This is important in large dimensional datasets where this assumption is usually made. Second, that the novel approach is stable and robust, i.e. consistently reaching the same convergence across ten

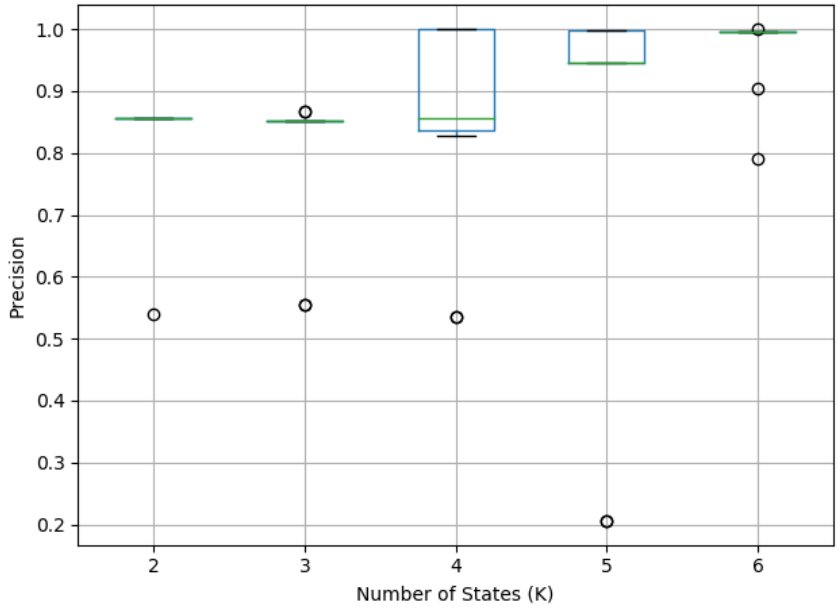


Figure 2.7: Visualization of the precision fluctuation of the benchmark HMM approach when independence of features is assumed for test dataset 2.

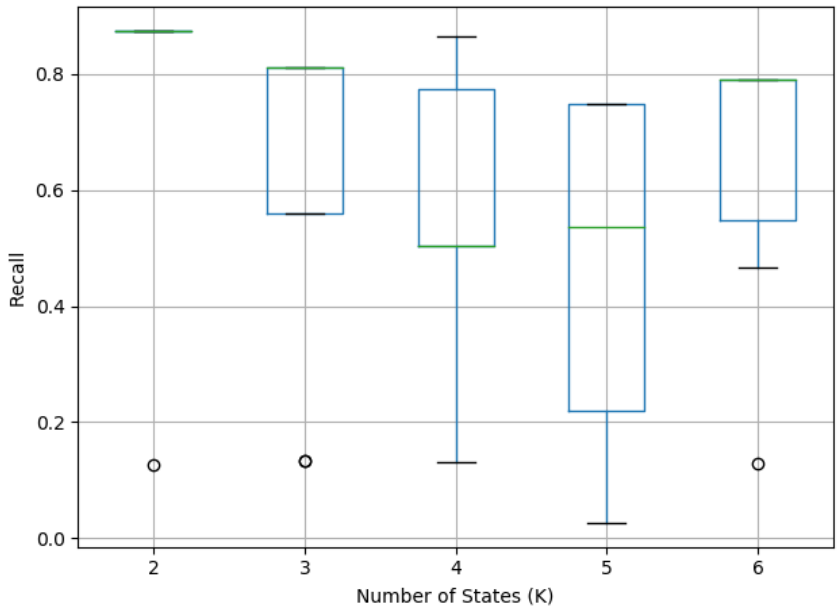


Figure 2.8: Visualization of the recall fluctuation of the benchmark HMM approach when independence of features is assumed for test dataset 1.

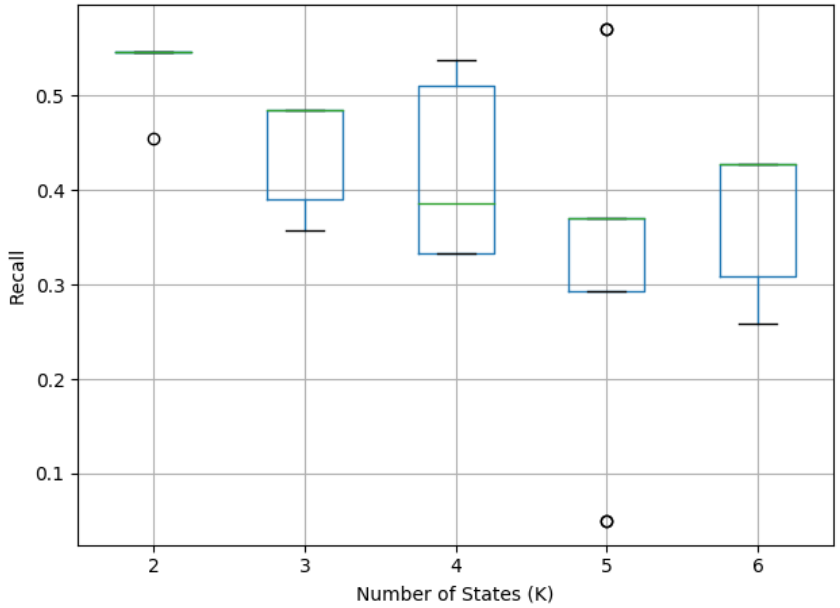


Figure 2.9: Visualization of the recall fluctuation of the benchmark HMM approach when independence of features is assumed for test dataset 2.

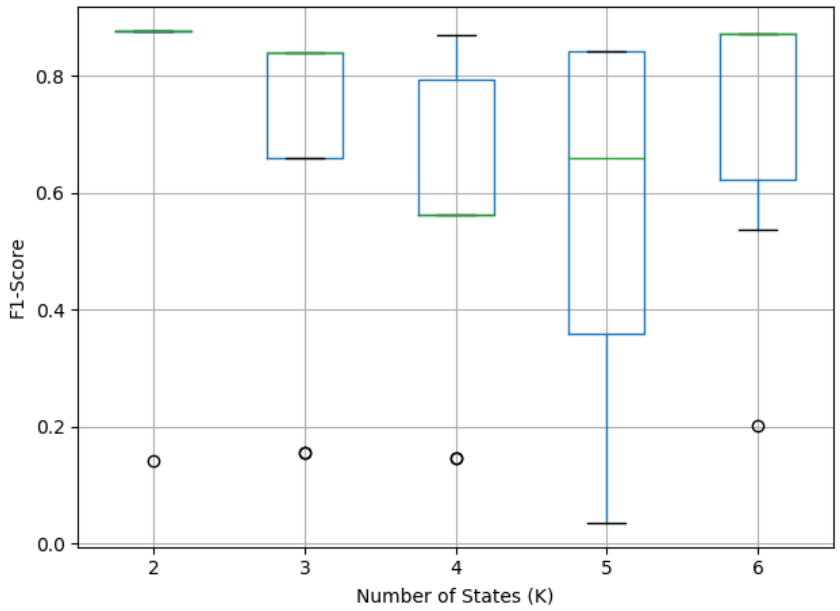


Figure 2.10: Visualization of the F1-score fluctuation of the benchmark HMM approach when independence of features is assumed for test dataset 1.

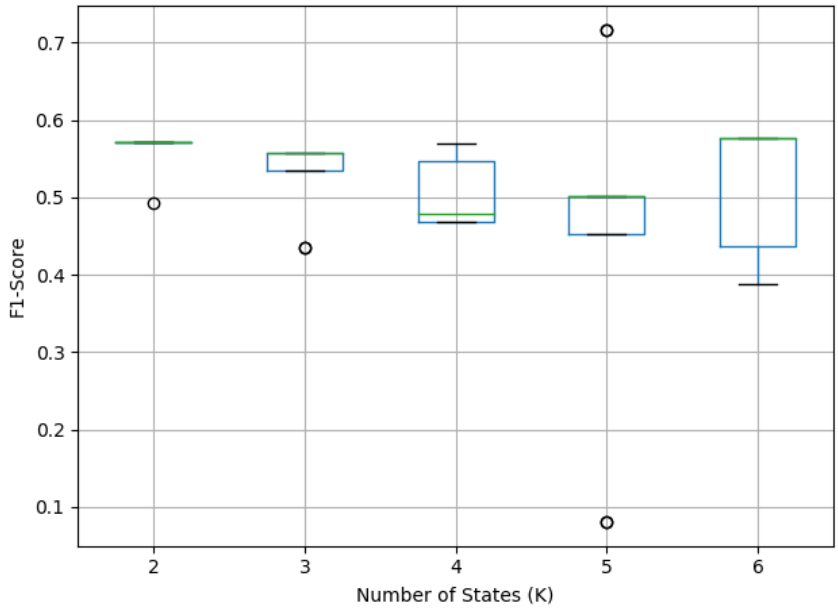


Figure 2.11: Visualization of the F1-score fluctuation of the benchmark HMM approach when independence of features is assumed for test dataset 2.

Table 2.7: Evaluation of the optimum HMM for both the benchmark and novel approaches.

		Accuracy	Precision	Recall	F1-Score
Benchmark Approach	Test set 1	91.52	93.12	91.52	91.66
	Test set 2	57.56	85.92	57.56	60.43
Novel Approach	Test set 1	97.75	94.71	99.38	96.99
	Test set 2	97.01	96.46	89.02	92.59

runs.

Finally, the confusion matrices for the presented novel approach with the optimum number of states can be observed in Fig. 2.14 and Fig. 2.15 for testing datasets 1 and 2, respectively. These represent the final chosen model for the novel approach with $((5 * 5 + 5) * 3 + 3 * 3 + 3) = 102$ parameters. We present the confusion matrices in order to show the class accuracy as well as the overall performance of the novel approach. The computed accuracy, precision, recall, and F1-score of the optimized novel approach for testing sets 1 and 2 are shown in Table 2.7 versus the best achieving benchmark results.

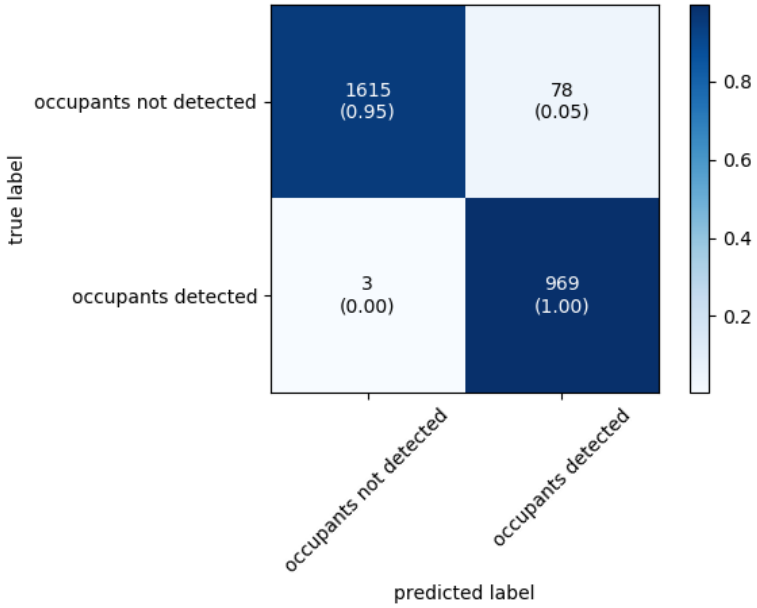


Figure 2.12: Confusion matrix for results achieved on test set 1 for the proposed approach when independence is assumed ($K = 3$).

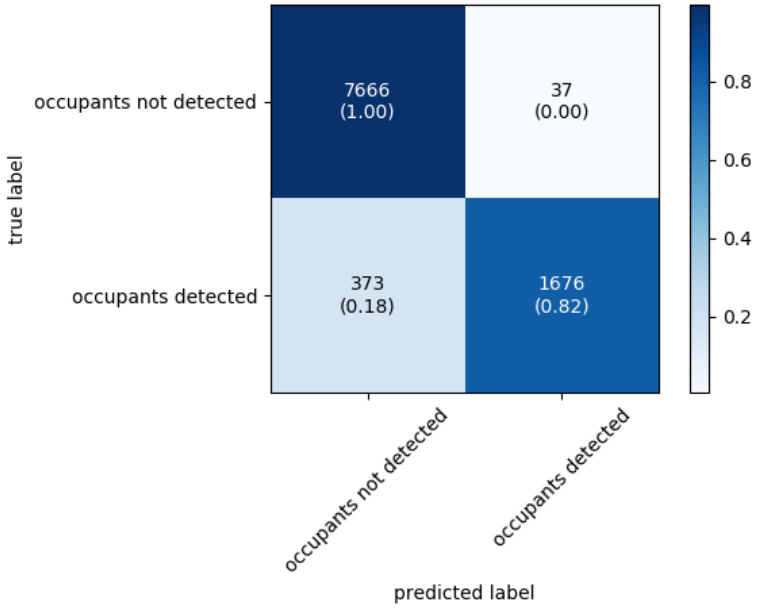


Figure 2.13: Confusion matrix for results achieved on test set 2 for the proposed approach when independence is assumed ($K = 4$).

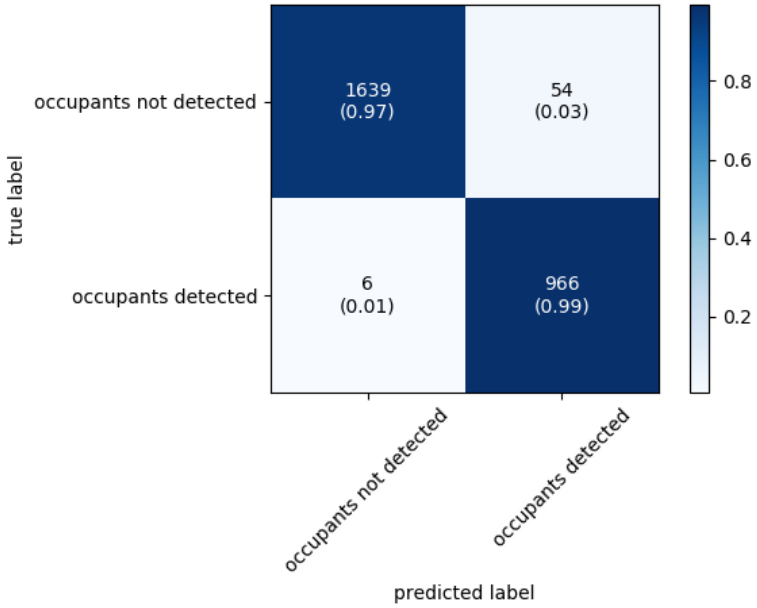


Figure 2.14: Confusion matrix for results achieved on test set 1 for the final chosen model of the proposed approach (Full covariance matrix and $K = 3$).

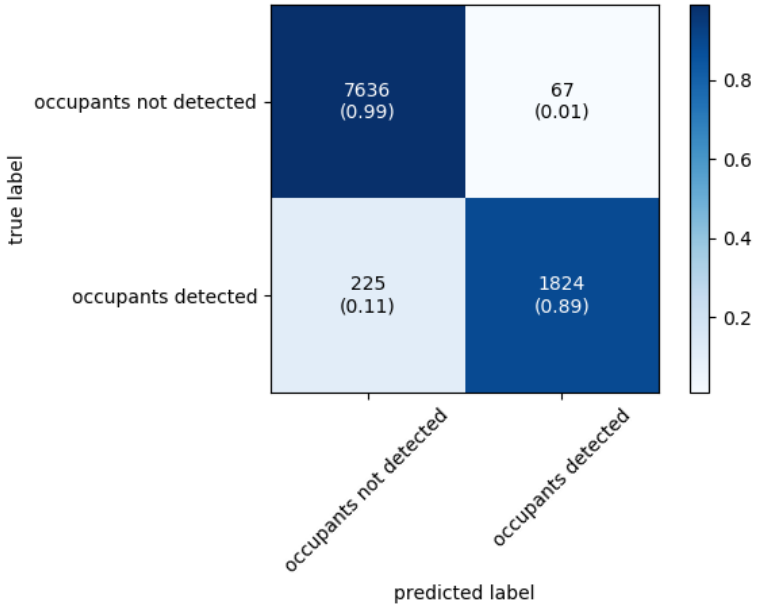


Figure 2.15: Confusion matrix for results achieved on test set 2 for the final chosen model of the proposed approach (Full covariance matrix and $K = 3$).

2.4 Conclusion

In this chapter, we have presented a novel approach for the application of HMMs in occupancy detection. The framework promises a scalable stable deployment of HMMs, especially in relation to the status quo. Future work includes a further study of the approach in occupancy estimation. An infinite HMM approach may also be studied for the dynamic update of the number of states of HMMs without the need to set them experimentally. Furthermore, an online estimation framework of HMMs may also be developed as a framework for the occupancy detection and estimation with incremental learning that adapts the model dynamically in real-time to adjust its parameters for incorporation of new data as it becomes available. Feature selection and feature engineering techniques may also be incorporated for further improvements of the results.

Chapter 3

Variational Inference of Beta-Liouville Hidden Markov Models and Multimodal Action Recognition

> In God we trust. All others must bring data.
>
> W. Edwards Deming

We now turn our attention into deriving novel learning techniques for the estimation of the model in relation to proportional data in particular. In this chapter, this is incarnated by variational inference. This chapter also incorporates an investigation of fusion techniques for multimodal action recognition with the proposed model.

3.1 Introduction

Automatic action recognition (AR) is a fundamental task for many applications such as video retrieval [73], video labeling [74], and video surveillance [74]. Consequently, research attention in AR has recently increased. The typical objective of automatic AR is the assignment of a given video or image sequence; i.e., classification, to a set of predefined classes [75]. Hence, it is dependent on tracking and segmentation as well as other lower level processing stages [76].

Though various approaches for AR has been researched for a number of years, the past decades have witnessed most of the major advances in the field [77]. This is especially true in the case of the visible spectrum where an abundance of data has been made available [78, 79]. These include UCF101 [80], KTH [81], and Weizmann [82] datasets.

However, this does not alleviate the burden of various challenges that still exist in the field. For instance, an individual may still carry out the same action differently than another; i.e., the famous intrinsic within-class variability [75]. Others are specific to the visible spectrum such as its high sensitivity to shadow, background clutter, occlusion, and changes in illumination [83].

On the other hand, utilizing thermal infrared (IR) cameras provides robustness to the aforementioned factors. Specifically, this is due to the relative lower temperatures of shadow, background clutter, and occlusion obstacles. Indeed, capturing humans in poor illumination conditions; i.e., in dim light or at night, is one of its characterizing advantages. Consequently, utilizing IR in AR is a research field with promise in exceeding the performance versus the visible light spectrum [79, 84].

Machine learning techniques that may be used for data modeling in IR AR classically fall under two main categories: discriminative or generative [85]. Generally, discriminative models are trained to infer a mapping between data inputs x to class labels y, while generative models first learn the distribution of the classes before predictions are made [4]. Mathematically, the former represents the posterior probability $p(y|x)$ with the latter denoting the joint probability $p(x, y)$ that is used to calculate the posterior probability accordingly for the classification. Each of the models have their own properties and advantages which we summarize some of them shortly.

Discriminative models usually achieve superior classification accuracy results due to their primary learning objective of the boundary between classes [5]. These include the famous Support Vector Machines (SVM) and decision tree classifiers. On the other hand, generative models require less training data, can be used for outlier detection, and provide the ability to generate more training data with the same input distribution upon completion of the training of the model [86, 87]. Mixture models and Hidden Markov Models (HMM) are examples of generative models.

A HMM [3] is one of the machine learning approaches that may be used for IR AR. A HMM is a principled double stochastic model that uses a compact set of features to extract underlying statistics [3]. Its structure is formed primarily from a Markov chain of latent variables with each

corresponding to the conditioned observation. A Markov chain is one of the least complicated ways to model sequential patterns in time series data. It allows us to maintain generality while relaxing the independent identically distributed assumption [12].

Traditional works in the literature focus on discrete and Gaussian based HMMs [10]. Nonetheless, better modeling of state emission probabilities dependent on the nature of the data is an important parameter that has been recently tackled [46]. Indeed, strictly positive data that sum up to one; i.e., proportional data, are one such significant category. These time series data naturally occur from various multiple preprocessing procedures, including the famous histograms, across the spectrum of pattern recognition applications. In this paper, we employ the Beta-Liouville (BL) distribution for proportional sequential data modeling [88].

While it is common to employ a Gaussian-based HMM in all instances, it is not the best practice to do so with proportional data [75]. In particular, the characters of the Gaussian distribution lead to a sub-optimal modeling. The latter include its unbounded infinite support and symmetry properties. As such, utilizing emission probability distributions fit to the data has shown better performance in recent research. These include HMMs for proportional data [29, 89, 30, 90] and Student's t data [91].

We also tackle the learning problem of HMMs in terms of utilizing a variational learning approach for the training process [92]. Usually, a variation of the Expectation Maximization method is used which is known as the Baum-Welch algorithm. This technique suffers from many disadvantages that the variational approach is capable of alleviating. These include inconsideration of prior knowledge into the training process and under-fitting as well as over-fitting.

All in all, this chapter expands on our recent findings in [46] where we proposed the first mathematical model for the variational learning of BL HMM. In particular, we apply the BL HMM to another IR AR dataset to examine its generalization capabilities. We also perform the first evaluation of the BL HMM in visible AR and in multimodal fusion for AR to the best of our knowledge as well as detail the algorithm execution steps.

The rest of this chapter is organized as follows. Section 3.2 details the proposed model. Section 3.3 discusses the experimental setup and results.

3.2 Variational Learning of the Beta-Liouville Hidden Markov Model

In this section, we examine the mathematical derivations of the HMM proposed for better modeling of proportional data. A HMM is generally characterized by an underlying stochastic process with K hidden states. These form a Markov chain. An initial probability π governs each of the states with a transition matrix $B = \{b_{ii'} = P(s_t = i'|s_{t-1} = i)\}$ between the states at time t. An observation is emitted corresponding to its distribution in each state s_t. This distribution may be discrete or continuous. It is also known as the observable stochastic process set.

$\Xi = \{\Xi_i(m) = P(O_t = \xi_m|s_t = i)\}$ denotes the emission matrix of the discrete observations where $[m, t, i] \in [1, M] \times [1, T] \times [1, K]$, and the set of all possible discrete observations $\Xi = \{\xi_1, ..., \xi_m, ..., \xi_M\}$. On the other hand, the respective parameters of a probability distribution define the observation emission for a continuous observed symbol sequence. This is usually modelled by the Gaussian distribution that is defined by its mean and covariance matrix $\varkappa = (\mu, \Sigma)$ [10, 14, 15]. Hence, a mixing matrix must be defined $C = \{c_{ij} = P(m_t = j|s_t = i)\}$ in the case of continuous HMM emission probability distribution where $j \in [1, M]$ such that M is the number of mixture components in set $L = \{m_1, ..., m_M\}$. Consequently, a discrete is defined by $\lambda = \{B, \Xi, \pi\}$. On the other hand, a continuous HMM may be defined with the following respective parameters $\{B, C, \varkappa, \pi\}$. In this paper, we consider the latter case which is defined as a proportional mixture model of BL distribution.

In D dimensions, a BL distribution is defined as:

$$BL(\vec{x}|\vec{\alpha}, \alpha, \beta) = \frac{\Gamma(\sum_{d=1}^{D} \alpha_d)\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \prod_{d=1}^{D} \frac{x_d^{\alpha_d - 1}}{\Gamma(\alpha_d)} \times \left(\sum_{d=1}^{D} x_d\right)^{\alpha - \sum_{d=1}^{D} \alpha_d} \left(1 - \sum_{d=1}^{D} x_d\right)^{\beta - 1} \tag{62}$$

where $\vec{\alpha} = (\alpha_1, ..., \alpha_D)$, α, and β are the real and strictly positive parameters of the BL distribution, $\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} dx$ is the Gamma function, and $\vec{x}$ is a D dimensional vector whereby $\vec{x} \in \mathbb{R}_+^D$ and $\sum_{d=1}^{D} x_d < 1$. For simplification, we also denote $\Lambda = [\vec{\alpha}, \alpha, \beta]$; i.e., the parameters of the BL distribution. These parameters once trained given the desired data result in a custom-fitted BL distribution that has a greater capability to represent the underlying distribution of the proportional

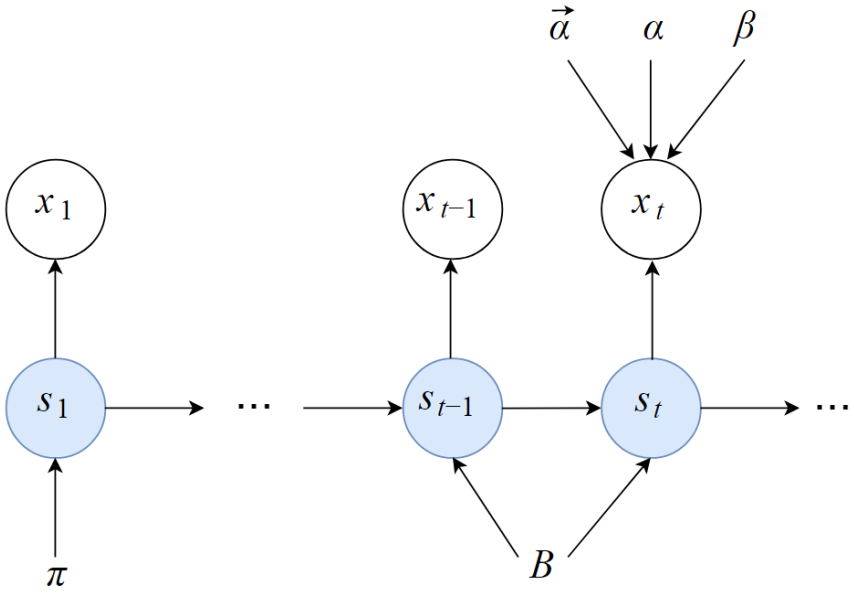


Figure 3.1: Graphical model representation of the Beta-Liouville based hidden Markov model. Symbols in unshaded circles denote the observed variables, symbols in shaded circles indicate the hidden states, and edges represent conditional dependencies between the states or the variables.

data at hand.

The likelihood of X, a time-series or sequence of observations of length T, given the model is expressed as:

$$p(X|B,C,\pi,\varkappa) = \sum_{S}\sum_{L} \pi_{s_1} \left[\prod_{t=2}^{T} b_{s_{t-1},s_t}\right] \times \left[\prod_{t=1}^{T} c_{s_t,m_t} p(x_t|\varkappa_{s_t,m_t})\right] \tag{63}$$

where $\varkappa_{ij} = (\varkappa_{1ij}, ..., \varkappa_{Dij})$ with $i \in [1, K]$ where K is the number of states in S; the set of hidden states, and $j \in [1, M]$ where M is the number of mixture components in L; the set of the components of the mixture. M is assumed to be uniform for all the states. Hence, the model is derived for a unique observation sequence for simplification purposes. To consider further observation sequences, an addition of a summation of these sequences would be logically required in the corresponding observation data equations. Furthermore, when $A > 1$, the parameter T is then dependent on each of the available time series observation sequences $\{X^a\}_{a=1,...,A}$ such that it would be denoted as T_a. It is also noteworthy to mention that such a setup is highly recommended since it prevents overfitting. A graphical depiction of the proposed HMM is shown in Fig. 3.1.

The exact computation of Equation (63) is intractable due to the need of summation over all possible combinations of mixture components and states. Consequently, the typical methodology for its solution constitutes of the maximization of the data likelihood with respect to the parameters of the model using the Baum-Welch algorithm [10]. Nonetheless, this approach suffers from several drawbacks. These include overfitting and absence of a convergence guarantee due to the general multimodal nature of the data likelihood function.

On the other hand, an estimation of the model may be derived using the variational Bayesian approach. This uses the posterior probabilities through the assignment of parameter priors for integrating out the marginal likelihood of the data. Hence, all the model parameters are regarded as random variables. The complete data likelihood is then denoted as:

$$p(X) = \int d\pi dB dC d\Lambda \sum_{S,L} p(B, C, \pi, \Lambda) p(X, S, L|B, C, \pi, \Lambda) \tag{64}$$

Equation (64) is still computationally intractable. This is due to the exponential growth of the number of possible sequences to be summed as the length of the time series increases [93]. However, an introduction of the approximate distribution $q(B, C, \pi, \Lambda, S, L)$ of the true posterior $p(B, C, \pi, \Lambda, S, L|X)$ enables us to derive a lower bound.

On the other hand, Jensen's inequality states that $\mathbf{E}[\mathcal{F}(x)] \geq \mathcal{F}(\mathbf{E}[x])$ where $\mathcal{F}(.)$ is a non-concave function and $\mathbf{E}[.]$ denotes the expectation. Thus, using Jensen's inequality and Equation (64), the lower bound can be expressed as:

$$\begin{aligned} \ln(p(X)) &= \ln \left\{ \int d\pi dB dC d\Lambda \sum_{S,L} p(B, C, \pi, \Lambda) p(X, S, L|B, C, \pi, \Lambda) \right\} \\ &\geq \int d\pi dB dC d\Lambda \sum_{S,L} q(B, C, \pi, \Lambda, S, L) \ln \left\{ \frac{p(B, C, \pi, \Lambda) p(X, S, L|B, C, \pi, \Lambda)}{q(B, C, \pi, \Lambda, S, L)} \right\} \end{aligned} \tag{65}$$

When q is equal the true posterior, the inequality is tight. Hence,

$$\ln(p(X)) = \mathcal{L}(q) - KL(q(B, C, \pi, \Lambda, S, L)||p(B, C, \pi, \Lambda, S, L|X)) \tag{66}$$

where $\mathcal{L}(q)$ is the lower bound and KL is the Kullback-Leibler distance between the true posterior

and the approximate distribution [92, 94].

The computation of the exact posterior distribution is intractable, so we only account for a certain family of distributions. As per the studied assumptions in [92, 94, 95, 93, 96], q may be factorized with the mean-field approximation; i.e., $q(B, C, \pi, \Lambda, S, L) = q(B)q(C)q(\pi)q(\Lambda)q(S, L)$ where $q(\Lambda) = q(\vec{\alpha})q(\alpha)q(\beta)$, with a similar factorization applying to p. $\mathcal{L}(q)$ can then be expressed as:

$$\begin{aligned}
\ln(p(X)) \geq \sum_{S,L} \int dBdCd\pi d\vec{\alpha}d\alpha d\beta q(B)q(C)q(\pi)q(\vec{\alpha})q(\alpha)q(\beta)q(S, L) \\
\{\ln(p(\pi)) + \ln(p(B)) + \ln(p(C)) + \ln(p(\vec{\alpha})) + \ln(p(\alpha)) + \ln(p(\beta)) + \\
\ln(p(\pi_{s_1})) + \sum_{t=2}^{T} \ln(b_{s_{t-1},s_t}) + \sum_{t=1}^{T} \ln(c_{s_t,m_t}) + \sum_{t=1}^{T} \ln(p(x_t|\vec{\alpha}_{s_t,m_t}, \alpha_{s_t,m_t}, \beta_{s_t,m_t})) \\
-\ln(q(S, L)) - \ln(q(\pi)) - \ln(q(B)) - \ln(q(C)) - \ln(q(\vec{\alpha})) - \ln(q(\alpha)) \\
-\ln(q(\beta))\} = F(q(\pi)) + F(q(B)) + F(q(C)) + F(q(\vec{\alpha})) + F(q(\alpha)) + F(q(\beta)) \\
+F(q(S, L))
\end{aligned} \tag{67}$$

In general, there are multiple maxima to the above lower bound; i.e., it is not convex. This suggests that the solution is dependent on the initialization. The priors of the parameters must then be defined to evaluate Equation (67). Since the coefficients of the parameters π, B, and C are all less than one, strictly positive, and with a sum result equal to one for each row summation, their priors are chosen as Dirichlet distributions as follows:

$$\begin{aligned}
p(\pi) = \mathcal{D}(\pi|\phi^{\pi}) = \mathcal{D}(\pi_1, ..., \pi_K|\phi^{\pi}_1, ..., \phi^{\pi}_K), \\
p(B) = \prod_{i=1}^{K} \mathcal{D}(b_{i_1}, ..., b_{i_K}|\phi^{B}_{i_1}, ..., \phi^{B}_{i_K}), \\
p(C) = \prod_{i=1}^{M} \mathcal{D}(c_{i_1}, ..., c_{i_M}|\phi^{C}_{i_1}, ..., \phi^{C}_{i_M})
\end{aligned} \tag{68}$$

This intuitively follows from their property to represent probabilities of the respective parameters that each of the symbols models.

Similarly, a conjugate prior must also be defined over the BL parameters $\vec{\alpha}$, α, and β. We

adopt the Gamma distribution $\mathcal{G}(.)$ for conjugate prior approximations of the latter parameters as previously proposed in [92]. This follows due to the strictly non-negative nature that defines these parameters. Consequently, we define the prior distributions as:

$$p(\{\vec{\alpha}\}_{i,j,l=1}^{K,M,D}) = \prod_{i=1}^{K}\prod_{j=1}^{M}\prod_{l=1}^{D}\mathcal{G}(\alpha_{ijl}|u_{ijl}, v_{ijl}), \tag{69}$$

$$p(\{\alpha\}_{i,j=1}^{K,M}) = \prod_{i=1}^{K}\prod_{j=1}^{M}\mathcal{G}(\alpha_{ij}|g_{ij}, h_{ij}), \tag{70}$$

$$p(\{\beta\}_{i,j=1}^{K,M}) = \prod_{i=1}^{K}\prod_{j=1}^{M}\mathcal{G}(\beta_{ij}|e_{ij}, r_{ij}) \tag{71}$$

where the hyperparameters u, g, h, e, r, and v are also strictly positive.

The iterative variational Bayesian inference process consists of two alternating steps; the E-step and the M-step. All of the parameters of the model are then learned through a sequential repetition of a M-step followed by an E-step until convergence. Hidden states and mixture components are updated in the M-step, so all (S, L) terms in Equation (67) are not considered. On the other hand, $q(S, L)$ is subsequently updated in the E-step; now keeping all other parameters fixed.

The following optimizations of $q(B)$, $q(C)$, and $q(\pi)$ are applicable to other continuous HMMs as they are independent of the emission distribution used. Therefore, these have already been studied in [95, 90]. As such, only the main equations are given and the reader is referred to the aforementioned references for further details. Indeed, these derivations are shared across all continuous based HMMs and intuitively connected to the model structure itself. Consequently, the derivation of the equations with terms pertaining only to the B parameter from Equation (67) gives:

$$F(q(B)) = \int q(B)\ln\left[\frac{\prod_{i=1}^{K}\prod_{j=1}^{K} b_{ij}^{\omega_{ij}^{B}-1}}{q(B)}\right] dB \tag{72}$$

with

$$\omega_{ij}^{B} = \sum_{t=2}^{T}\gamma_{ijt}^{B} + \phi_{ij}^{B} \tag{73}$$

and

$$\gamma_{ijt}^{B} \triangleq q(s_{t-1} = i, s_t = j) \tag{74}$$

where γ_{ijt}^{B} is a local probability typically computed with a forward-backward algorithm in a HMM framework [10]. To maximize $F(q(B))$, we apply the Gibbs inequality which results in:

$$q(B) = \prod_{i=1}^{K} \mathcal{D}(a_{i1}, ..., a_{iK} | \omega_{i1}^{B}, ..., \omega_{iK}^{B}) \tag{75}$$

Similarly for the π parameter:

$$q(\pi) = \mathcal{D}(\pi_1, ..., \pi_K | \omega_1^{\pi}, ..., \omega_K^{\pi}) \tag{76}$$

with

$$\omega_i^{\pi} = \gamma_i^{\pi} + \phi_i^{\pi} \tag{77}$$

and

$$\gamma_i^{\pi} \triangleq q(s_1 = i) \tag{78}$$

Finally, for the C parameter:

$$q(C) = \prod_{i=1}^{K} \mathcal{D}(c_{i1}, ..., c_{iM} | \omega_{i1}^{C}, ..., \omega_{iM}^{C}) \tag{79}$$

with

$$\omega_{ij}^{C} = \sum_{t=1}^{T} \gamma_{ijt}^{C} + \phi_{ij}^{C} \tag{80}$$

and

$$\gamma_{ijt}^{C} \triangleq q(s_t = i, m_t = j) \tag{81}$$

Next, we tackle the optimization of $F(q(\Lambda))$. From Equation (67), we obtain:

$$F(q(\Lambda)) = \int q(\Lambda) \ln \left\{ \frac{\prod_{i=1}^{K} \prod_{j=1}^{M} p(\Lambda_{ij}) \prod_{t=1}^{T} p(x_t \Lambda_{ij})^{\gamma_{ijt}^{C}}}{q(\Lambda)} \right\} d\Lambda \tag{82}$$

In order to achieve tractability, we apply the previously discussed factorial approximation of $q(\Lambda)$ as in [12]. We note that the solution thus far is presented corresponding to that of a finite BL mixture model as investigated in [88]. This leads to the following evaluations:

$$q(\vec{\alpha}) = \prod_{l=1}^{D} \prod_{i=1}^{K} \prod_{j=1}^{M} \mathcal{G}(\alpha_{ijl} | u_{ijl}^{*}, v_{ijl}^{*}) \tag{83}$$

$$q(\alpha) = \prod_{i=1}^{K} \prod_{j=1}^{M} \mathcal{G}(\alpha_{ij} | g_{ij}^{*}, h_{ij}^{*}) \tag{84}$$

$$q(\beta) = \prod_{i=1}^{K} \prod_{j=1}^{M} \mathcal{G}(\beta_{ij} | e_{ij}^{*}, r_{ij}^{*}) \tag{85}$$

where

$$\begin{aligned} u_{ijl}^{*} = u_{ijl} + \sum_{p=1}^{P} \langle Z_{pij} \rangle \bar{\alpha}_{ijl} \Bigg[\Psi \left(\sum_{d=1}^{D} \bar{\alpha}_{ijd} \right) - \Psi(\bar{\alpha}_{ijl}) + \\ \sum_{d=1, d \neq l}^{D} \Psi' \left(\sum_{d=1}^{D} \bar{\alpha}_{ijd} \right) \bar{\alpha}_{ijd} (\langle \ln(\alpha_{ijd}) \rangle - \ln(\bar{\alpha}_{ijd})) \Bigg] \end{aligned} \tag{86}$$

$$v_{ijl}^{*} = v_{ijl} - \sum_{p=1}^{P} \langle Z_{pij} \rangle \left[\ln(X_{pl}) - \ln \left(\sum_{d=1}^{D} X_{pd} \right) \right] \tag{87}$$

$$g_{ij}^{*} = g_{ij} + \sum_{p=1}^{P} \langle Z_{pij} \rangle [\Psi(\bar{\alpha}_{ij} + \bar{\beta}_{ij}) - \Psi(\bar{\alpha}_{ij}) + \bar{\beta}_{ij} \Psi'(\bar{\alpha}_{ij} + \bar{\beta}_{ij})(\langle \ln(\beta_{ij}) \rangle - \ln(\bar{\beta}_{ij}))] \bar{\alpha}_{ij} \tag{88}$$

$$h_{ij}^{*} = h_{ij} - \sum_{p=1}^{P} \langle Z_{pij} \rangle \ln \left(\sum_{d=1}^{D} X_{pd} \right) \tag{89}$$

$$e_{ij}^{*} = e_{ij} + \sum_{p=1}^{P} \langle Z_{pij} \rangle [\Psi(\bar{\alpha}_{ij} + \bar{\beta}_{ij}) - \Psi(\bar{\beta}_{ij}) + \bar{\alpha}_{ij} \Psi'(\bar{\alpha}_{ij} + \bar{\beta}_{ij})(\langle \ln(\alpha_{ij}) \rangle - \ln(\bar{\alpha}_{ij}))] \bar{\beta}_{ij} \tag{90}$$

$$r_{ij}^{*} = r_{ij} - \sum_{p=1}^{P} \langle Z_{pij} \rangle \ln \left(1 - \sum_{d=1}^{D} X_{pd} \right) \tag{91}$$

with i and j fixed for P observation vectors where $l \in [1, D]$, $i \in [1, K]$, and $j \in [1, M]$. $\Psi(.)$ is the digamma function, and $\Psi'(.)$ is the trigamma function; the logarithmic first and second derivatives of the Gamma function respectively. The $*$ superscript implies the optimization of each of the corresponding parameters that the symbol is presented upon and $\langle . \rangle$ denotes the expectation with respect to the optimized parameter, accordingly. Moreover, $Z_{pij} = 1$ if X_{pt} belongs to state i and mixture component j and $Z_{pij} = 0$ otherwise; i.e., it is an indicator function. Then, the weights of the data samples with respect to each mixture component are defined within the HMM framework. These are also known as the responsibilities. Consequently, $\langle Z_{pij} \rangle = \sum_{t=1}^{T} \gamma_{pijt}^{C} = p(s = i, m = j|X)$ and the responsibilities are computed via the forward-backward algorithm [10]. The definitions of the expected values of the parameters in the aforementioned equations are as

follows:

$$\bar{\alpha}_{ijl} = \frac{u^*_{ijl}}{v^*_{ijl}}, \bar{\alpha}_{ij} = \frac{g^*_{ij}}{h^*_{ij}}, \bar{\beta}_{ij} = \frac{e^*_{ij}}{r^*_{ij}} \tag{92}$$

$$\langle \ln(\alpha_{ijl}) \rangle = \Psi(u^*_{ijl}) - \ln(v^*_{ijl}) \tag{93}$$

$$\langle \ln(\alpha_{ij}) \rangle = \Psi(g^*_{ij}) - \ln(h^*_{ij}) \tag{94}$$

$$\langle \ln(\beta_{ij}) \rangle = \Psi(e^*_{ij}) - \ln(r^*_{ij}) \tag{95}$$

This concludes the M-step of the algorithm. $q(S, L)$ is then estimated in the E-step with the previously evaluated parameters now fixed. Equation (67) can be rearranged as studied in [90] to:

$$\mathcal{L}(q) = F(q(S, L)) - KL(q(B, C, \pi, \Lambda)||p(B, C, \pi, \Lambda)) \tag{96}$$

where

$$\begin{aligned} F(q(S,L)) = &\sum_S q(S) \int q(\pi) \ln(\pi_{s1}) d\pi + \sum_S q(S) \int q(B) \sum_{t=2}^{T} \ln(b_{s_{t-1},s_t}) dB \\ &+ \sum_{S,L} q(S,L) \int q(C) \sum_{t=1}^{T} \ln(c_{s_t,m_t}) dC + \sum_{S,L} q(S,L) \\ &\int q(\Lambda) \sum_{t=1}^{T} \ln(p(x_t|\vec{\alpha}_{s_t,m_t}, \alpha_{s_t,m_t}, \beta_{s_t,m_t})) d\Lambda - \sum_{S,L} q(S,L) \ln(q(S,L)) \end{aligned} \tag{97}$$

and we naturally define:

$$\begin{aligned}
\pi_i^* &\triangleq \exp\left[\langle \ln(\pi_i)\rangle_{q(\pi)}\right], \\
\pi_i^* &= \exp\left[\Psi(\omega_i^\pi) - \Psi(\sum_i \omega_i^\pi)\right], \\
\\
b_{jj'}^* &\triangleq \exp\left[\langle \ln(b_{jj'})\rangle_{q(B)}\right], \\
b_{jj'}^* &= \exp\left[\Psi(\omega_{jj'}^B) - \Psi(\sum_{j'} \omega_{jj'}^B)\right], \\
\\
c_{ij}^* &\triangleq \exp\left[\langle \ln(c_{ij})\rangle_{q(C)}\right], \\
c_{ij}^* &= \exp\left[\Psi(\omega_{ij}^C) - \Psi(\sum_j \omega_{ij}^C)\right]
\end{aligned} \tag{98}$$

The final optimization that needs to be performed is:

$$\ln(p^*(X_t|\vec{\alpha}_{s_t,m_t}, \alpha_{s_t,m_t}, \beta_{s_t,m_t})) = \int q(\Lambda)\ln(p(X_t|\vec{\alpha}_{s_t,m_t}, \alpha_{s_t,m_t}, \beta_{s_t,m_t}))d\Lambda, \tag{99}$$

where

$$\begin{aligned}
p(X_t|\vec{\alpha}_{s_t,m_t}, \alpha_{s_t,m_t}, \beta_{s_t,m_t}) = \Bigg[& \frac{\Gamma(\sum_{d=1}^D \alpha_{ijd})\Gamma(\alpha_{ij}+\beta_{ij})}{\Gamma(\alpha_{ij})\Gamma(\beta_{ij})} \prod_{d=1}^D \frac{X_{td}^{\alpha_{ijd}-1}}{\Gamma(\alpha_{ijd})} \\
& \times \left(\sum_{d=1}^D X_{td}\right)^{\alpha_{ij}-\sum_{d=1}^D \alpha_{ijd}} \left(1 - \sum_{d=1}^D X_{td}\right)^{\beta_{ij}-1} \Bigg]^{\gamma_{ijt}^C}
\end{aligned} \tag{100}$$

We then substitute Equation (100) in Equation (99) and breakdown the distribution $BL(\vec{x}|\vec{\alpha}, \alpha, \beta)$ to a product decomposition corresponding to the prior factorization assumption made to $q(\Lambda)$. This

yields the following evaluation:

$$\begin{aligned}\ln(p^*(X_t|\vec{\alpha}_{s_t,m_t},\alpha_{s_t,m_t},\beta_{s_t,m_t})) &= \gamma_{ijt}^C \int q(\vec{\alpha})q(\alpha,\beta) \\ &\ln(\nu(X_t|\vec{\alpha}_{s_t,m_t})\eta(X_t|\alpha_{s_t,m_t},\beta_{s_t,m_t}))d\vec{\alpha}d\alpha d\beta \\ &= \gamma_{ijt}^C \left(\langle \ln(\nu(X_t|\vec{\alpha}))\rangle_{q(\vec{\alpha})} + \langle \ln(\eta(X_t|\alpha,\beta))\rangle_{q(\alpha,\beta)}\right) \quad (101)\end{aligned}$$

where

$$\begin{aligned}\langle \ln(\nu(X_t|\vec{\alpha}))\rangle_{q(\vec{\alpha})} &= \left\langle \ln\left(\frac{\Gamma(\sum_{d=1}^D \alpha_{ijd})}{\prod_{d=1}^D \Gamma(\alpha_{ijd})}\right)\right\rangle_{q(\vec{\alpha})} + \\ \sum_{d=1}^D \ln(X_{td})\langle \alpha_{ijd}-1\rangle_{q(\vec{\alpha})} &- \ln\left(\sum_{d=1}^D X_{td}\right)\sum_{d=1}^D \langle \alpha_{ijd}\rangle_{q(\vec{\alpha})} \\ &= J(\alpha_{ijl}) + \sum_{d=1}^D \ln(X_{td})\left(\frac{u_{ijd}}{v_{ijd}} - 1\right) \\ &\quad -\ln\left(\sum_{d=1}^D X_{td}\right)\sum_{d=1}^D \left(\frac{u_{ijd}}{v_{ijd}}\right) \quad (102)\end{aligned}$$

and

$$\begin{aligned}\langle \ln(\eta(X_t|\alpha_{ij},\beta_{ij}))\rangle_{q(\alpha_{ij},\beta_{ij})} &= \left\langle \ln\left(\frac{\Gamma(\alpha_{ij}+\beta_{ij})}{\Gamma(\alpha_{ij})\Gamma(\beta_{ij})}\right)\right\rangle_{q(\alpha,\beta)} \\ &+\ln\left(\sum_{d=1}^D X_{td}\right)\langle \alpha_{ij}\rangle_{q(\alpha,\beta)} + \ln\left(1-\sum_{d=1}^D X_{td}\right) \\ \times\langle \beta_{ij}-1\rangle_{q(\alpha,\beta)} &= J(\alpha_{ij},\beta_{ij}) + \ln\left(\sum_{d=1}^D X_{td}\right)\left(\frac{g_{ij}}{h_{ij}}\right) \\ &+\ln\left(1-\sum_{d=1}^D X_{td}\right)\left(\frac{e_{ij}}{r_{ij}} - 1\right) \quad (103)\end{aligned}$$

$J(\alpha_{ijl})$ and $J(\alpha_{ij},\beta_{ij})$ are analytically intractable. Consequently, they are approximated by

their lower bounds as derived in [92]. Using the second order Taylor approximation method, $J(\alpha_{ijl})$ and $J(\alpha_{ij}, \beta_{ij})$ are then denoted as follows:

$$J(\alpha_{ijl}) \geq \ln\left(\frac{\Gamma(\sum_{d=1}^{D}\bar{\alpha}_{ijd})}{\prod_{d=1}^{D}\Gamma(\bar{\alpha}_{ijd})}\right) + \sum_{d=1}^{D}\bar{\alpha}_{ijd}\left[\Psi\left(\sum_{l=1}^{D}\bar{\alpha}_{ijl}\right) - \right.$$
$$\Psi(\bar{\alpha}_{ijd})\left]\left[\langle\ln(\alpha_{ijd})\rangle - \ln(\bar{\alpha}_{ijd})\right] + \frac{1}{2}\sum_{d=1}^{D}\bar{\alpha}_{ijd}^{2}\left[\Psi'\left(\sum_{l=1}^{D}\bar{\alpha}_{ijl}\right)\right.\right.$$
$$\left.\left.-\Psi'(\bar{\alpha}_{ijd})\right]\left\langle(\ln(\alpha_{ijd}) - \ln(\bar{\alpha}_{ijd}))^{2}\right\rangle + \frac{1}{2}\sum_{d=1}^{D}\sum_{l=1,l\neq d}^{D}\bar{\alpha}_{ijd}\bar{\alpha}_{ijl}\times$$
$$\Psi'\left(\sum_{y=1}^{D}\bar{\alpha}_{ijy}\right)\left(\langle\ln(\alpha_{ijd})\rangle - \ln(\bar{\alpha}_{ijd})\right)\left(\langle\ln(\alpha_{ijl})\rangle - \ln(\bar{\alpha}_{ijl})\right) \quad (104)$$

$$J(\alpha_{ij}, \beta_{ij}) \geq \ln\left(\frac{\Gamma(\bar{\alpha}_{ij} + \bar{\beta}_{ij})}{\Gamma(\bar{\alpha}_{ij})\Gamma(\bar{\beta}_{ij})}\right) + \bar{\alpha}_{ij}(\Psi(\bar{\alpha}_{ij} + \bar{\beta}_{ij})$$
$$- \Psi(\bar{\alpha}_{ij}))(\langle\ln(\alpha_{ij})\rangle - \ln(\bar{\alpha}_{ij})) + \bar{\beta}_{ij}(\Psi(\bar{\alpha}_{ij} + \bar{\beta}_{ij}) - \Psi(\bar{\beta}_{ij}))$$
$$(\langle\ln(\beta_{ij})\rangle - \ln(\bar{\beta}_{ij})) + \frac{1}{2}\bar{\alpha}_{ij}^{2}(\Psi'(\bar{\alpha}_{ij} + \bar{\beta}_{ij}) - \Psi'(\bar{\alpha}_{ij}))$$
$$\langle(\ln(\alpha_{ij}) - \ln(\bar{\alpha}_{ij}))^{2}\rangle + \frac{1}{2}\bar{\beta}_{ij}^{2}(\Psi'(\bar{\alpha}_{ij} + \bar{\beta}_{ij}) - \Psi'(\bar{\beta}_{ij}))$$
$$\langle(\ln(\beta_{ij}) - \ln(\bar{\beta}_{ij}))^{2}\rangle + \bar{\alpha}_{ij}\bar{\beta}_{ij}\Psi'(\bar{\alpha}_{ij}$$
$$+ \bar{\beta}_{ij})(\langle\ln(\alpha_{ij})\rangle - \ln(\bar{\alpha}_{ij}))(\langle\ln(\beta_{ij})\rangle - \ln(\bar{\beta}_{ij})) \quad (105)$$

where $\langle(\ln(\alpha_{ijd}) - \ln(\bar{\alpha}_{ijd}))^{2}\rangle = (\Psi(u_{ijd}) - \ln(u_{ijd}))^{2} + \Psi'(u_{ijd})$, $\langle(\ln(\alpha_{ij}) - \ln(\bar{\alpha}_{ij}))^{2}\rangle = (\Psi(g_{ij}) - \ln(g_{ij}))^{2} + \Psi'(g_{ij})$, and $\langle(\ln(\beta_{ij}) - \ln(\bar{\beta}_{ij}))^{2}\rangle = (\Psi(e_{ij}) - \ln(e_{ij}))^{2} + \Psi'(e_{ij})$ as derived in [97].

Finally, by substituting Equation (104) into Equation (242), Equation (105) into Equation (103),

and Equation (98) into Equation (97), we yield:

$$F(q(S,L)) = \sum_{S,L} q(S,L) \ln \left(p^*(X_t|\vec{\alpha}_{s_t,m_t}, \alpha_{s_t,m_t}, \beta_{s_t,m_t}) \left(\frac{\pi^*_{s_1} \prod_{t=2}^{T} b^*_{s_{t-1},s_t} \prod_{t=1}^{T} c^*_{s_t,m_t}}{q(S,L)} \right) \right) \tag{106}$$

whereby the optimized $q(S,L)$ can then be denoted as:

$$q(S,L) = \frac{1}{W} \pi^*_{s_1} \prod_{t=2}^{T} b^*_{s_{t-1},s_t} \times \prod_{t=1}^{T} c^*_{s_t,m_t} p^*(X_t|\vec{\alpha}_{s_t,m_t}, \alpha_{s_t,m_t}, \beta_{s_t,m_t}) \tag{107}$$

where W is a normalizing constant and represents the likelihood of the optimized HMM which can be computed with a forward-backward algorithm [10]. This is defined as:

$$W = \sum_{S,L} \pi^*_{s_1} \prod_{t=2}^{T} b^*_{s_{t-1},s_t} \prod_{t=1}^{T} c^*_{s_t,m_t} \times p^*(X_t|\vec{\alpha}_{s_t,m_t}, \alpha_{s_t,m_t}, \beta_{s_t,m_t}) \tag{108}$$

Considering that Θ represents the approximated initialization of the BL parameters using the method of moments approach as previously studied by Epaillard and Bouguila in [90], the proposed algorithm may be summarized as follows:

- input: $X, \Theta, M, K, tol, maxIter$

  (1) Initialize hyperparameters:

  - $\phi^{\pi} = [1/K, ..., 1/K]$, where $len(\phi^{\pi}) = K$
  - $\phi^{B} = [1/K, ..., 1/K]$, where $len(\phi^{B}) = K$
  - $\phi^{C} = [1/M, ..., 1/M]$, where $len(\phi^{C}) = M$
  - $v_{ijl} = 1, \forall i, j, l$
  - $u_{ijl} = \zeta_{init}, \forall i, j, l$
  - $h_{ij} = 1, \forall i, j$

- $g_{ij} = \kappa_{init}, \forall i, j$
- $r_{ij} = 1, \forall i, j$
- $e_{ij} = \theta_{init}, \forall i, j$

(2) Initialize HMM parameters:

- Draw initial responsibilities γ^{π}, γ^{B}, and γ^{C} from prior distributions with Equation (68)
- Compute ω^{π}, ω^{B}, and ω^{C} with Equations (77), (73), and (80)
- Initialize π, B, and C with the computed quantities from Equation (98)

(3) Initialize iteration count and HMM likelihood:

- $iter = 0; lik^{old} = 10^{6}; lik^{new} = 10^{5}$

(4) ***While*** $|lik^{old} - lik^{new}| \geq tol \;\&\; iter \leq maxIter$

- **E-Step**
  - Compute the data likelihood lik^{data} with X, u, v, g, h, e, r, and Θ with Equations 62 and 107
  - Compute responsibilities γ^{π}, γ^{B}, and γ^{C} using the forward-backward algorithm with lik^{data}, π, B and C where the latter three quantities are computed with Equations 78, 74, and 81 respectively
  - Update u, v, g, h, e, and r with Equations 86, 87, 88, 89, 90, and 91
- **M-Step**
  - Update ω^{π}, ω^{B}, and ω^{C} with the corresponding responsibilities γ^{π}, γ^{B}, and γ^{C} using Equations 77, 73, and 80
  - Update B, C, and π using the computed ω^{π}, ω^{B}, and ω^{C} in Equation (98)
- **Update conditions to check convergence**
  - $lik^{old} \leftarrow lik^{new}$
  - Compute lik^{new} with Equation (108) and the forward-backward algorithm
  - iter += 1

Figure 3.2: InfAR dataset's sample frames.

3.3 Experimental Results

In this section, we evaluate the proposed model on two challenging AR datasets: IR *InfAR* [98] and multispectral *IOSB* [99]. These datasets were chosen for their unique position to enable the evaluation of the proposed model with others in the field as well as investigate the multi spectral fusion aspect of the paper. In particular, the *InfAR* dataset is the most used dataset for the evaluation of machine learning models in the literature as will be shown by the many comparisons made in our results. The *IOSB* dataset, on the other hand, is made up of videos that were taken simultaneously in both spectra. We report our result in terms of confusion matrices, average precision (AP), and accuracy measures. AP is defined as $AP = 1/\varsigma \sum_{\varrho=1}^{\varsigma}(TP_\varrho/(TP_\varrho + FP_\varrho))$ where ς refers to the total number of classes, while the accuracy measure may be calculated using $(TP + TN)/(TP + TN + FP + FN)$ where TP, FP, TN, FN denote the number of true positives, false positives, true negatives, and false negatives respectively.

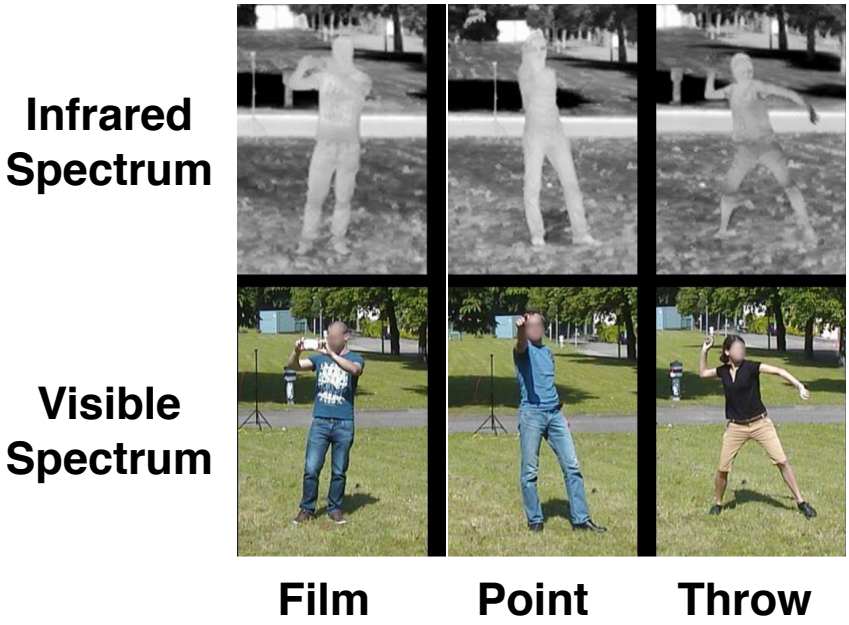


Figure 3.3: IOSB dataset's sample frames.

3.3.1 Datasets and Setup

InfAR IR Action Dataset

We choose 10 sample videos from each of the single class action classes in this dataset. In total, it contains twelve classes and six hundred videos. The resolution of the each of the videos is 293 $\times$ 256 with a frame rate of 25 for a four second duration. All in all, the training and testing pool used from this dataset is then seven classes and example frames from these classes can be observed in Fig. 3.2.

IOSB Multispectral Action Dataset

This dataset consists of visible and IR action videos that have been recorded at a sunny summer day of ten people; eight males and two females in the age range of 31.2$\pm$5.7 [99]. We test our proposed algorithm on three classes of the dataset; namely, film, point, and throw. Each of the classes has ten videos with sample frames shown in Fig. 3.3. Two Q1922 cameras with a spectral range of eight to fourteen μm; i.e., longwave IR, are used for generating IR data with a resolution of 640$\times$480 pixels with a frame rate of 25 frames per second. A similar frame rate of the captured

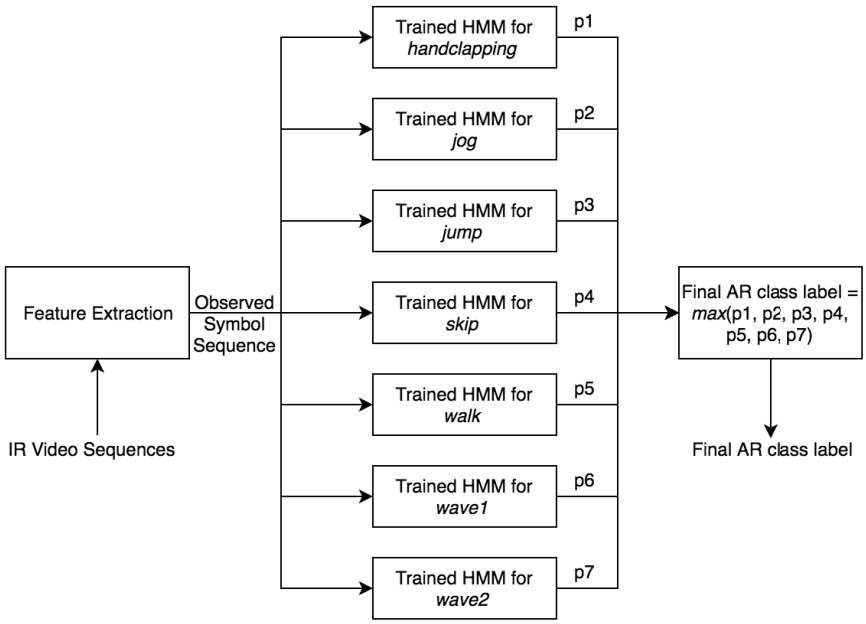


Figure 3.4: Infrared action recognition InfAR dataset classification experimental setup with the proposed trained hidden Markov models (HMM). The likelihoods of each of the trained HMMs are denoted by p1, p2, p3, p4, p5, p6, and p7, respectively.

actions in visible spectrum is generated by AXIS Q5534 and AXIS Q1755 cameras with a resolution of 800×600 pixels.

Experimental Setup

We choose histogram of optical flow (HOF) and motion boundary histogram (MBH) descriptors for representation of the AR videos. That is we extract a time series of these histograms for our experiments as features. This may be performed with an interest point detector [100]. For the latter, we choose extraction along the motion trajectory [101].

For evaluation, we utilize a leave-one-out cross validation scheme and train an HMM for each class independently. The likelihood of each of the testing video sequences is then calculated by each the respective seven trained HMMs to assign the class label appropriately. This corresponds to the maximum resultant likelihood. Fig. 3.4 shows the experimental setup with the number of states and BL mixture components set experimentally to 2; i.e., $K = 2$ and $M = 2$ respectively.

In order to ensure robustness of the pipeline and the results, each set of features is used nine times for the training of the utilized BL HMM for a total of 630 trained HMMs on the InfAR

Table 3.1: Comparison of the Average Precision (AP) of the proposed models.

Method	AP
Two stream 3D CNN [102]	75.42%
Optical flow field 3D CNN [102]	77.50%
Deep-convolutional descriptors [103]	79.25%
HOF [98]	68.58%
Dense trajectories [104]	68.66%
Improved dense trajectories [105]	71.83%
BL HMM (HOF) - InfAR	*78.41%*
BL HMM (Horizontal MBH) - InfAR	*89.57%*
BL HMM (Vertical MBH) - InfAR	*92.29%*
BL HMM (HOF) - IOSB	*58.05%*
BL HMM (Horizontal MBH) - IOSB	*81.53%*
BL HMM (Vertical MBH) - IOSB	*73.95%*

dataset. The average results are then reported across the various trained HMMs. For benchmarking the results, we perform experiments with a Gaussian-based HMM applying the same setup.

3.3.2 Unimodal Results

We achieve an accuracy of 77.94% when training with the HOF features extracted from the InfAR dataset compared with 42.86% with the Gaussian HMM. Moreover, the average accuracy of proposed model is 89.05% and 92.06% with the horizontal and vertical MBH features respectively versus 85.7% using the benchmark. As such, the proposed HMM clearly outperforms the benchmark and shows promising results. The confusion matrices of the different features with the proposed BL HMM with the InfAR dataset can be observed in Fig. 3.6, Fig. 3.7, and Fig. 3.8. Also, a compact representation of the comparison of the accuracy results can be observed in Fig. 3.5.

Moreover, generalization of the performance of the model is demonstrated by the results of the proposed model on the IOSB IR frames as shown in Fig. 3.9, Fig. 3.10, and Fig. 3.11. We carry out a similar setup for the IOSB dataset with the training setup limited to only two iterations. This results in a total of 60 trained HMMs for testing. Overall, our results are comparable to several other methods reported in the literature. A comparison of the achieved results with the AP of the proposed model can be observed in Table 3.1.

We also train our proposed HMM model on the IOSB visible spectrum frames. These results

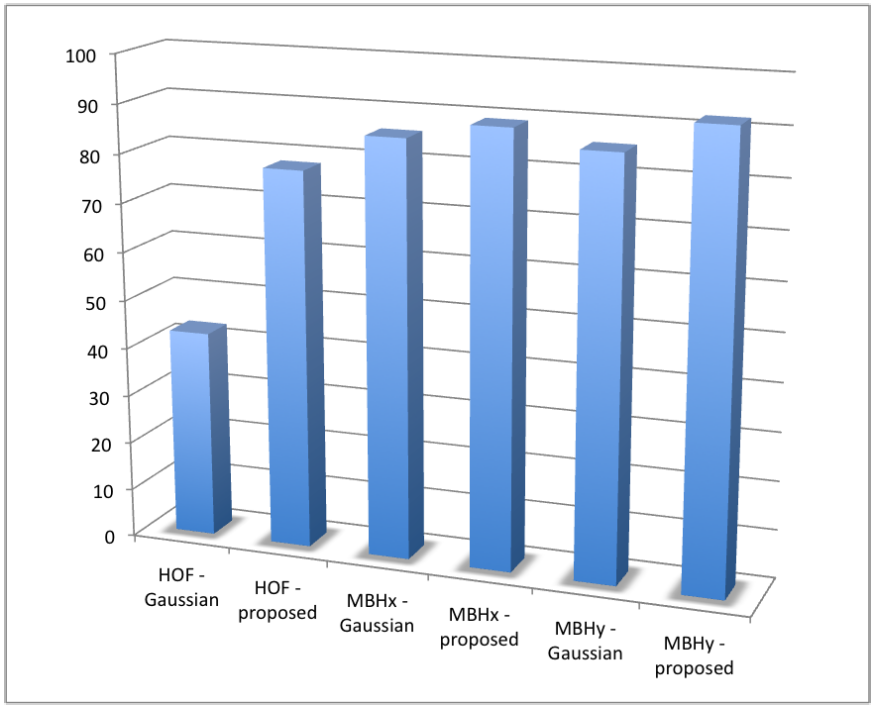


Figure 3.5: Comparison of the accuracy of the proposed HMM using the different extracted features against the benchmark in the literature; i.e., the Gaussian HMM.

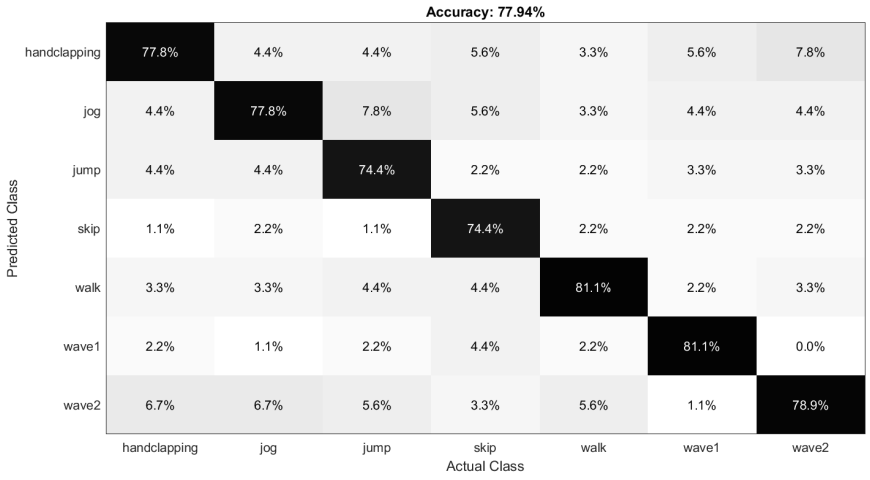


Figure 3.6: Confusion matrix for BL HMM trained with HOF features extracted from the InfAR dataset.

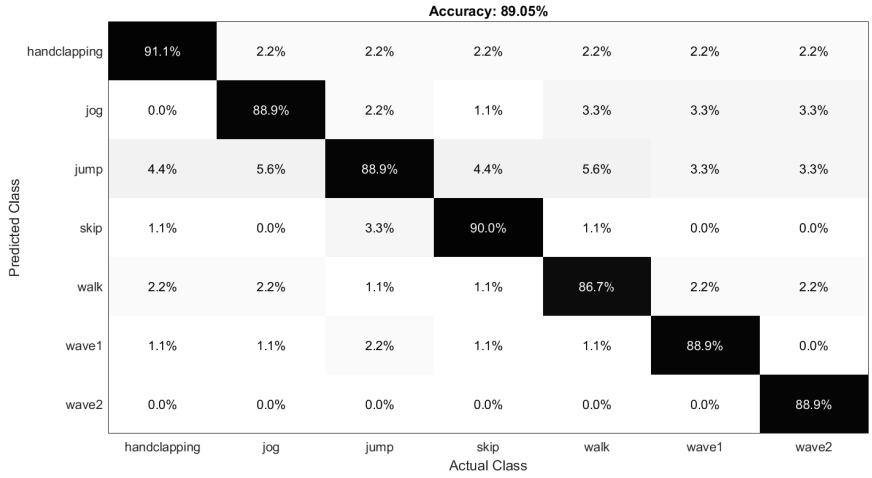


Figure 3.7: Confusion matrix for BL HMM trained with horizontal MBH features extracted from the InfAR dataset.

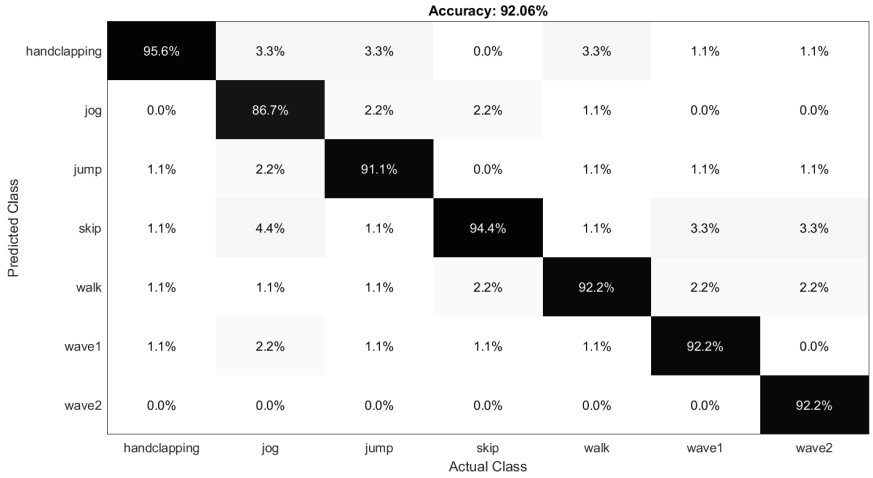


Figure 3.8: Confusion matrix for BL HMM trained with vertical MBH features extracted from the InfAR dataset.

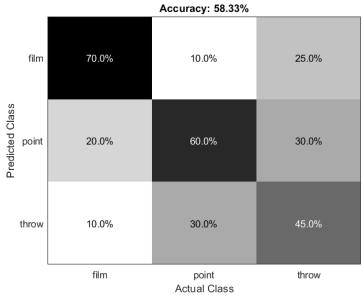


Figure 3.9: Confusion matrix for BL HMM trained with HOF features extracted from the IOSB IR frames.

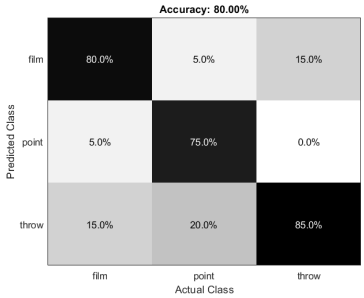


Figure 3.10: Confusion matrix for BL HMM trained with horizontal MBH features extracted from the IOSB IR frames.

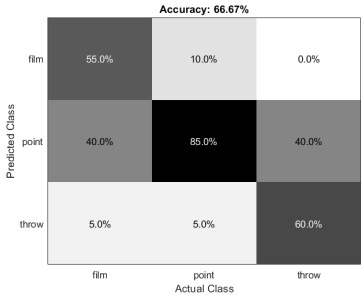


Figure 3.11: Confusion matrix for BL HMM trained with vertical MBH features extracted from the IOSB IR frames.

may be observed in Fig. 3.13 for the HOF features, Fig. 3.14 for the horizontal MBH features, and Fig. 3.15 for the vertical MBH features. Due to the characteristics of the visible spectrum that includes high sensitivity to shadow, background clutter, occlusion, and changes in illumination, the results of the proposed model is not as satisfactory as in the IR spectrum. Nonetheless, these investigations back up the importance of using the IR spectrum in AR as well as act as motivation for the multimodal fusion approach that we propose in Section 3.3.3. Also, a compact representation of the comparison of the accuracy and AP results can be observed in Fig. 3.12.

3.3.3 Multimodal Fusion

There are many classifier fusion methods such as fusing single class labels and the class ranking based techniques [106]. We choose a soft-output classifier fusion approach. Specifically, Bayesian fusion approaches that are based on retaining the posterior probabilities of each of the classifiers to fuse. For our paper, this translates to the likelihoods of each of the HMM classifiers for the different modalities with a prior that is set to be a uniform distribution. Hence, this has the effect of fusing

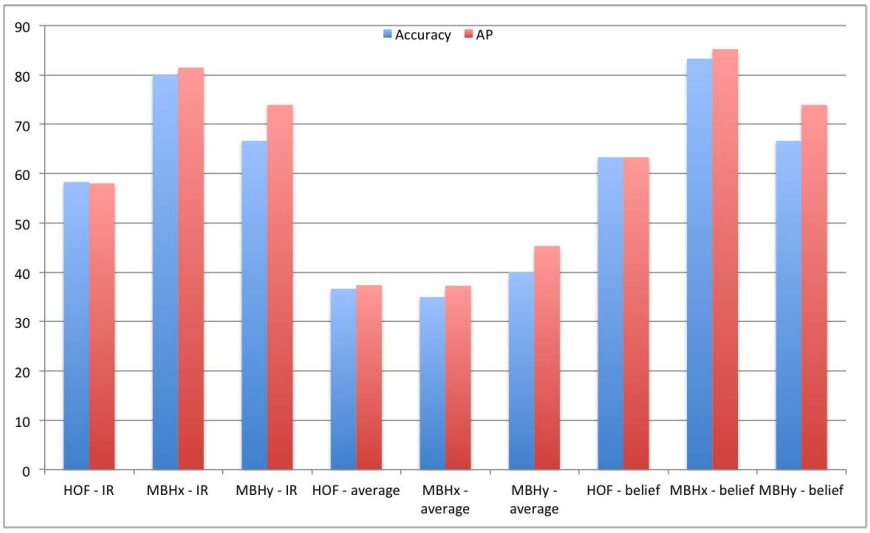


Figure 3.12: Comparison of the accuracy and AP of the proposed HMM on the different extracted features using the different the fusion methods against the IR unimodal results where *average* denotes the Average Bayes method and *belief* defines the Bayes Belief Integration method respectively.

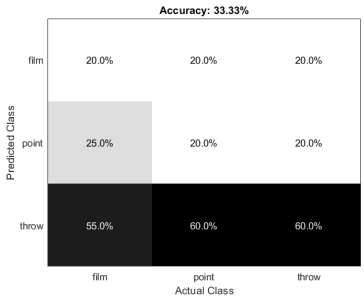


Figure 3.13: Confusion matrix for BL HMM trained with HOF features extracted from the IOSB visible spectrum frames.

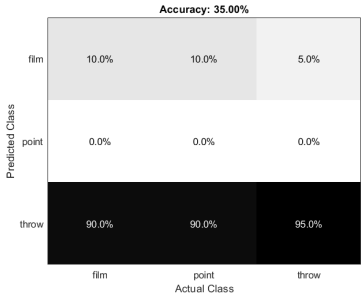


Figure 3.14: Confusion matrix for BL HMM trained with horizontal MBH features extracted from the IOSB visible spectrum frames.

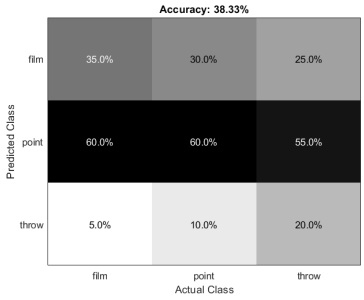


Figure 3.15: Confusion matrix for BL HMM trained with vertical MBH features extracted from the IOSB visible spectrum frames.

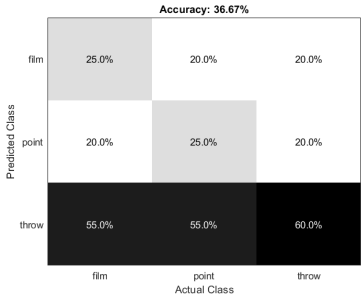


Figure 3.16: Confusion matrix for BL HMM trained with HOF features extracted from the IOSB visible spectrum and IR frames fused with the Average Bayes method.

the modalities using the IR and the visible likelihoods as appropriate. We investigate two methods of Bayesian approaches: the simple Average Bayes and the Bayes Belief Integration [106].

The Average Bayes method consists of finding the average of the posterior probabilities proportional to the likelihood results of the HMMs which we denote by $p\iota$ of the different classifiers $\iota = 1, \ldots, \upsilon$ following the notation used in Fig. 3.4. This Average Bayes classifier is then denoted by:

$$p_{avg} = \frac{1}{\upsilon} \sum_{\iota=1}^{\upsilon} p\iota \tag{109}$$

where υ is the total number of classifiers to be fused; i.e., $\upsilon = 2$ for the two modalities in our case.

The outcome of using this method may be observed in Fig. 3.16, Fig. 3.17, and Fig. 3.18 for the HOF, the horizontal MBH, and the vertical MBH features respectively. These correspond to AP values of 37.41%, 37.27%, and 45.34%. These poor performance results can be attributed to giving the same weight or importance to each of the multimodal classifiers which in effect disregards the better outturn of using the IR spectrum and leads to the degraded repercussion observed.

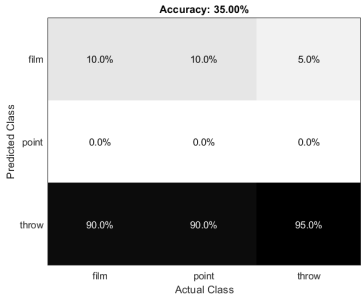


Figure 3.17: Confusion matrix for BL HMM trained with horizontal MBH features extracted from the IOSB visible spectrum and IR frames fused with the Average Bayes method.

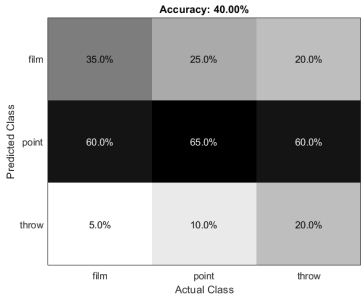


Figure 3.18: Confusion matrix for BL HMM trained with vertical MBH features extracted from the IOSB visible spectrum and IR frames fused with the Average Bayes method.

On the other hand, the Bayes Belief Integration method operates by the incorporation of the results of the confusion matrix of each of the classifiers. The final fusion result is then based on choosing the label output by the classifier with the higher belief measure. In effect, this belief measure is set by comparing the corresponding confusion matrix entries and selecting the classifier of the highest correct classification for each of the classes.

This fusion method yields the results shown in Fig. 3.19 and Fig. 3.20 for the HOF and the horizontal MBH features, while the confusion matrix for the vertical MBH features is the same as the IR spectrum which may be observed in Fig. 3.11. The latter is due to the superior performance of the IR HMM classifier. In other words, this methods settles on the best HMM for each of the classes for each of the modalities. Hence, the AP values for this final fusion are 63.33%, 85.24%, and 73.95% for the HOF, horizontal MBH, and vertical MBH features respectively. These results are the best achieved for the proposed model and proves the advantage of using multimodal fusion in AR.

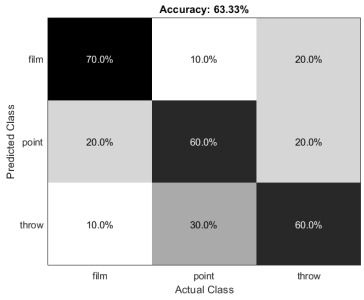


Figure 3.19: Confusion matrix for BL HMM trained with HOF features extracted from the IOSB visible spectrum and IR frames fused with the Bayes Belief Integration method.

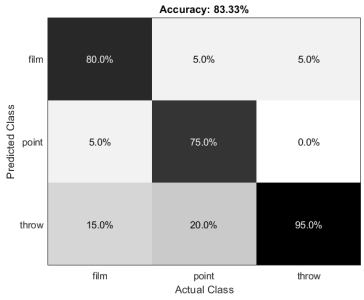


Figure 3.20: Confusion matrix for BL HMM trained with horizontal MBH features extracted from the IOSB visible spectrum and IR frames fused with the Bayes Belief Integration method.

3.3.4 Further Discussions

In this section, we discuss miscellaneous aspects of the results and their consequent implications. In particular, we address two facets of additional experimentation: time complexity and comparison with deep learning technique.

Time complexity refers to the time that is needed for the HMM to process a video sequence. We found this to be 9.85 seconds for the proposed model. The experiments were carried out on a machine with 32 GB RAM and 3.6 GHz processor and the code was written using the MATLAB software. We also note the time taken for the benchmark Gaussian-based HMM that follows an independence assumption. That is the diagonal of the co-variance matrix is the used as the modeling parameter along with the means. The testing time yielded was 0.08 seconds.

This difference in performance may be explained by the simplicity of the benchmark's parameters versus the proposed model. Moreover, the lower computational cost of the pre-optimized functions that are used for the computation of these parameters also contributes to this variation.

Nonetheless, the superior performance of the proposed model in comparison to the benchmark frames this as a trade-off relationship. Furthermore, this computational time needed may be reduced using a more powerful machine as well as by potentially implementing this code in C. However, this falls outside the scope of this paper and the time complexity details are only included for a thorough inspection of the results.

On the other hand, we also implement a convolutional neural network model which is a deep learning approach for action recognition. The network consists of 4 convolutional layers, 2 fully connected linear layers, and finally a max pooling layer. This setup is trained for 75 epochs for each of the datasets with learning rate of 0.001 and batch size of 32. This rendered a training accuracies of 88.83%, 98.6%, and 99.3% and validation accuracies of 97.22%, 100.0%, and 100.0% for the InfAR, visible IOSB, and IR IOSB datasets respectively. This shows overfitting phenomenon due to the high complexity of the model. On the other hand, the testing results achieved were suboptimal at best. Given a leave one out scheme the testing sequences had 66.7% for the IOSB datasets and 57.1% for the InfAR dataset. This demonstrates the flaw of deep learning techniques; i.e., discrimintative based machine learning, particularly in comparison to generative ones. The latter in this case represents HMMs; the branch under which the proposed method lies and which is less prone to suffer from overfitting whose case is clearly shown in this case. Moreover, this technique has features that are covert and hence renders in-explainable results. Moreover, the computational complexity of deep learning is much higher at 272,295 parameters for a relatively simple structure.

Chapter 4

Hybrid Generative Discriminative Approach with Hidden Markov Models and Support Vector Machines

> Make your life a masterpiece; imagine no limitations on what you can be, have, or do.
>
> Brian Tracy

Our investigations thus far have been purely generative in nature. Whereas variational inference has shown an undeniable improvement of the performance of proportional-based hidden Markov models, an interesting problem arises for discriminative models where the lengths of the sequential data of interest is not the same. Ergo, in this chapter, we delve into generative discriminative approaches and propose novel models as appropriate for the topic and the application of dynamic texture categorization.

4.1 Introduction

Dynamic textures (DT) are videos that constitute of complex dynamical objects such as sea waves and grass waving in the wind [107]. The DT generative model is of a particularly attractive research interest for its proven effectiveness in many domains such as video classification [108, 109,

110], video segmentation [111, 112, 113, 114], human action recognition [115], video synthesis [116, 114], and abnormal motion detection [117].

These objects may be modeled due to their stationary behavior in time [116]. As such, DT recognition falls under the broad theme of spatiotemporal modeling. This is intuitively inferred as the modeling of objects that involve dependencies between spatial and time dimensions. For instance, facial expression classification is another task where space and time are omnipresent and where DT has been impactful [118, 119]. Hidden Markov models (HMM) are key models for capturing this space time dependency [120].

A HMM is a double stochastic model that extracts underlying statistic through the employment of a compact set of features [3]. Generative models, such as HMMs, require less training data [87] than discrimintaive models. This defines the main two categories of machine learning algorithms. Generally, training discriminative models corresponds to inferring a mapping between data inputs to class labels, while generative models first learn the distribution of the classes before predictions are made [4]. As such, discriminative models usually achieve superior classification accuracy results [5]. These include the famous Support Vector Machines (SVM) [121, 122, 123].

SVMs are one of the most popular data modeling techniques due to its capability to construct a linear boundary in a projected space of the original data via a kernel which translates to a non-linear boundary in the original space. Motivation behind using SVMs is well-researched and an interested reader is referred to [124]. One of the main challenges of applying SVMs is determining the right kernel. Popular choices include the linear, polynomial, and radial basis function kernels [125]. However, these kernels may only be applied to features or data of the same length. This may not always be the case for extracted features from DTs.

In order to overcome this hindrance, we propose the use of Fisher Kernels (FK) generated with HMMs [126, 127] of Dirichlet [128, 94], Beta-Liouville (BL) [129, 46], and Generalized Dirichlet (GD) [30, 94] emission distributions. The Fisher kernel captures the intrinsic properties of the data resulting in data-driven kernels [130]. We then derive a more powerful model by combining both generative models with the SVM approach [131, 132].

In particular, the contributions of this chapter are fivefold: (i) we apply the BL HMM for DT recognition with variational learning; (ii) we perform the first evaluation of the variational-based

Dirichlet HMM on the same application; (iii) we derive a hybrid generative-discriminative approach for both Dirichlet and BL HMM with FK for SVM-based proportional data modeling and apply the model for DT recognition; (iv) we derive a hybrid generative-discriminative approach for the GD HMM with FK for SVM-based proportional data modelling; (v) we apply the proposed model for DT recognition.

4.2 Hybrid Generative-Discriminative Approach with Fisher Kernels

In this section, we discuss the main components of the proposed approach. Section 4.2.1 examines the HMM structure with the Dirichlet, BL, and GD parameters; Section 4.2.2 overviews the forward-backward algorithm; and Section 4.2.3 presents the FK derivations.

4.2.1 Hidden Markov Models for Proportional Data

A HMM is characterized by an underlying stochastic process with K hidden states, each governed by an initial probability π, and the transition between the states $B = \{b_{ii'} = P(s_t = i'|s_{t-1} = i)\}$ at time t. In each state s_t, an observation is emitted corresponding to its respective parameters of a probability distribution $\varkappa$ with a mixing matrix $C = \{c_{ij} = P(m_t = j|s_t = i)\}$ where $j \in [1, M]$ such that M is the number of mixture components in set $L = \{m_1, ..., m_M\}$. Hence, a continuous HMM may be defined as $\lambda = \{B, C, \varkappa, \pi\}$.

Formally, a D-dimensional Dirichlet distribution is denoted by:

$$DR(X|\vec{\varepsilon}) = \frac{\Gamma(\sum_{d=1}^{D} \varepsilon_d)}{\prod_{d=1}^{D} \Gamma(\varepsilon_d)} \prod_{d=1}^{D} x_d^{\varepsilon_d - 1} \tag{110}$$

where $\vec{\varepsilon} = (\varepsilon_1, ..., \varepsilon_D)$ is the real and strictly positive parameter of the distribution and $X \in \mathbb{R}_+^D$, $\sum_{d=1}^{D} x_d = 1$ corresponding to the D-dimension proportional vector that adds up to one.

A better model of proportional time series data is also proposed with the BL distribution [46]. This distribution is closely related to the Dirichlet, but it relaxes the constraint of negative covariance. This comes at the cost of two additional parameters. The BL distribution is denoted by:

$$BL(X|\vec{\delta},\alpha,\beta) = \frac{\Gamma(\sum_{d=1}^{D}\delta_d)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\prod_{d=1}^{D}\frac{x_d^{\delta_d-1}}{\Gamma(\delta_d)} \left(\sum_{d=1}^{D}x_d\right)^{\alpha-\sum_{d=1}^{D}\alpha_d}\left(1-\sum_{d=1}^{D}x_d\right)^{\beta-1} \tag{111}$$

where $\vec{\delta} = (\delta_1, ..., \delta_D)$, α, and β are real and strictly positive parameters of the BL distribution, $\Gamma(t) = \int_0^\infty X^{t-1}e^{-X}dX$ is the Gamma function, and X is a $D+1$ dimensional vector whereby $X \in \mathbb{R}_+^D$ and $\sum_{d=1}^{D} x_d < 1$. It is noteworthy to mention that the Dirichlet distribution is a special case of the BL distribution. Interested readers are referred to [94, 46] respectively for the detailed derivations of the Dirichlet and the BL HMMs with variational learning.

$\varkappa$ may also be defined according to the GD distribution for proportional data denoted by:

$$GD(X|\vec{\iota},\vec{\vartheta}) = \prod_{d=1}^{D}\frac{\Gamma(\iota_d+\vartheta_d)}{\Gamma(\iota_d)\Gamma(\vartheta_d)}x_d^{\iota_d-1}\left(1-\sum_{r=1}^{d}x_r\right)^{\zeta_d} \tag{112}$$

where $\vec{\iota} = (\iota_1, ..., \iota_D), \vec{\vartheta} = (\vartheta_1, ..., \vartheta_D)$ are the real and strictly positive parameters of the GD distribution and $X \in \mathbb{R}_+^D$ and $\sum_{d=1}^{D} x_d < 1$ corresponding to the $(D+1)$-dimensional proportional vector that adds up to one. Finally, ζ_d is computed using the parameters of the distribution as $\vartheta_d - \iota_{d+1} - \vartheta_{d+1}$, when $d \neq D$. Otherwise, $\zeta_d = \vartheta_D - 1$.

4.2.2 Forward-Backward Algorithm

The forward algorithm calculates the probability of being in state s_i at time t after the corresponding partial observation sequence given the HMM model λ. This defines the forward variable $\rho_t(i) = P(X_1, X_2, ..., X_t, i_t = s_i|\lambda)$ which is solved recursively as follows:

(1) initiate the forward probabilities with the joint probability of state s_i and the initial observation X_1: $\rho_1(i) = \pi_i\varkappa_i(X_1), 1 \leqslant i \leqslant K$;

(2) calculate how state $q_{i'}$ is reached at time $t+1$ from the K possible states s_i, $i = 1, 2, ..., K$ at time t and sum the product over all the K possible states: $\rho_{t+1}(i') = \left[\sum_{i=1}^{K}\rho_t(i)b_{ii'}\right]\varkappa_{i'}(X_{t+1})$

for $t = 1, 2, ..., T-1$, $1 \leqslant i' \leqslant K$

(3) Finally, compute $P(X|\lambda) = \sum_{i=1}^{K} \rho_T(i)$.

The forward algorithm has a computational complexity of K^2T which is considerably less than a naive direct calculation approach. Similar to the forward algorithm, but for computing the tail probability of the partial observation from $t+1$ to the end of an observation sequence, given that we are starting at state s_i at time t and model λ, is the backward algorithm. This has the variable $\theta_t(i) = P(X_{t+1}, X_{t+2}, ..., X_T, i_t = s_i|\lambda)$ and is solved as follows:

(1) Compute an arbitrary initialization $\theta_T(i) = 1$, $1 \leqslant i \leqslant K$;

(2) $\theta_t(i) = \sum_{i'=1}^{K} b_{ii'} \varkappa_{i'}(X_{t+1}) \theta_{t+1}(i')$ for $t = T-1, T-2, ..., 1$, $1 \leqslant i \leqslant K$

Together with the forward algorithm, this forms the forward-backward algorithm through consequent iteration that is used for the calculation of the probability of a DT sequence X given λ:

$$P(X|\lambda) = \sum_{i=1}^{K} \sum_{i'=1}^{K} \rho_t(i) b_{ii'} \varkappa_{i'}(X_{t+1}) \theta_{t+1}(i') \tag{113}$$

4.2.3 Fisher Kernels

Kernels project features into higher dimensional space with a function $\kappa(y_\varsigma, y_\varphi) = \langle \phi(y_\varsigma), \phi(y_\varphi) \rangle$ where y_ς and y_φ are observations not necessarily of the same length, ϕ is a projection function chosen as FK in this paper, and $\langle ., . \rangle$ implies the inner product. FK is generally used for building hybrid generative-discriminative models for classification and is denoted by:

$$FK(X_\varsigma, X_\varphi) = \langle FS(X_\varsigma, \lambda), FS(X_\varphi, \lambda) \rangle \tag{114}$$

where $FS(X_\varsigma, \lambda)$ is the Fisher Score (FS), given two observations X_ς and X_φ, that is characterized by the log-likelihood of the generative model with respect to all the parameters and we have restricted the FK to the practical case where the Fisher Information Matrix (FIM) F_λ defined by

$$F_\lambda = \mathbb{E}[FS(X_\varsigma, \lambda) FS(X_\varsigma, \lambda)^T] \tag{115}$$

is assumed to be $\mathbf{I}$ with $\mathbb{E}$ denoting the expectation. On the other hand, FS is defined as:

$$FS(X, \lambda) = \nabla_{\lambda} \ln P(X|\lambda) \tag{116}$$

The FS reduces quantization error in comparison to other well-known methods such as the Bag of Features or the Bag of Visual words due to its primary and secondary statistics components. Effectively, the FK compares objects in higher spaces formed by the generative model as points in the Riemannian manifold. This enables the measurement of geodesic distances between the points along the manifold [127].

Given λ of a particular trained HMM, the log likelihood may be calculated using:

$$L(X|\lambda) = \ln P(X|\lambda) \tag{117}$$

$$= \ln \sum_{i=1}^{K} \rho_T(i) \tag{118}$$

$$= \ln \sum_{i=1}^{K} \pi_i \varkappa_i(X_1)\theta_1(i) \tag{119}$$

The derivatives of the Dirichlet-based HMM may then be defined as:

$$\nabla_{\lambda} L(X|\lambda) = \left[\frac{\partial L(X|\lambda)}{\partial \pi_i}, \frac{\partial L(X|\lambda)}{\partial b_{ii'}}, \frac{\partial L(X|\lambda)}{\partial \varepsilon_{id}}\right] \tag{120}$$

On the other hand, the derivatives of the BL-based HMM are denoted by:

$$\nabla_{\lambda} L(X|\lambda) = \left[\frac{\partial L(X|\lambda)}{\partial \pi_i}, \frac{\partial L(X|\lambda)}{\partial b_{ii'}}, \frac{\partial L(X|\lambda)}{\partial \delta_{id}}, \frac{\partial L(X|\lambda)}{\partial \alpha_i}, \frac{\partial L(X|\lambda)}{\partial \beta_i}\right] \tag{121}$$

Finally, the derivatives of the Generalized Dirichlet-based HMM may then be expressed as:

$$\nabla_{\lambda} L(X|\lambda) = \left[\frac{\partial L(X|\lambda)}{\partial \pi_i}, \frac{\partial L(X|\lambda)}{\partial b_{ii'}}, \frac{\partial L(X|\lambda)}{\partial \iota_{ij}}, \frac{\partial L(X|\lambda)}{\partial \vartheta_{ij}}\right] \tag{122}$$

Calculating each of these derivatives with respect to their respective parameters using Eq. (118) and

Eq. (119) results in:

$$\frac{\partial L(X|\lambda)}{\partial \pi_i} = \frac{\varkappa_i(X_1)\theta_1(i)}{\sum_{i=1}^{K} \pi_i \varkappa_i(X_1)\theta_1(i)} \tag{123}$$

$$\begin{aligned}\frac{\partial L(X|\lambda)}{\partial b_{ii'}} &= \frac{1}{P(X|\lambda)} \sum_{k=1}^{K} \frac{\partial \rho_T(k)}{\partial b_{ii'}} \\ &= \frac{1}{P(X|\lambda)} \sum_{k=1}^{K} \left(\frac{\partial}{\partial b_{ii'}} \sum_{l=1}^{K} \rho_{T-1}(l) b_{lk} \varkappa_k(X_T) \right) \\ &= \frac{1}{P(X|\lambda)} \sum_{k=1}^{K} \sum_{l=1}^{K} \frac{\partial \rho_{T-1}(l)}{\partial b_{ii'}} b_{lk} \varkappa_k(X_T) \\ &+ \partial \rho_{T-1}(i) \varkappa_{i'}(X_T)\end{aligned} \tag{124}$$

$$\begin{aligned}\frac{\partial L(X|\lambda)}{\partial \varepsilon_{id}} =& \frac{1}{P(X|\lambda)} \left(\sum_{i'=1}^{K} \sum_{k=1}^{K} \frac{\partial \rho_{T-1}(k)}{\partial \varepsilon_{id}} b_{ki'} \varkappa_{i'}(X_T) \right. \\ &\left. + \sum_{k=1}^{K} \partial \rho_{T-1}(k) b_{ki} \frac{\partial \varkappa_i(X_T)}{\partial \varepsilon_{id}} \right)\end{aligned} \tag{125}$$

$$\begin{aligned}\frac{\partial L(X|\lambda)}{\partial \delta_{id}} =& \frac{1}{P(X|\lambda)} \left(\sum_{i'=1}^{K} \sum_{k=1}^{K} \frac{\partial \rho_{T-1}(k)}{\partial \delta_{id}} b_{ki'} \varkappa_{i'}(X_T) \right. \\ &\left. + \sum_{k=1}^{K} \partial \rho_{T-1}(k) b_{ki} \frac{\partial \varkappa_i(X_T)}{\partial \delta_{id}} \right)\end{aligned} \tag{126}$$

$$\begin{aligned}\frac{\partial L(X|\lambda)}{\partial \alpha_i} =& \frac{1}{P(X|\lambda)} \left(\sum_{i'=1}^{K} \sum_{k=1}^{K} \frac{\partial \rho_{T-1}(k)}{\partial \alpha_i} b_{ki'} \varkappa_{i'}(X_T) \right. \\ &\left. + \sum_{k=1}^{K} \partial \rho_{T-1}(k) b_{ki} \frac{\partial \varkappa_i(X_T)}{\partial \alpha_i} \right)\end{aligned} \tag{127}$$

$$\frac{\partial L(X|\lambda)}{\partial \beta_i} = \frac{1}{P(X|\lambda)} \left(\sum_{i'=1}^{K} \sum_{k=1}^{K} \frac{\partial \rho_{T-1}(k)}{\partial \beta_i} b_{ki'} \varkappa_{i'}(X_T) + \sum_{k=1}^{K} \partial \rho_{T-1}(k) b_{ki} \frac{\partial \varkappa_i(X_T)}{\partial \beta_i} \right) \tag{128}$$

$$\frac{\partial \varkappa_i(X_t)}{\partial \varepsilon_{id}} = \Psi \left(\sum_{d=1}^{D} \varepsilon_{id} \right) - \Psi(\varepsilon_{id}) + \ln x_d \tag{129}$$

$$\frac{\partial \varkappa_i(X_t)}{\partial \delta_{id}} = \Psi \left(\sum_{d=1}^{D} \delta_{id} \right) - \Psi(\delta_{id}) + \ln x_d - \ln \sum_{d=1}^{D} x_d \tag{130}$$

$$\frac{\partial \varkappa_i(X_t)}{\partial \alpha_i} = \Psi(\alpha_i + \beta_i) - \Psi(\alpha_i) + \ln \sum_{d=1}^{D} x_d \tag{131}$$

$$\frac{\partial \varkappa_i(X_t)}{\partial \beta_i} = \Psi(\alpha_i + \beta_i) - \Psi(\beta_i) + \ln \sum_{d=1}^{D} (1 - x_d) \tag{132}$$

$$\frac{\partial \varkappa_i(X_t)}{\partial \iota_{id}} = \Psi(\iota_{id} + \vartheta_{id}) - \Psi(\iota_{id}) + \ln \sum_{d=1}^{D} \mathcal{Y}_d \tag{133}$$

$$\frac{\partial \varkappa_i(X_t)}{\partial \vartheta_{id}} = \Psi(\iota_{id} + \vartheta_{id}) - \Psi(\vartheta_{id}) + \ln \sum_{d=1}^{D} (1 - \mathcal{Y}_d) \tag{134}$$

with $\Psi(.)$ and $\Psi'(.)$ denoting the logarithmic first and second derivatives of the Gamma function, or the digamma and trigamma functions respectively.

4.3 Experimental Results for Dirichlet and Beta-Liouville

We evaluate our proposed models on the *Alpha DynTex* DT recognition benchmark dataset [133]. The dataset consists of three texture classes: grass, sea, and trees; with a total of 60 DTs. Samples of the dataset may be observed in Fig. 4.1

Figure 4.1: Samples from the *Alpha DynTex* dataset.

We first train the Dirichlet-based and BL-based HMM generative models on the dataset where we represent each of the DT video sequences with a series of extracted Local Binary Pattern (LBP) features. LBP are one of the most efficient features in texture recognition applications which was originally proposed in [134]. This set of extracted features represent the training and testing data with a leave-one-out cross validation scheme. A HMM is then trained for each class using the aforementioned data. For the testing stage, the likelihood of each testing video sequence is calculated by the respective three trained HMMs and the class label is assigned according to the maximum resulting likelihood.

Our experimental setup can be observed in Fig. 4.2. It is noteworthy to mention that the number of states are set to two and with the respective number of mixture components to be equal two for the Dirichlet HMM as well as the BL HMM as experimentally tested. A similar setup is carried out for the hybrid generative-descriminative model with a HMM trained for every single of the sequences each with a mixture of a single component. Then the FK is computed to train the SVM accordingly.

Our results for the generative approach can be observed for the Dirichlet and the BL HMMs

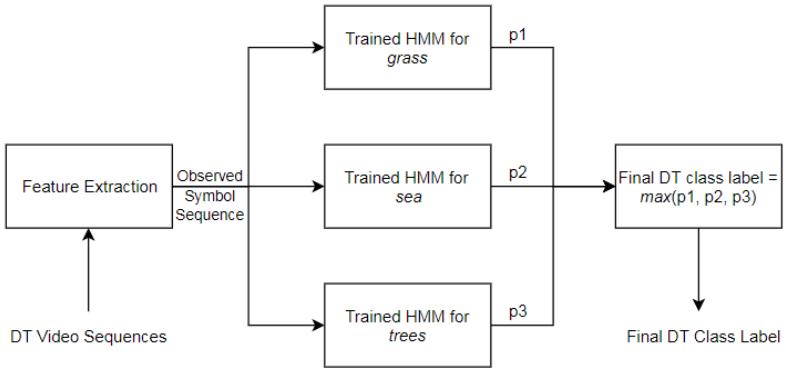
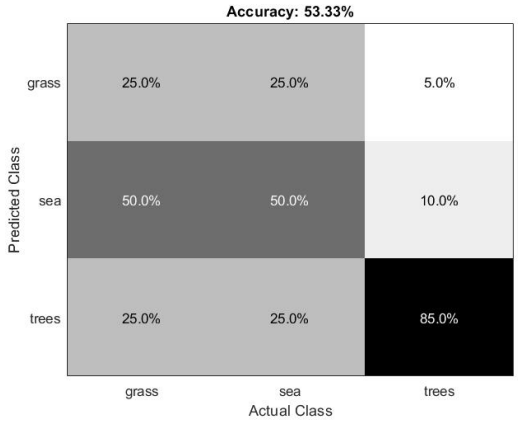
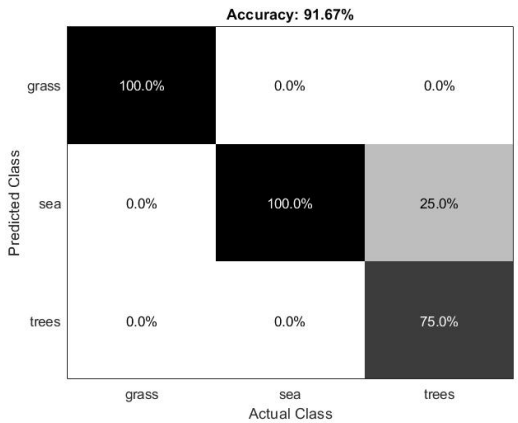


Figure 4.2: Experimental setup for testing for the proportional based hidden Markov models (HMM) for dynamic texture recognition *Alpha DynTex* dataset. p1, p2, and p3 are the respective likelihoods of each of the trained HMMs.



(a) Dirichlet-based HMM.



(b) BL-based HMM.

Figure 4.3: Resultant confusion matrices from the trained generative models.

in Fig. 4.3a and Fig. 4.3b respectively. As expected, the BL HMM achieves better results than the Dirichlet HMM given its improved capability to model proportional data. On the other hand, experimental results of the hybrid generative-discriminative approach can be seen in Table 4.1. From the results, it can be straightforwardly deduced that the generative-discriminative approach improves the modeling accuracy compared to the generative HMMs for both the Dirichlet-based and the BL-based HMMs respectively.

Table 4.1: Results of the proposed hybrid generative-discriminative approach.

HMM SVM	***Dirichlet-based***	***BL-based***
Accuracy	90.0%	94.7%

4.4 Experimental Results for Generalized Dirichlet

We evaluate the proposed model on the *Alpha DynTex* DT recognition benchmark dataset [133]. Three texture classes make up the dataset: grass, sea, and trees; for a total of sixty DTs. We compare our results with the generative Gaussian, Dirichlet, and GD HMMs. The Gaussian distribution represents the most popular choice that exists in the literature, but it might not always be the most appropriate choice [135, 136].

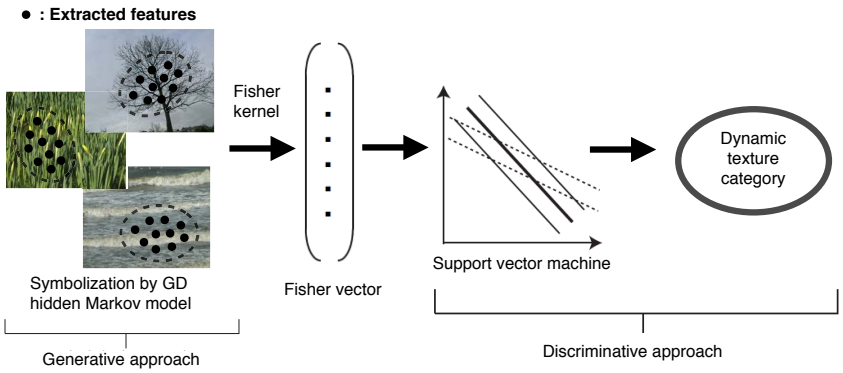


Figure 4.4: Experimental setup for the proposed model.

In our setup, we first train the benchmarking HMM generative models on each of the DT video sequences with a series of extracted Local Binary Pattern (LBP) features in a leave-one-out cross

validation scheme. The experimental setup for our proposed approach is shown in Fig. 4.4.

We then compute the accuracy of the models in order to evaluate the efficiency of our approach. As expected, the GD HMM achieves better results of 58.33% than the Dirichlet HMM at 53.33% given its improved capability to model proportional data. Both generative HMMs perform better than the Gaussian based HMM (50.00%) due to their finer modelling capabilities. Finally, our proposed hybrid generative-discriminative approach achieves superior accuracy of 73.33%.

Chapter 5

Maximum A Posteriori Approximation of Proportional Hidden Markov Models

Believe you can and you're halfway there.

Theodore Roosevelt

The MAP estimation is utilized in this chapter due to its parallel treatment of the parameter estimation to variational inference in terms of adding priors for a better result. However, it remains a point estimate; hence, its computational complexity is lower. On the other hand, it may reach comparable results to variational inference as is shown in this thesis and starting this chapter.

5.1 Maximum A Posteriori Approximation of the Dirichlet and Beta-Liouville Hidden Markov Models

In this section, we detail the formulation of the MAP approximation for proportional sequential data and focus on the Dirichlet and the BL-based HMMs.

5.1.1 Proposed Method

Proportional hidden Markov models

A HMM is characterized by an underlying stochastic process with K hidden states, each governed by an initial probability π, and the transition between the states $B = \{b_{ii'} = P(s_t = i'|s_{t-1} = i)\}$ at time t. In each state s_t, an observation is emitted corresponding to its respective parameters of a probability distribution $\varkappa$ with a mixing matrix $C = \{c_{ij} = P(m_t = j|s_t = i)\}$ where $j \in [1, M]$ such that M is the number of mixture components in set $L = \{m_1, ..., m_M\}$. Hence, a continuous HMM may be defined as $\lambda = \{B, C, \varkappa, \pi\}$. A graphical model of the latter HMM is depicted in Fig. 5.1. The likelihood of a sequence may then be denoted by:

$$p(X|B, C, \varkappa, \pi) = \sum_S \sum_L \pi_{s_1} \left[\prod_{t=2}^{T} b_{s_{t-1}, s_t}\right] \left[\prod_{t=1}^{T} c_{s_t, m_t} p(X_t|\Lambda_{s_t, m_t})\right] \tag{135}$$

where $\Lambda_{ij} = (\Lambda_{1ij}, ..., \Lambda_{Dij})$ with $\varkappa$ defined according to the Dirichlet, GD, or BL distributions for proportional data. For simplification purposes, we derive the model for a unique sequence. A summation over sequences may then be added for inclusion of more sequences; a usual case to prevent overfitting.

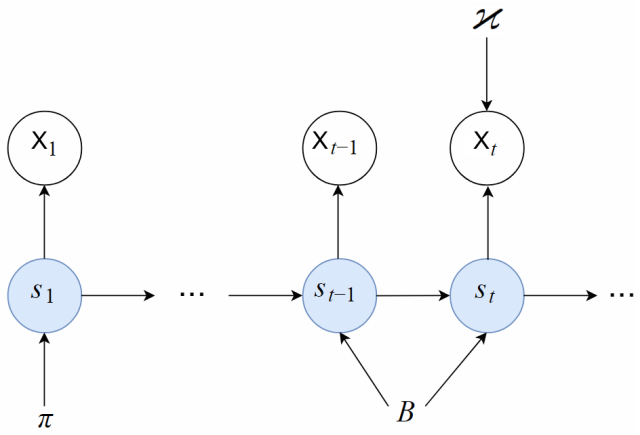


Figure 5.1: Graphical model representation of a continuous hidden Markov model. Symbols in unshaded circles denote the observed variables, symbols in shaded circles indicate the hidden states, and edges represent conditional dependencies between the states or the variables.

Maximum A Posteriori approximation

We use the EM algorithm in order to implement the proposed MAP approximation for Dirichlet and BL HMMs. First, we denote the estimates of the state and mixture component $\gamma^t_{s_t,m_t} \triangleq p(s_t, m_t|X_0, \ldots, X_T)$ and $\eta^t_{s_t,s_{t+1}|X_0,\ldots,X_T} \triangleq p(s_t, s_{t+1}|X_0, \ldots, X_T)$ for the estimate of the local states sequence given the entire observation sequence. These are obtained for all t in the E-step with the traditional HMM forward-backward algorithm that is not detailed here. An interested reader is referred to [10]. That is we maximize the data log-likelihood via its lower bound:

$$\begin{aligned}\mathcal{L}(\lambda|X) &= p(X|\lambda) = E(X,\lambda) - R(Z) \\ &= \sum_Z p(Z|X)\ln(p(X,Z)) - \sum_Z p(Z|X)\ln(p(Z|X)) \\ &= \sum_Z p(Z|X)\ln(p(X)) \\ &= \ln(p(X))\sum_Z p(Z|X) = \ln(p(X))\end{aligned} \tag{136}$$

where Z represents the hidden variables, $E(X,\lambda)$ is the complete-data log-likelihood with the true or the maximized parameters λ, and $R(Z)$ is the log-likelihood of the hidden states given the observations or the sequences. The latter also representing the amount of information brought by the hidden data in the form of an entropy. The expected complete-data log-likelihood may then be expressed as:

$$E(X,\lambda,\lambda^{old}) = \sum_Z p(Z|X,\lambda^{old})\ln(p(X,Z|\lambda)) \tag{137}$$

where the non-optimized parameters complete data log-likelihood $E(X,\lambda,\lambda^{old}) \leq E(X,\lambda)$; hence, the lower bound of the data likelihood is $E(X,\lambda,\lambda^{old}) - R(Z)$. This is equivalent to Eq. (64) when expanded where $p(X_t|\varkappa_{s_t,m_t})$ is a Dirichlet or BL mixture in this work.

Formally, a D-dimensional Dirichlet distribution is denoted by Eq. (183) where $\vec{\varepsilon} = (\varepsilon_1, ..., \varepsilon_D)$ is the real and strictly positive parameter of the distribution and $X \in \mathbb{R}^D_+$, $\sum_{d=1}^D x_d = 1$ corresponding to the D-dimension proportional vector that adds up to one. Consequently, the complete data log-likelihood with the Dirichlet mixture may be split with the logarithm sum-product property

as follows:

$$\ln(p(X,Z|\lambda)) = \sum_{t=2}^{T}\ln(b_{s_{t-1},s_t}) + \sum_{t=1}^{T}\ln(c_{s_t,m_t}) + \ln(\pi_{s_1}) + \sum_{t=1}^{T}\left[\sum_{d=1}^{D}[\ln(x_d)+ \Psi(\sum_{d=1}^{D}\varepsilon_d) - \Psi(\varepsilon_d) - \ln(\sum_{d=1}^{D}x_d)\right]\right] \tag{138}$$

A better model of proportional time series data has been proposed with the BL distribution in [46]. This distribution is closely related to the Dirichlet, but it relaxes the constraint of negative covariance at the cost of two additional parameters. As a matter of fact, the Dirichlet distribution is a special case of the BL distribution. The latter is expressed by:

$$BL(X|\vec{\delta},\alpha,\beta) = \frac{\Gamma(\sum_{d=1}^{D}\delta_d)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\prod_{d=1}^{D}\frac{x_d^{\delta_d-1}}{\Gamma(\delta_d)}\left(\sum_{d=1}^{D}x_d\right)^{\alpha-\sum_{d=1}^{D}\alpha_d}\left(1-\sum_{d=1}^{D}x_d\right)^{\beta-1} \tag{139}$$

where $\vec{\delta} = (\delta_1, ..., \delta_D)$, α, and β are real and strictly positive parameters of the BL distribution, $\Gamma(t) = \int_0^\infty X^{t-1}e^{-X}dX$ is the Gamma function, and X is a $D+1$ dimensional vector whereby $X \in \mathbb{R}_+^D$ and $\sum_{d=1}^{D}x_d < 1$. In this case, the complete data log-likelihood is expanded as:

$$\ln(p(X,Z|\lambda)) = \sum_{t=2}^{T}\ln(b_{s_{t-1},s_t}) + \sum_{t=1}^{T}\ln(c_{s_t,m_t}) + \ln(\pi_{s_1}) + \sum_{t=1}^{T}\Bigg[\Psi(\sum_{d=1}^{D}\delta_d) + \Psi(\alpha+\beta) - \Psi(\alpha) - \Psi(\beta) + (\alpha - \sum_{d=1}^{D}\delta_d)\ln(\sum_{d=1}^{D}x_d)) + (\beta-1)\ln(1-\sum_{d=1}^{D}x_d)) + \sum_{d=1}^{D}[(\delta_d-1) \times\ln(x_d)) - \Psi(\delta_d)]\Bigg] \tag{140}$$

We then denote the complete data log-likelihood $\ln(p(X,Z|\lambda))$ with:

$$\mathcal{Q}(\lambda^t, \lambda^{t-1}) = \mathbb{E}[\ln(p(X,Z|\lambda^t)|Z,\lambda^{t-1}] \tag{141}$$

where λ^t is the set of the HMM parameters for the current iteration while λ^{t-1} represents the

parameters from the previous iteration. We then incorporate additional terms to $\mathcal{Q}(\lambda^t, \lambda^{t-1})$ to integrate priors $\mathcal{A}(\lambda^t)$ for the HMM parameters which characterize the MAP estimation. This is in contrast to the Baum Welch; i.e., the maximum likelihood algorithm for finding the optimum parameters for HMMs, where Eq. (141) would suffice. The modified expression is then formulated as:

$$\mathcal{S} = \mathcal{Q}(\lambda^t, \lambda^{t-1}) + \ln(\mathcal{A}(\lambda^t)) \tag{142}$$

Next, we determine appropriate priors for the HMM parameters for the proposed MAP estimation. Since the coefficients of the parameters π, B, and C are all strictly positive, with values less than one, and sum to one for each row summation, we choose Dirichlet distributions for their priors as follows:

$$\begin{aligned} p(\pi) = \mathcal{D}(\pi|\phi^\pi) &= \mathcal{D}(\pi_1, ..., \pi_K|\phi_1^\pi, ..., \phi_K^\pi), \\ p(B) &= \prod_{i=1}^{K} \mathcal{D}(b_{i_1}, ..., b_{i_K}|\phi_{i_1}^B, ..., \phi_{i_K}^B), \\ p(C) &= \prod_{i=1}^{M} \mathcal{D}(c_{i_1}, ..., c_{i_M}|\phi_{i_1}^C, ..., \phi_{i_M}^C) \end{aligned} \tag{143}$$

Hence, the update equations to be computed in the M-step of the MAP estimation are the following:

$$\pi_i = \frac{\gamma_i^0 + \phi_i^\pi - 1}{\sum_{i=1}^{K}(\gamma_i^0 + \phi_i^\pi - 1)} \tag{144}$$

$$B_{ii'} = \frac{\sum_{t=1}^{T} \eta_{i,i'}^t + \phi_{i'_i}^B - 1}{\sum_{i=1}^{K}(\sum_{t=1}^{T} \eta_{i,i'}^t + \phi_{i'_i}^B - 1)} \tag{145}$$

$$C_{ij} = \frac{\sum_{t=1}^{T} \gamma_{i,j}^t + \phi_{ij}^C - 1}{\sum_{j=1}^{M}(\sum_{t=1}^{T} \gamma_{i,j}^t + \phi_{ij}^C - 1)} \tag{146}$$

Similarly, conjugate priors must be defined over the Dirichlet and the BL parameters $\vec{\varepsilon}$, $\vec{\delta}$, α, and β. The Gamma distribution $\mathcal{G}(.)$ is a suitable fit for positive conjugate prior approximations of

these parameters [92]. As such, the priors over the distribution specific parameters are:

$$p(\{\vec{\varepsilon}\}_{i,j,d=1}^{K,M,D}) = \prod_{i=1}^{K}\prod_{j=1}^{M}\prod_{d=1}^{D}\mathcal{G}(\varepsilon_{ijd}|\rho_{ijd},\zeta_{ijd}), \tag{147}$$

$$p(\{\vec{\delta}\}_{i,j,d=1}^{K,M,D}) = \prod_{i=1}^{K}\prod_{j=1}^{M}\prod_{d=1}^{D}\mathcal{G}(\delta_{ijd}|u_{ijd},v_{ijd}), \tag{148}$$

$$p(\{\alpha\}_{i,j=1}^{K,M}) = \prod_{i=1}^{K}\prod_{j=1}^{M}\mathcal{G}(\alpha_{ij}|g_{ij},h_{ij}), \tag{149}$$

$$p(\{\beta\}_{i,j=1}^{K,M}) = \prod_{i=1}^{K}\prod_{j=1}^{M}\mathcal{G}(\beta_{ij}|e_{ij},r_{ij}) \tag{150}$$

where the hyperparameters ρ, ζ, u, g, h, e, r, and v are strictly positive.

The update equations for the distribution specific parameters require the use of the Newton-Raphson estimation method for maximizing the lower bound of the respective mixtures. This estimation has the following generic formula:

$$\theta^{new} = \theta^{old} - H(\theta^{old})^{-1}\frac{\partial\mathcal{L}(X|\theta^{old})}{\partial\theta^{old}} \tag{151}$$

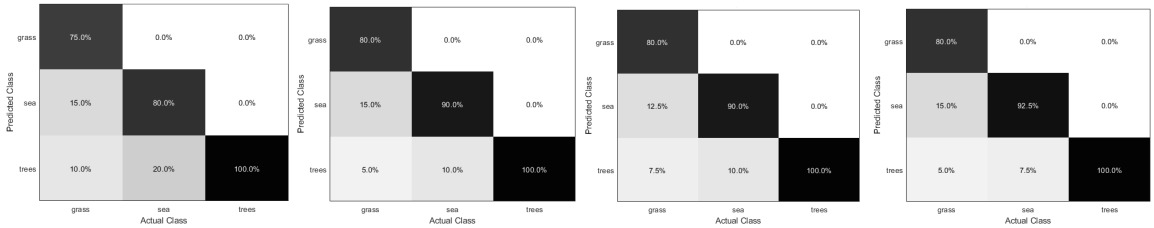
where $H()$ is the Hessian matrix that is formed of the second order derivatives of the data likelihood corresponding to θ in the former equation. The computation of this equation and its sub-parts is not detailed here, but an interested reader is referred to the full derivations in [29]. We also note that the derivations for both of the distributions are highly related since the Dirichlet distribution is merely a special case of the BL as mentioned beforehand.

5.1.2 Experimental Results

Dynamic texture classification

In this section, we validate our model on dynamic texture classification [48, 7]. We employ our proposed models on the *Alpha DynTex* dynamic texture recognition benchmark dataset [133]. The dataset consists of three texture classes: grass, sea, and trees; with a total of sixty sequences. Samples of the dataset may be observed in Fig. 4.1. We chose this dataset for its balanced classes.

This enables us to seamlessly train each of the classes model with a correspondingly equivalent leave-one-out cross-validation schema with a series of extracted Local Binary Pattern (LBP) features [134].

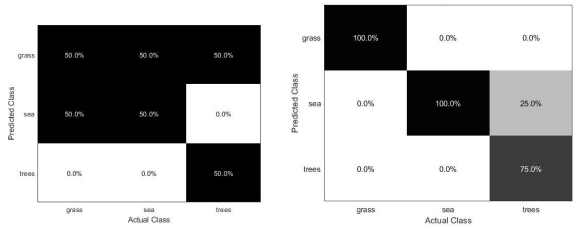


(a) Baum Welch trained Dirichlet HMMs.

(b) MAP trained Dirichlet HMMs.

(c) Baum Welch trained BL HMMs.

(d) MAP trained BL HMMs.



(e) Baum Welch trained Gaussian HMMs.

(f) Variational inference trained BL HMMs.

Figure 5.2: Resultant confusion matrices from the trained hidden Markov models (HMM) for dynamic texture classification.

For the testing stage, the likelihood of each testing video sequence is calculated by the respective three trained HMMs and the class label is assigned according to the maximum resulting likelihood. The experimental setup can be observed in Fig. 4.2. It is noteworthy to mention that the number of states are set to two with the respective number of mixture components to be equal two as experimentally tested.

Table 5.1: Accuracy of the trained Dirichlet and BL HMMs for dynamic texture classification.

HMM	**Accuracy (%)**
Gaussian (Baum Welch)	50.00
Dirichlet (Baum Welch)	85.00
BL (Baum Welch)	90.00
Dirichlet (MAP)	90.00
BL (MAP)	90.83
BL (Variational Inference)	91.76

The results of the Baum Welch and MAP based Dirichlet and BL HMMs are shown in the form of confusion matrices in Fig. 5.2a, Fig. 5.2b, Fig. 5.2c, and Fig. 5.2d respectively. Fig. 5.2e

and Fig. 5.2f depict the confusion matrices of the benchmarking Gaussian HMMs and variational inference trained BL HMMs. The former represents the traditional choice that is usually made in the literature, while the latter is the latest proposed approach for proportional HMMs.

We then compute the accuracy of the models in order to evaluate the efficiency of our approach. Accuracy refers to the number of correctly identified dynamic texture sequences and is commonly calculated with $\mathrm{TP}/(\mathrm{TP}+\mathrm{TN})$ where TP represents the number of true positives correctly identified by the approach, and TN denotes the number of true negatives. The accuracy for each of the models can be observed in Table 5.1. As expected, the MAP HMMs achieve better results than the Baum Welch trained and comparably to the variational inference method. However, MAP achieves such results at lower computational cost as well as utilizing less complex derivations which backs up its wide applicability. Moreover, using Dirichlet and BL HMMs improve results in comparison to using the Gaussian distribution for the emissions given their improved capabilities to model proportional data.

Infrared action recognition

In this section, we validate our proposed model on infrared (IR) action recognition (AR). We present our experimental results on the challenging AR IR dataset, *InfAR* [98]. The training and testing sets consist of single person action with 10 video samples for seven classes in a leave-one-out cross validation scheme. Sample images for each of the classes are depicted in Fig. 3.2. We extract interest point detectors along the motion trajectory as in [101] to represent each of the sequences with a series of extracted histogram of optical flow (HOF) and motion boundary histogram (MBH) [100].

We utilize the same experimental setup as the one used for the dynamic texture classification application. The confusion matrices of the different features with the proposed Baum Welch and MAP trained HMMs can be observed in Fig. 5.5 with the respective accuracy computed shown in the top center of each of the subfigures.

It is evident that the horizontal and vertical MBH features are better suited for the trained models across the various setups given the accuracy results. Using feature selection may improve the results of the HOF based models given that it has a higher dimension than the MBH features. Furthermore,

as expected, the MAP approximated HMMs show better performance than the ones trained with the Baum Welch approach. The employment of the more flexible BL distribution also improves the results due to its better capability to model the proportional data sequence. This is due to the additional number of parameters that enable the shaping of an emission distribution that is better fit for the data. Furthermore, it also overcomes the negative covariance limitation that the Dirichlet distribution enforces on the data. These results may be observed in Fig. 5.3.

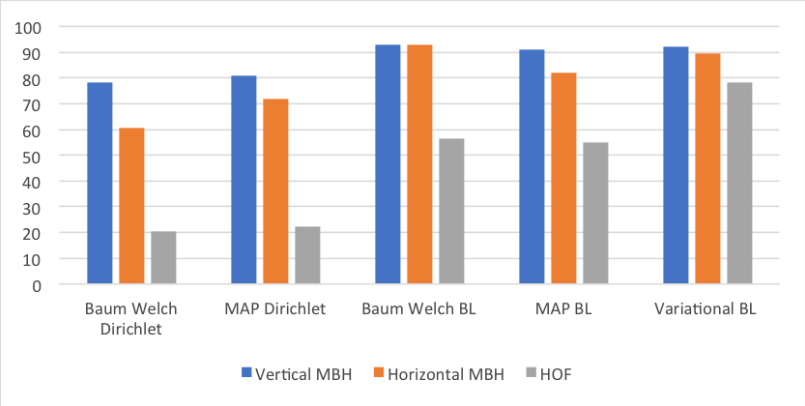


Figure 5.3: Comparison of trained proposed HMMs for infrared action recognition task on the InfAR dataset with state of the art HMM methods cited in the manuscript. Labels across the x-axis depict the names of the models while AP percentages are shown across the y-axis.

Furthermore, our results are also comparable to several others in the literature as can be observed in Fig. 5.4 with the average precision (AP). This includes various handcrafted features extracted for the InfAR dataset such as HOF [98], dense trajectories [104], and improved dense trajectories [105]. We also compare to deep learning models such as the two-stream 3D convolutional neural network (CNN) [102], the optical flow field 3D CNN [102], and the three-stream trajectory-pooled deep-convolutional descriptors methodology in [103].

5.2 Maximum A Posteriori Approximation of the Generalized Dirichlet Hidden Markov Models

Recently, [46] have also successfully presented variational inference as a way to mitigate hindrances in the estimation of the parameters of proportional HMMs. Nevertheless, in this chapter, we propose the use of the Maximum A Posteriori (MAP) approximation for the GD HMMs. This

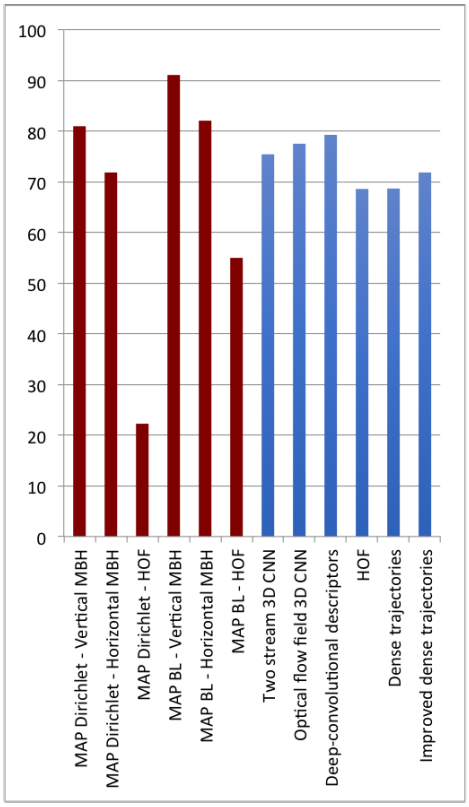
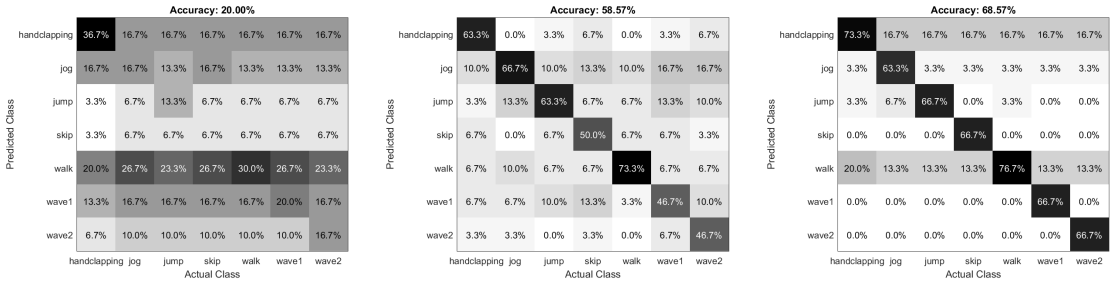
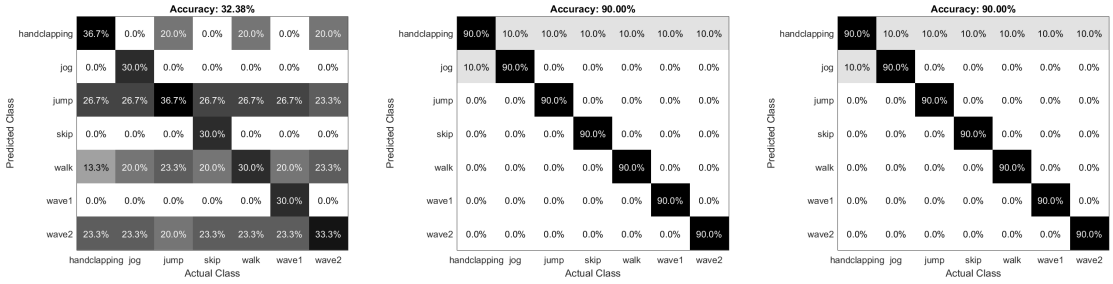


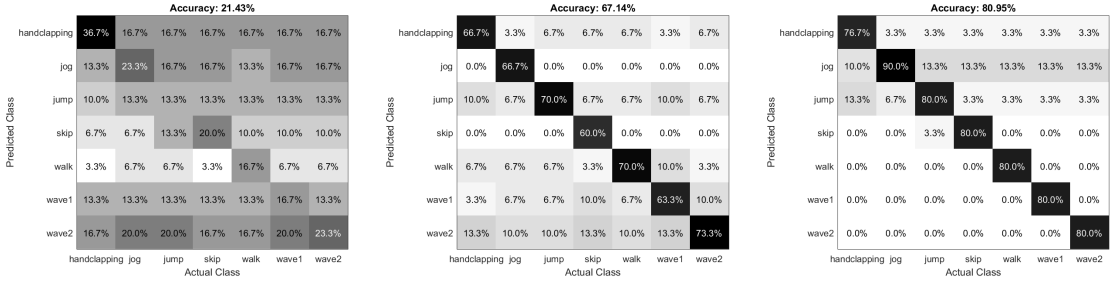
Figure 5.4: Comparison of trained proposed HMMs (in red) for infrared action recognition task on the InfAR dataset with state of the art methods cited in the manuscript (in blue). Labels across the x-axis depict the names of the models while AP percentages are shown across the y-axis.



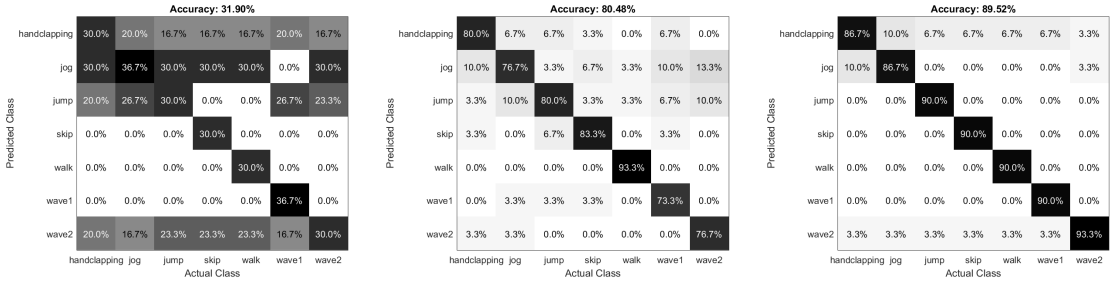
(a) Dirichlet HMM with HOF. (b) Dirichlet HMM with horizontal MBH. (c) Dirichlet HMM with vertical MBH.



(d) BL HMM with HOF. (e) BL HMM with horizontal MBH. (f) BL HMM with vertical MBH.



(g) Dirichlet HMM with HOF. (h) Dirichlet HMM with horizontal MBH. (i) Dirichlet HMM with vertical MBH.



(j) BL HMM with HOF. (k) BL HMM with horizontal MBH. (l) BL HMM with vertical MBH.

Figure 5.5: Resultant confusion matrices from the trained hidden Markov models (HMM) for infrared action recognition. (a)-(f) are trained using the Baum Welch approach, while (g)-(l) are approximated with the Maximum A Posteriori method proposed.

is because while both are approximation approaches, the MAP method has a lower computational cost than variational inference. This is due to the reduced number of mathematical computations that is required by the prior approach.

Indeed, a naive view of the MAP-based learning of HMMs would reduce it into the famous Baum Welch approach with the addition of priors over the parameters of the model. Moreover, both the variational and the MAP approaches share the same fundamental principle of placing appropriate priors over the parameters to be estimated for improving the performance of the evaluation.

5.2.1 Proposed Method

We propose a MAP approach for proportional time series data with GD-based HMM in this chapter. The procedure is similar as the one detailed in Chapter 5.1. Hence, we only discuss the differences in the equations as well as the resultant experimental results. This starts in Eq. (64) when expanded where $p(X_t|\varkappa_{s_t,m_t})$ is now a GD mixture in this work.

Formally, a GD is denoted by:

$$GD(X|\vec{\iota},\vec{\vartheta}) = \prod_{d=1}^{D} \frac{\Gamma(\iota_d+\vartheta_d)}{\Gamma(\iota_d)\Gamma(\vartheta_d)} x_d^{\iota_d-1} \left(1-\sum_{r=1}^{d} x_r\right)^{\zeta_d} \tag{152}$$

where $\vec{\iota} = (\iota_1,...,\iota_D), \vec{\vartheta} = (\vartheta_1,...,\vartheta_D)$ are the real and strictly positive parameters of the GD distribution and $X \in \mathbb{R}_+^D$ and $\sum_{d=1}^{D} x_d < 1$ corresponding to the $(D+1)$-dimensional proportional vector that adds up to one. Finally, ζ_d is computed using the parameters of the distribution as $\vartheta_d - \iota_{d+1} - \vartheta_{d+1}$, when $d \neq D$. Otherwise, $\zeta_d = \vartheta_D - 1$.

Next, the update equations to be computed in the M-step of the MAP estimation. AS such, conjugate priors must be defined over the GD parameters. The Gamma distribution $\mathcal{G}(.)$ is a suitable fit for positive conjugate prior approximations of these parameters [92]. As such, the priors over the

distribution specific parameters are:

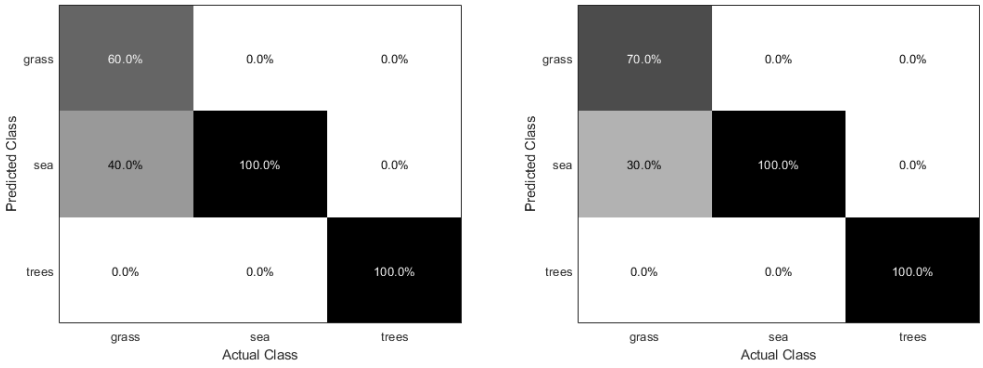
$$p(\{\vec{\iota}\}_{i,j,d=1}^{K,M,D}) = \prod_{i=1}^{K}\prod_{j=1}^{M}\prod_{d=1}^{D}\mathcal{G}(\iota_{ijd}|\rho_{ijd},\zeta_{ijd}), \tag{153}$$

$$p(\{\vec{\vartheta}\}_{i,j,d=1}^{K,M,D}) = \prod_{i=1}^{K}\prod_{j=1}^{M}\prod_{d=1}^{D}\mathcal{G}(\vartheta ijd|u_{ijd},v_{ijd}), \tag{154}$$

$$\tag{155}$$

where the hyperparameters ρ, ζ, u, and v are strictly positive. The update equations for the distribution specific parameters require the use of the Newton-Raphson estimation method for maximizing the lower bound of the respective mixtures.

5.2.2 Experimental Results



(a) Baum Welch GD. (b) MAP GD.

Figure 5.6: Resultant confusion matrices from the trained hidden Markov models for dynamic texture recognition. GD denotes the Generalized Dirichlet.

In this section, we validate our model on dynamic texture classification. Dynamic textures have proven to be of diverse benefits across various domains that include but are not limited to video synthesis [116], abnormal motion detection [117], human action recognition [115], video segmentation [111, 137], and video classification [108]. They constitute of dynamic complex objects such as grass moving in the wind [107]. As such, they are characterized by a stationary behaviour in time [116, 138].

We validate our proposed HMM on the *Alpha DynTex* dynamic texture recognition benchmark

dataset [133]. To train the GD HMM models on the dataset, we represent each of the dynamic texture video sequences with a series of extracted Local Binary Pattern (LBP) features. Indeed, we follow a similar training and testing experimental setup as in Section 5.1.2.

For benchmarking purposes, we also compare with Baum Welch training of GD HMM and the Gaussian HMM as well as the latest proposed learning approach for proportional HMMs. The latter refers to the variational learning of BL-based HMMs [46]. These have reported superior results for proportional time series data modeling to many other methods in the literature, and an interested reader is referred to the paper for further details.

Table 5.2: Accuracy of the trained GD HMMs for dynamic texture classification.

HMM	**Accuracy (%)**
Gaussian (Baum Welch)	50.00
GD (Baum Welch)	86.67
GD (MAP)	90.00
BL (Variational Inference)	91.76

We then compute the accuracy of the models in order to evaluate the efficiency of our approach. Accuracy refers to the number of correctly identified dynamic texture sequences. In other words, accuracy represents the overall correctness of the system, and is given by

$$ACC = \frac{TP + TN}{P + N} \tag{156}$$

where ACC stands for accuracy, TP is the number of true positives, TN is the number of true negatives, P are all the positive or correct occurrences, while N are all the negative or wrong occurrences. The accuracy for each of the models can be observed in Table 5.2.

We also evaluate the model in terms of two other measures; namely, the precision and the recall. Precision (or positive predictive) value is the measure of accuracy when a certain class is predicted. Precision is defined as

$$PPV = \frac{TP}{TP + FP} \tag{157}$$

where PPV stands for precision, or positive predictive value. Finally, we use recall (or sensitivity) as a measure of the model's capability to select occurrences of a particular class from a given data

set. Recall is associated with a true positive rate and is given by

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{158}$$

where TPR stands for recall, or true positive rate. These are shown in Fig. 5.7.

As expected, the MAP HMM achieves better results than the Baum Welch and comparably to the variational inference method. However, the MAP trained HMM achieves such results at lower computational cost due to utilizing less complex derivations which backs up its wide applicability. Moreover, using GD and BL HMMs improve results in comparison to using the Gaussian distribution for the emissions given their improved capabilities to model proportional data. It is important also to mention that the slight improvement of the recall in case of the BL-HMMs that are trained with variational inference is due to the superiority of the learning approach. Nonetheless, the MAP is significantly less mathematically complex as can be observed from the mathematical derivations of both approaches. In particular, we do not seek to present the best accuracy but rather present a model that is capable of performing comparably to the best proposed HMMs for proposed model at a conservatively less requirement for mathematical computations.

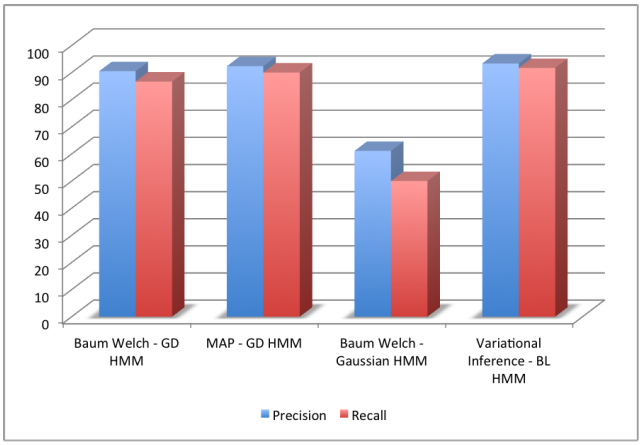


Figure 5.7: Precision and recall measures of the trained HMMs for dynamic texture classification.

Chapter 6

Simultaneous Feature Selection Paradigm for Proportional HMMs

> Nothing is IMPOSSIBLE. The word itself says "I'm Possible!"
>
> Audrey Hepburn

In this chapter, we now focus on incorporating a simultaneous feature selection paradigm in the MAP approximation of the GD-based HMM framework. This results in a holistic treatment of proportional sequential data without the need for a preprocessing feature selection algorithm. The remainder of the chapter introduces, derives, and analyses the results of the proposed.

6.1 Introduction

Recently, data modeling and analysis have shown unprecedented advances. This is majorly in debt to the amelioration of computational resources and the steady growth of the daily rate of generated data. A persistent increase in media content is a particular highlight [139, 49]. Ergo, video-based algorithms continue to be actively researched. This is a subcategory of spatiotemporal modeling; a research theme where space and time dimensions are interconnected.

One of the main modeling methods to capture such dual dependency is the hidden Markov model (HMM) [120]. HMMs employ a compact set of features to extract underlying statistics in a

double stochastic model [140, 3]. Since HMMs fall under the generative model category of machine learning, they require less training data [87, 48] than discrimintaive models. HMMs are thoroughly recognized for speech recognition [141], speech processing [10], genomics [142], handwritten word recognition [143], and financial prediction [144] applications. It is also currently used in various areas including object classification [145] and detection of unusual events [146, 147].

Computation of a sequence probability given a model that we have trained is required for the utilization of HMMs in classification. The parameters of each of the HMMs are appraised to model classes correspondingly.The training procedure optimizes the probability of the training observation set for a class, traditionally with the Baum Welch approach by deploying the Expectation Maximization (EM) framework [10]. However, there is no guarantee for the convergence of the Baum Welch. Indeed, the large multimodal nature of the likelihood function renders it vulnerable to either underfit or overfit the estimation of the parameters [148].

On the other hand, a significant part of the encompassing performance of the model lies on choosing a proper emission distribution for the HMM that has the best capability to model the underlying nature of the data. A typical assumption is to utilize the Gaussian distribution. Nonetheless, this impacts the final performance of the model as various applications employ proportional input features whereby the unbounded support character of the Gaussian distribution is not necessary. Indeed, the Gaussian distribution is not the most suitable option in such cases. Such a choice is best taken by inspecting the characteristics of the data as supported by [149, 150].

Recent research has also presented a variational approximation method for proportional HMMs to alleviate such obstacles in the estimation of the parameters [46]. In contrast, we alternatively propose the employment of the Maximum A Posteriori (MAP) approach for Generalized Dirichlet (GD)-based HMMs. Though the latter is also an approximation technique where the improvement of the performance of the evaluation depends on the placement of suitable priors over the parameters to be estimated, the MAP methodology has a lower computational overhead. The process also results in smoothing the likelihood function which consequently lessens its multimodality; improving the estimation of the objective global maximum. This was also proven in other proportional HMMs that we investigated in [2], particularly for the Dirichlet and the Beta-Liouville HMMs, and we also draw inspiration from [50] to further inspect the MAP technique as well as [51].

We also propose the incorporation of a feature selection paradigm [151, 152]. Intuitively, a larger set of features has better capability of modeling a dataset and consequently a finer efficiency of the resultant model.

Nonetheless, noise, practical informativeness, or redundancy of select features can hamper the performance [153]. Such irrelevancy can lead to uncertain measures of homogeneity through that introduced bias. Reducing such a set of extracted features according to its relevancy is performed via feature selection. In addition to increasing the models' performance, this also aids in enhancing the interpretation of the model and reducing the chances of overfitting [154]. We validate the proposed model with two challenging real applications. It is noteworthy to mention that this study is novel in its treatment of the feature selection paradigm in terms of the mathematical derivations required for the deployment of proportional HMMs in contrast to the traditional Gaussian-based [155, 50, 156, 157].

Thus, our contributions are fourfold:

- We provide the derivations of a novel approach for proportional data modeling using MAP approximation for HMMs with a simultaneous feature selection algorithm. We focus on the GD distribution given its established effectiveness in modeling proportional sequential data.

- For validation, we apply our approach in categorization of dynamic textures and recognition of infrared actions. Each of the applications may be applied to form the basis for various critical tasks such as security threat detection in videos.

- We perform the first evaluation of the MAP trained GD HMMs for infrared action classification in addition to the proposed model evaluation. This aims to clearly advocate the proposed simultaneous feature selection framework.

- For benchmarking, we compare our proposed framework with multiple relevant approaches in the literature including the Gaussian and the Dirichlet benchmarks.

The rest of the chapter is organized as follows: Section 6.2 describes the proposed model and algorithms and Section 6.3 presents the experimental setup and analyzes the results for each of the applications.

6.2 Proposed proportional hidden Markov models with simultaneous feature selection

We begin by detailing the principal elements of our approach. Section 6.2.1 inspects the HMM for proportional sequential data with the GD distribution and the traditional forward-backward approach, Section 6.2.2 discuses the feature saliency model, Section 6.2.3 details the MAP approximation with simultaneous feature selection framework, and Section 6.2.4 discusses the complete proposed algorithm. For ease of mathematical reference, a list of the symbols utilized in this manuscript with their corresponding definitions can be observed in Table 6.1.

6.2.1 Proportional hidden Markov models

K hidden states incarnate the underlying stochastic process that characterize a HMM. Each of these states has an initial probability π with the matrix $B = \{b_{ii'} = P(s_t = i'|s_{t-1} = i)\}$ that defines the transition between them at time t. The emission of an observation in s_t has parameters that follow the modeling probability density $\varkappa$ with mixing weights $C = \{c_{ij} = P(m_t = j|s_t = i)\}$ where $j \in [1, M]$, M represents the mixture's number of components in $L = \{m_1, ..., m_M\}$ [2]. Ergo, $\Lambda = \{B, C, \varkappa, \pi\}$ may define a continuous HMM which is shown in Fig. 6.1. Mathematically, this is denoted by Eq. (64) where S and L are the sets of states and mixture components in the HMM, respectively. $\varkappa_{ij} = (\varkappa_{1ij}, ..., \varkappa_{Dij})$ with $\varkappa$ defined by the GD. We deduce the model for a particular observation for clarity. However, a mere summation over sequences can incorporate more as needed. Indeed, that is the approach that is undertaken in our experiments. This aids in mitigating overfitting.

When HMMs are utilized for classification, the probability that a sequence was produced by Λ may be computed by the forward-backward algorithm [51]. Λ corresponds to a class whose parameters are computed or approximated as appropriate. For each class, this training procedure then involves the maximization of the probability of the training data using MAP.

The probability of the occurrence of state s_i at time t is computed by the forward algorithm using the relevant partial observation and Λ. The forward variable is defined by $\rho_t(i) = P(X_1, X_2, ..., X_t, s_t = i|\Lambda)$ that is computed by [48]:

Table 6.1: Definitions of symbols utilized in the article.

Symbol	Definition
t	time index
K	number of states
D	number of feature dimensions
B	transition matrix
i, i'	state index
d	dimension index
$b_{ii'}$	transition matrix index
$\varkappa$	probability distribution for HMM emission
π	initial probability
C	mixing matrix
M	number of mixture components
j	mixture component index
L	set of mixtures
S	set of states
m_M	each component of a mixture in a state
c_{ij}	mixing index
$p(X\|B, C, \varkappa, \pi)$	likelihood of a sequence
$\lambda_{ij} = (\lambda_{1ij}, ..., \lambda_{Dij})$	parameters of the GD
c_{s_t,m_t}	HMM mixing weight of a mixture component in a state
b_{s_{t-1},s_t}	a transition weight between states of a HMM
X	time series data, Y is also used in the manuscript for sequential data
$\rho_t(i)$	forward variable
$\theta_t(i)$	backward variable
$\gamma^t_{s_t,m_t}, \eta$	forward-backward algorithm resultant variables
$\mathfrak{Z}$	hidden variables
ι, ϑ	GD distribution parameters
ζ_d	GD distribution parameters
ϱ, ζ, u, and v	GD hyperparameters
Λ	A HMM model
Beta $(X_{dt}\|\epsilon_d, \tau_d)$	the distribution of irrelevant feature(s)
φ	feature saliency
z_d	feature assignment
$\mathbb{L}$	sequence likelihood
$\mathbb{H}$	Hessian matrix
$\mathbb{E}$	expectation of a variable
Dirichlet	Dirichlet distribution prior
$\mathcal{G}$	Gamma distribution prior
$\mathfrak{E}(Y, \Lambda)$	complete data log-likelihood with set of true parameters
$\mathfrak{R}(\mathfrak{Z})$	log-likelihood of the hidden variables given the observations
$\mathbb{S}$	complete data log-likelihood
$\mathbb{Q}(\Lambda^t, \Lambda^{t-1})$	complete data log-likelihood without MAP priors
$\mathbb{A}(\Lambda^t)$	MAP estimation priors for the model parameters
TP	true positives
TN	true negatives

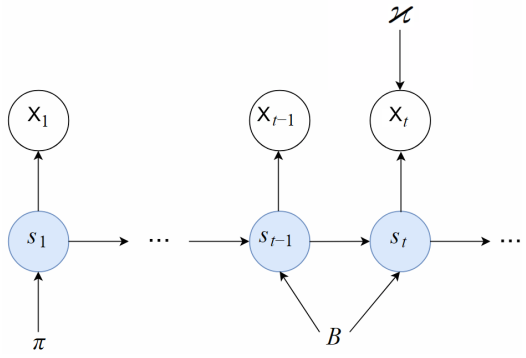


Figure 6.1: Depiction of Λ. Symbols in uncoloured circles represent observed variables whereas states are in coloured ones and conditional dependencies are denoted by edges [2].

(1) Initialize forward probabilities with the joint probability of s_t and the first observation X_1: $\rho_1(i) = \pi_i \varkappa_i(X_1)$, $1 \leqslant i \leqslant K$;

(2) At $t+1$, compute the path to $q_{i'}$ from the K possibilities ($s_t = i$; $i = 1, 2, ..., K$) at t with a summation over the product of all: $\rho_{t+1}(i') = \left[\sum_{i=1}^{K} \rho_t(i) b_{ii'}\right] \varkappa_{i'}(X_{t+1})$ for $t = 1, 2, ..., T-1$, $1 \leqslant i' \leqslant K$

(3) Lastly, calculate $P(X|\Lambda) = \sum_{i=1}^{K} \rho_T(i)$.

The computational complexity of the forward algorithm is K^2T; substantially lower than a direct calculation approach [48]. Likewise, the backward algorithm computes the probability of the partial sequence from $t+1$ to T. We begin by i of Λ at t [48]. This may be denoted with $\theta_t(i) = P(X_{t+1}, X_{t+2}, ..., X_T, s_t = i|\Lambda)$ and computed by:

(1) Calculate a random initialization $\theta_T(i) = 1$, $1 \leqslant i \leqslant K$;

(2) $\theta_t(i) = \sum_{i'=1}^{K} b_{ii'} \varkappa_{i'}(X_{t+1}) \theta_{t+1}(i')$ for $t = T-1, T-2, ..., 1$, $1 \leqslant i \leqslant K$

Jointly, through consequent iteration, the resultant forward-backward algorithm is employed for the computation of the probability of an observation X given Λ:

$$P(X|\Lambda) = \sum_{i=1}^{K} \sum_{i'=1}^{K} \rho_t(i) b_{ii'} \varkappa_{i'}(X_{t+1}) \theta_{t+1}(i') \tag{159}$$

6.2.2 Feature selection

We define whether a feature is relevant or not using a feature saliency technique. Feature saliency formulates the feature selection process as parameter estimation [155]. We incorporate feature saliencies; i.e., parameters, to the hidden variable model and to find clusters embedded in the feature subspace [158]. Mathematically, given a certain state, assume that each of the dimensions of the features is independent with latent indicator variable z_d, $z = (z_1, \ldots, z_D)$ of the component that the dth sequence belongs to, $z_d = (z_{d1}, \ldots, z_{dM})$ and each element z_{dj} is assigned value 1 when X_i is associated with component j; else, 0. Then:

$$\begin{aligned} p\left(X_t|z, s_t = i, \Lambda\right) \\ = \prod_{d=1}^{D} p\left(X_{dt}|\lambda_{id}\right)^{z_d} \text{Beta}\left(X_{dt}|\epsilon_d, \tau_d\right)^{1-z_d} \end{aligned} \tag{160}$$

where Beta is the conditional Beta distribution that is used to model irrelevant features and defined as:

$$\text{Beta}(X_{dt}|\epsilon_d, \tau_d) = \prod_{d=1}^{D} \frac{\Gamma(\epsilon_d + \tau_d)}{\Gamma(\epsilon)\Gamma(\tau_d)} X_d^{\epsilon_d - 1} \left(1 - X_d\right)^{\tau_d - 1} \tag{161}$$

The joint distribution of X_t and z given s is:

$$\begin{aligned} p\left(X_t, z|s_t = i, \Lambda\right) \\ = \prod_{d=1}^{D} \left[\varphi_d p\left(X_{dt}|\lambda_{id}\right)\right]^{z_d} \left[\left(1 - \varphi_d\right) \text{Beta}(X_{dt}|\epsilon_d, \tau_d)\right]^{1-z_d} \end{aligned} \tag{162}$$

where the marginal probability of z and X_t given s are given by:

$$P(z|\Lambda) = \prod_{d=1}^{D} \varphi_d^{z_d} \left(1 - \varphi_d\right)^{1-z_d} \tag{163}$$

$$\begin{aligned} p(X_t|\lambda_{s_t, m_t}) = \prod_{d=1}^{D} \left[\varphi_d p\left(X_{dt}|\lambda_{id}\right)\right] \left[\left(1 - \varphi_d\right)\right. \\ \left. \times \text{Beta}(X_{dt}|\epsilon_d, \tau_d)\right] \end{aligned} \tag{164}$$

respectively. This may then be used for the calculation of the complete data likelihood in Eq. (1) accordingly.

6.2.3 MAP approximation

We employ the EM technique to deploy the MAP GD HMMs with feature selection. Initially, $\gamma^{t}_{s_t,m_t} \triangleq p(s_t, m_t|Y_0, \ldots, Y_T)$ represents the estimate of the state and mixture component and $\eta^{t}_{s_t,s_{t+1}|Y_0,\ldots,Y_T} \triangleq p(s_t, s_{t+1}|Y_0, \ldots, Y_T)$ the estimate of the local states sequence given the complete sequence [2], where $Y = X$. The E-step of the forward-backward algorithm computes these for all time steps t. In other words, the lower bound of the data log-likelihood is used for its maximization [2]:

$$\begin{aligned}
\mathbb{L}(\Lambda|Y) &= p(Y|\Lambda) = \mathfrak{E}(Y, \Lambda) - \mathfrak{R}(\mathfrak{Z}) \\
&= \sum_{\mathfrak{Z}} p(\mathfrak{Z}|Y)\ln(p(Y, \mathfrak{Z})) - \sum_{\mathfrak{Z}} p(\mathfrak{Z}|Y)\ln(p(Z|Y)) \\
&= \sum_{\mathfrak{Z}} p(\mathfrak{Z}|Y)\ln(p(Y)) \\
&= \ln(p(Y)) \sum_{\mathfrak{Z}} p(\mathfrak{Z}|Y) = \ln(p(Y))
\end{aligned} \tag{165}$$

where hidden variables are denoted by $\mathfrak{Z}$, complete-data log-likelihood by $\mathfrak{E}(Y, \Lambda)$ with the maximized parameters, Λ, and log-likelihood of the hidden states given the sequence of observations by $\mathfrak{R}(\mathfrak{Z})$. $\mathfrak{R}(\mathfrak{Z})$ also quantifies the amount of information appended by the hidden data in the form of an entropy [2]. Hence, the expected complete-data log-likelihood is denoted by:

$$E(Y, \Lambda, \Lambda^{old}) = \sum_{\mathfrak{Z}} p(\mathfrak{Z}|Y, \Lambda^{old})\ln(p(Y, \mathfrak{Z}|\Lambda)) \tag{166}$$

where the non-optimized parameters complete data log-likelihood $\mathfrak{E}(Y, \Lambda, \Lambda^{old}) \leq \mathfrak{E}(Y, \Lambda)$; ergo, $\mathfrak{E}(Y, \Lambda, \Lambda^{old}) - \mathfrak{R}(\mathfrak{Z})$ represents the lower bound of the likelihood. That is analogous to Eq. (64) when expanded in which $p(Y_t|\varkappa_{s_t,m_t})$ is a GD mixture in this work. This ifs mathematically formulated as:

$$GD(Y|\vec{\iota}, \vec{\vartheta}) = \prod_{d=1}^{D} \frac{\Gamma(\iota_d + \vartheta_d)}{\Gamma(\iota_d)\Gamma(\vartheta_d)} Y_d^{\iota_d - 1} \left(1 - \sum_{r=1}^{d} Y_r\right)^{\zeta_d} \tag{167}$$

where $\vec{\iota} = (\iota_1, ..., \iota_D) and \vec{\vartheta} = (\vartheta_1, ..., \vartheta_D)$ are real and strictly positive parameters of the GD distribution and $Y \in \mathbb{R}_+^D$ where $\sum_{d=1}^{D} Y_d < 1$ for the $(D+1)$-dimensional proportional vector with a unit sum [51]. Finally, ζ_d is computed using the parameters of the distribution as $\vartheta_d - \iota_{d+1} - \vartheta_{d+1}$, when $d \neq D$. Otherwise, $\zeta_d = \vartheta_D - 1$. The complete data log-likelihood $\ln(p(Y, Z|\Lambda))$ can be formulated as:

$$\mathbb{Q}(\Lambda^t, \Lambda^{t-1}) = \mathbb{E}[\ln(p(Y, \mathfrak{Z}|\Lambda^t))|\mathfrak{Z}, \Lambda^{t-1}] \tag{168}$$

where Λ^t represents the HMM parameters for the current iteration while Λ^{t-1} is the set of parameters from the previous one. Extra terms are added to $\mathbb{Q}(\Lambda^t, \Lambda^{t-1})$ to integrate priors $\mathbb{A}(\Lambda^t)$ for the HMM parameters that distinguish the MAP approximation. The maximum likelihood technique is different; though it is the most widely used one for finding the optimal HMM parameters. It is known as the Baum Welch algorithm where Eq. (168) would be sufficient. Consequently, the updated formula is expressed as:

$$\mathbb{S} = \mathbb{Q}(\Lambda^t, \Lambda^{t-1}) + \ln(\mathbb{A}(\Lambda^t)) \tag{169}$$

It is noteworthy to mention that the forward and backward probabilities are also required for the incorporation of the simultaneous selection of features. The saliencies of the features are computed in the E-step:

$$\begin{aligned} e_{idt} &= p\left(Y_{dt}, z_d = 1|s_t = i, \Lambda^{t-1}\right) \\ &= \varphi_d \, p\left(Y_{dt}|\lambda_{id}\right) \end{aligned} \tag{170}$$

$$\begin{aligned} h_{idt} &= p\left(Y_{dt}, z_d = 0|s_t = i, \Lambda^{t-1}\right) \\ &= (1 - \varphi_d) \, \text{Beta}\left(Y_{dt}|\epsilon_d, \tau_d\right) \end{aligned} \tag{171}$$

$$\begin{aligned} g_{idt} &= p\left(Y_{dt}|s_t = i, \Lambda^{t-1}\right) \\ &= e_{idt} + h_{idt} \end{aligned} \tag{172}$$

$$\begin{aligned}\alpha_{idt} &= p\left(z_d = 1, s_t = i | Y, \Lambda^{t-1}\right) \\ &= \frac{\gamma_i^t e_{idt}}{g_{idt}}\end{aligned} \tag{173}$$

$$\begin{aligned}\beta_{idt} &= p\left(z_d = 0, s_t = i | Y, \Lambda^{t-1}\right) \\ &= \frac{\gamma_i^t h_{ilt}}{g_{ilt}} \\ &= \gamma_i^t - \alpha_{ilt}\end{aligned} \tag{174}$$

where γ_i^t, $\eta_{i,i'}^t$, α_{idt}, and β_{idt} are essential in the computations of the M-step.

We employ the Dirichlet distribution as the determined suitable priors for the parameters of the HMM. This corresponds to the strict positive nature of the coefficients of π, B, and C, with values < 1 that sum to one for each row:

$$\begin{aligned}p(\pi) &= \text{Dirichlet}(\pi|\phi^\pi) \\ &= \text{Dirichlet}(\pi_1, ..., \pi_K|\phi_1^\pi, ..., \phi_K^\pi), \\ p(B) &= \prod_{i=1}^{K} \text{Dirichlet}(b_{i_1}, ..., b_{i_K}|\phi_{i_1}^B, ..., \phi_{i_K}^B), \\ p(C) &= \prod_{i=1}^{M} \text{Dirichlet}(c_{i_1}, ..., c_{i_M}|\phi_{i_1}^C, ..., \phi_{i_M}^C)\end{aligned} \tag{175}$$

Ergo, MAP estimated variables are updated in the M-step as follows:

$$\pi_i = \frac{\gamma_i^0 + \phi_i^\pi - 1}{\sum_{i=1}^{K}(\gamma_i^0 + \phi_i^\pi - 1)} \tag{176}$$

$$B_{ii'} = \frac{\sum_{t=1}^{T} \eta_{i,i'}^t + \phi_{i'_i}^B - 1}{\sum_{i=1}^{K}(\sum_{t=1}^{T} \eta_{i,i'}^t + \phi_{i'_i}^B - 1)} \tag{177}$$

$$C_{ij} = \frac{\sum_{t=1}^{T} \gamma_{i,j}^t + \phi_{ij}^C - 1}{\sum_{j=1}^{M}(\sum_{t=1}^{T} \gamma_{i,j}^t + \phi_{ij}^C - 1)} \tag{178}$$

Similarly, conjugate priors must be defined across the GD parameters. The Gamma distribution $\mathcal{G}(.)$ is an appropriate choice as a prior [92]. Consequently, the assigned priors over the parameters

of the distribution are:

$$p(\{\vec{\iota}\}_{i,j,d=1}^{K,M,D}) = \prod_{i=1}^{K}\prod_{j=1}^{M}\prod_{d=1}^{D}\mathcal{G}(\iota_{ijd}|\varrho_{ijd},\zeta_{ijd}), \tag{179}$$

$$p(\{\vec{\vartheta}\}_{i,j,d=1}^{K,M,D}) = \prod_{i=1}^{K}\prod_{j=1}^{M}\prod_{d=1}^{D}\mathcal{G}(\vartheta_{ijd}|u_{ijd},v_{ijd}) \tag{180}$$

where the hyperparameters ϱ, ζ, u, and v are strictly positive.

The update equations for these parameters employ the Newton-Raphson estimation approach for the lower bound approximation. This methodology obeys:

$$\theta^{new} = \theta^{old} - \mathbb{H}(\theta^{old})^{-1}\frac{\partial\mathbb{L}(Y|\theta^{old})}{\partial\theta^{old}} \tag{181}$$

where $\mathbb{H}(.)$ is the Hessian matrix which constitutes of the second order derivatives of the likelihood function; corresponding to θ in the earlier equation. We do not detail the calculation of this formula and its sub-parts in this paper; however, we refer the reader to [149]. The complete graphical model of the proposed framework is shown in Fig. 6.2.

6.2.4 Complete Algorithm

In this article, the convergence is traced systematically through monitoring the update difference in the estimated parameters of Λ. This is set with an adaptive threshold which we have set at 10^{-3} between the iterations or reaching a maximum number of iterations set at 300. The complete algorithm is detailed in Algorithm 1.

6.3 Experimental Results

6.3.1 Categorization of dynamic textures

We validate the proposed MAP with simultaneous feature selection model on the benchmarking *Alpha DynTex* dataset for dynamic texture classification [159, 160, 133]. The dataset contains sixty sequences, split equally across three classes: trees, grass, and sea. Samples of the classes are shown

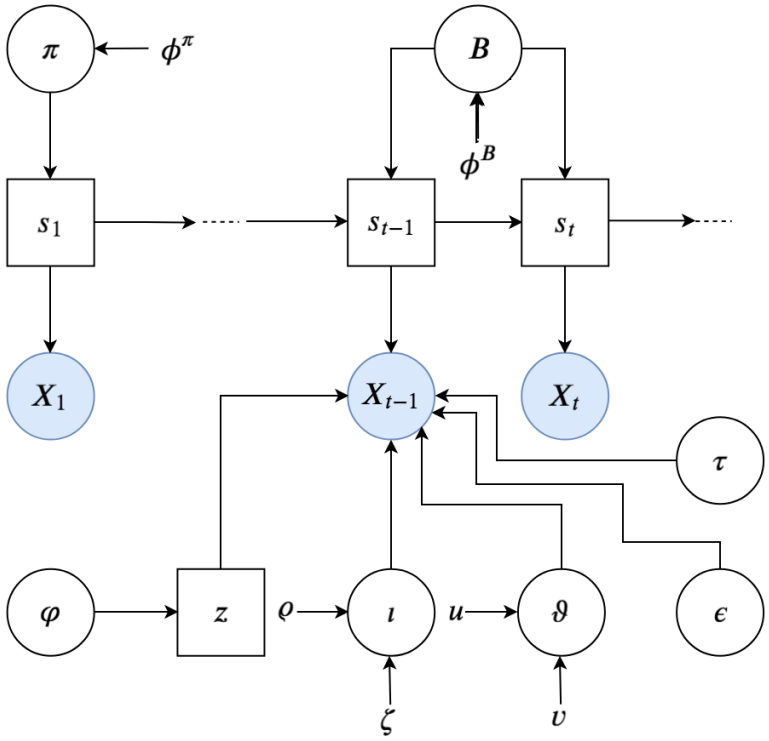


Figure 6.2: Graphical model of the proposed MAP GD HMM with simultaneous feature selection. Circles represent model parameters and filled ones are observed variables. Squares represent hidden variables.

Algorithm 1: Proposed algorithm.

Result: MAP trained HMM model for a particular class
Initialize priors $\mathbb{A}(\Lambda^t)$, K, M, π, B, C, $\varkappa$;
while *convergence NOT reached* **do**
  Execute forward backward algorithm;
  if *current dimension* $d \leq D$ **then**
    Execute emission parameter update for feature d;
    Compute feature d saliency;
    $d = d + 1$;
  else
    return Λ;
  end
end

in Fig. 4.1. We train the GD HMMs on the dataset with Local Binary Pattern (LBP) features as the input series. These are popular and effective in texture recognition applications [134].

We deploy a leave-one-out cross validation scheme on these features for training and testing. Each class has an independent HMM trained. In the testing stage, for each of the HMMs, the likelihood of each video is computed. The final label is appointed according to the highest likelihood. We experimentally set the number of states and components of the mixture to two. We benchmark with Baum Welch trained Gaussian HMM and the latest proposed approach for proportional HMMs. We also compare with Baum Welch and MAP learning of a traditional proportional HMM that uses the Dirichlet distribution.

Table 6.2: Models' accuracy for dynamic texture categorization. The proposed framework is highlighted in bold.

HMM	**Accuracy (%)**
Baum Welch trained - Gaussian	50.00
Baum Welch trained - Dirichlet	85.00
Baum Welch trained - GD	86.67
Hybrid generative descriminative - GD	73.33
MAP trained - Dirichlet	90.00
MAP trained - GD	90.00
Variational inference trained - BL	91.76
MAP trained with feature selection - GD	93.33
DFS [150]	83.60
2D+T [161]	85.00
OTDL [162]	86.60
LBP-TOP [119]	86.67
MBSIF-TOP [138]	90.00
ASF-TOP [163]	91.67
MPCAF-TOP [164]	96.67

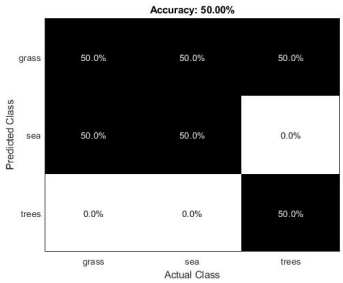
To evaluate the proposed approach, we calculate the accuracy of the models. Accuracy may be computed in correspondence to the number of correctly recognized dynamic textures. Mathematically, that is $\mathrm{TP}/(\mathrm{TP}+\mathrm{TN})$ where TP represents the number of true positives and TN denotes the number of true negatives correctly identified by the approach. The results of the metric are shown

in Table 6.2. As expected, the MAP HMM outperforms the Baum Welch [2]. It also performs better than the hybrid model that combine proportional HMMs with support vector machines (SVM) in a generative-discriminative approach.

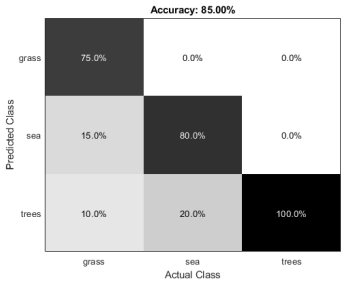
The MAP trained HMM achieves such results at a relatively low computational cost in terms of addition of a prior and utilizing less complex derivations than other widely applied approximation methods such as variational inference. This supports its broad applicability. Additionally, utilizing the GD distribution improves the results in comparison to the employment of the Gaussian distribution for the emission probability of the HMMs. This is because of its improved ability to capture the underlying pattern of the data. Finally, the best performing model is the proposed MAP GD with a simultaneous feature selection framework given its incorporated ability to model relevant features. A breakdown of the exact accuracy results for each of the classes is presented for each of the methods in a confusion matrix form. These may be observed in Fig. 6.3.

One of the latest sophisticated proposed HMM methods for proportional data modeling is the variational inference based Beta Liouville (BL) HMM [46]. The BL HMM has reported higher accuracy results than the proposed MAP HMM. This is due to two reasons. The first is due to the variational inference procedure that is used for the training of the BL HMM. Such a method approximates the lower bound of the marginal likelihood; hence, allowing for a better approximation of the data. However, this comes at the expense of extra computational power that is needed, which is justified by the extra mathematical operations that are required for the construction of the model. These details are fully described in [46]. The second advantage that the variational based BL HMM has is the use of the BL distribution which has better modeling capabilities than the GD distribution. In particular, it requires a lower number of parameters for a similar representation potential of the data. That is, it also surmounts the Dirichlet's restriction for modeling of proportional data; i.e., negative data covariance. Nonetheless, the proposed method with the simultaneous feature selection paradigm still outperforms the variational inference trained BL HMM. It is then important to mention that this proposed simultaneous feature selection may only be included with the GD distribution given its factorization characteristics. This also further motivates our choice of distribution for the proposed approach.

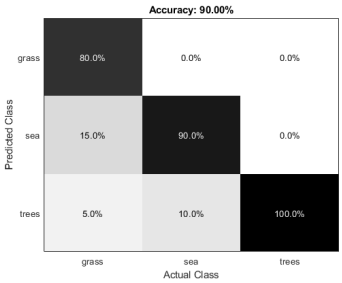
We also compare to several other methods in the literature, particularly ones where the feature



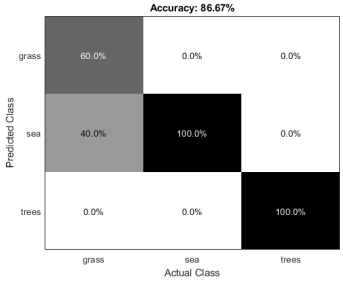
(a) Baum Welch - Gaussian HMM.



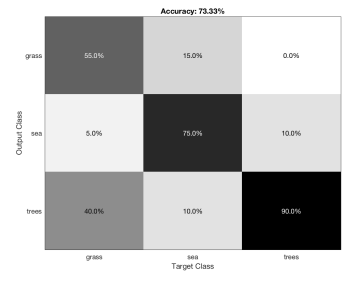
(b) Baum Welch - Dirichlet HMM.



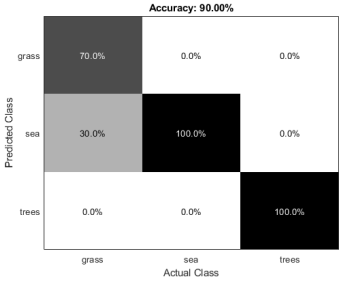
(c) MAP - Dirichlet HMM.



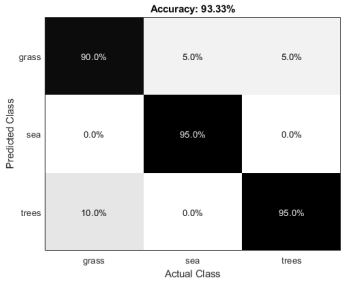
(d) Baum Welch - GD HMM.



(e) Hybrid GD HMM/SVM.



(f) MAP - GD HMM.



(g) MAP - GD HMM with feature selection.

Figure 6.3: Dynamic texture classification confusion matrices.

extraction process is dependent on three orthogonal planes [160]. These are evaluated with the nearest class center (NCC) classifier with the Chi-square distance or the nearest neighbor (NN) classifier. The methods comprise of DFS [150], 2D+T [161], OTDL [162], LBP-TOP [119], MBSIF-TOP [138], ASF-TOP [163], and MPCAF-TOP [164]; with an interested reader referred to the original papers for their respective descriptions as such discussions are outside the scope of the presented article. Overall, the proposed HMM achieves comparable accuracy results to the other methods. It is also noteworthy to reiterate that the main aim of our method is to propose a model that best fits the statistical properties of the input data, especially with regards to dynamic ones. Nonetheless, the latter methods are presented for a more inclusive comparative experimental evaluation as well as to offer another potential aspect for future investigation for further improvement of the HMM method.

6.3.2 Recognition of infrared actions

Here, we present our experimental results on the *InfAR* [98] action recognition (AR) infrared (IR) dataset to validate our proposed model [165, 166]. This constitutes of single person action training and testing sets. This is actuated by 7 classes of ten videos each that we deploy with leave-one-out cross validation. Sample depictions for each of the classes are shown in Fig. 3.2. A series of histogram of optical flow (HOF) and motion boundary histogram (MBH) are extracted to represent each of the sequences.

For the HOF, the orientations are quantized into 9 bins and normalized with the L_2 norm. Derivatives of the optical flow are evaluated separately along the horizontal (MBHx) and vertical (MBHy) components to compute the MBH. The latter effectively captures relative motion between pixels and suppresses constant motion information to mute noise from background motion. The HOF and MBH descriptors may be extracted using any interest point detector [100]. We extract the points along the motion trajectory for both the training and the testing video sequences [101].

We deploy a similar experimental setup as in the application, as well as for the Baum Welch and the MAP trained Dirichlet HMMs for comparison. Our comparative results of the features with the Baum Welch and MAP trained HMMs can be seen in Fig. 6.5 with the computed accuracy placed on the top center of each of the subfigures. The results can be seen in Fig. 6.6. A graph of the average precision (AP) results is also shown in Fig. 6.4.

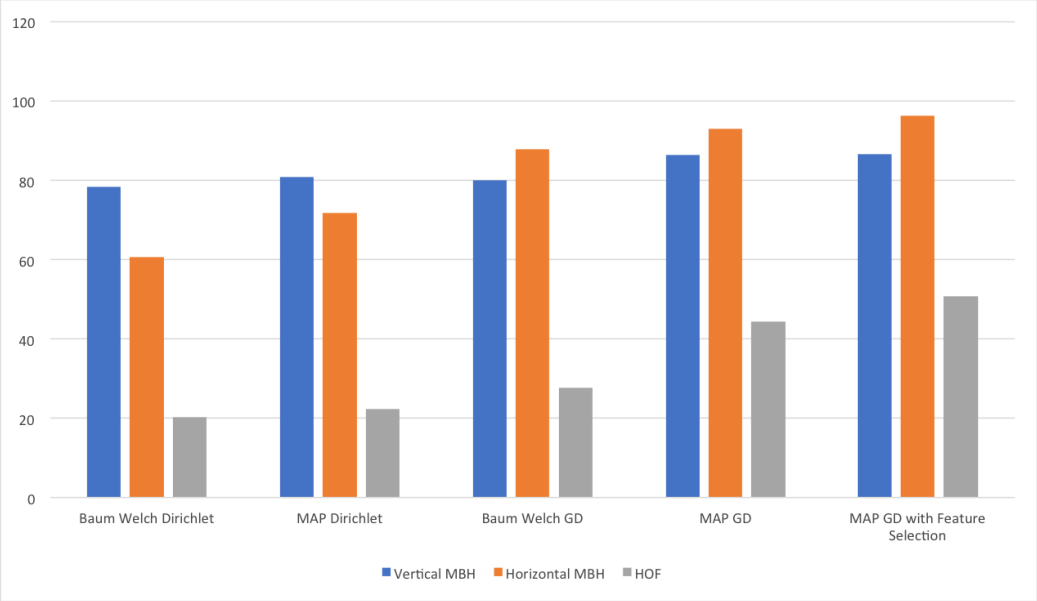


Figure 6.4: A contrast of HMMs for the recognition of infrared actions. The approaches are trained with the same features used in the proposed algorithms. The models names are displayed over the horizontal axis while AP (in $\%$) are depicted across the vertical axis.

It is apparent that the MBH features along both axes are more suitable for the proposed HMMs over the different setups given the accuracy results. Utilizing feature selection improves performance across the various features. Moreover, the MAP approximated HMMs expectedly have a better performance in comparison to Baum Welch trained. Moreover, the results are influenced by utilizing the more flexible GD distribution becuase of its improved ability to capture the underlying patterns of proportional data than the Dirichlet distribution. The increase in the accuracy also reflects that the restrictions that the Dirichlet requires is not an inherent property of the data at hand.

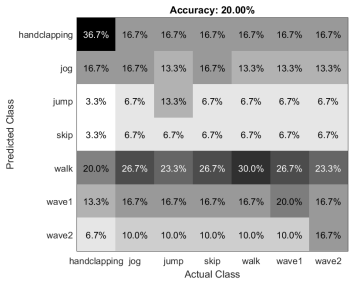
In the state-of-the-art, these results are comparable to multiple methods. These include different handcrafted features extracted for the InfAR dataset; for instance, HOF [98], dense trajectories [104], and improved dense trajectories [105], the two-stream 3D convolutional neural network (CNN) [102], the optical flow field 3D CNN [102], and the three-stream trajectory-pooled deep-convolutional descriptors technique in [103]. A comparison of the achieved results with the proposed models is shown in Fig. 6.8. Furthermore, we also compare to the variational inference based BL HMM whose results are shown in Fig. 6.7. Though the latter outperforms the proposed method in this task; it is important to mention that our main aim is to propose a model with the finest fit

to the data characteristics. As with any machine learning method, data rules whether a particular approach outperforms another and this is again proven in regards to this particular dataset and task. However, the proposed approach still represents a novel method for the modeling of proportional data with HMMs.

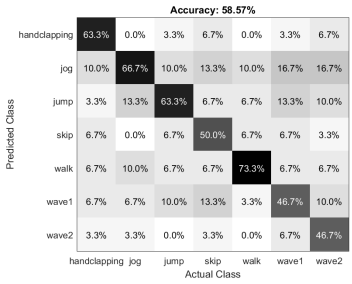
6.4 Conclusion

A cornerstone in time series and sequential data modeling is the HMM. Its effective estimation of parameters and emission distribution choice are significant challenges to be tackled in the research and employment of HMMs. We focus on proposing a method based on MAP with simultaneous feature selection for efficient estimation of proportional HMMs; specifically, GD-based HMMs. Applying the MAP technique is superior over the commonly-used Baum Welch approach in terms of better performance sans the computational cost. In contrast, the concurrent feature selection algorithm allows us to seamlessly assign weights to the various input features to better model the data with the best representation and a reduced overhead in terms of the number of features utilized. For validation of the developed models, we apply the proposed approach in classification of dynamic textures and recognition of infrared actions. We achieve comparable results with several relevant approaches and state-of-the-art methods. Performance enhancement distinctly underscores the importance of deriving and applying the MAP approximation with feature selection and the choice of the distribution corresponding to the data support. Future works may include the consideration of Hierarchical Dirichlet processes as well as considering a feature selection technique with variational inference of the proposed model. Furthermore, we plan to investigate the ubiquitous deep learning approaches as an incorporation into the framework of HMMs for further improvements of the performance.

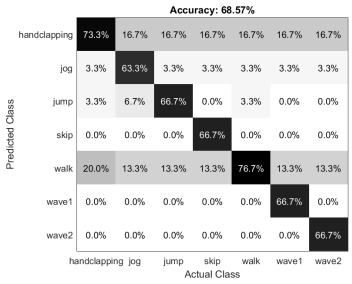
The proposed approach can improve performance of HMMs across different fields. For instance, it may be applied to HMMs utilized to forecast weather [28] or to detect fraud in bank transactions [31]. It can also be applied for the training of HMMs on gesture recognition for artificially intelligent cockpit control [32]. It may also be incorporated into smart city applications which are highly dependent on Internet of Things (IoT) technologies. For example, a methodology is developed in



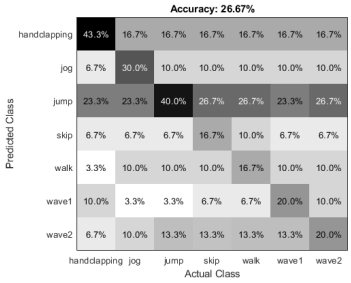
(a) Dirichlet HMM - HOF.



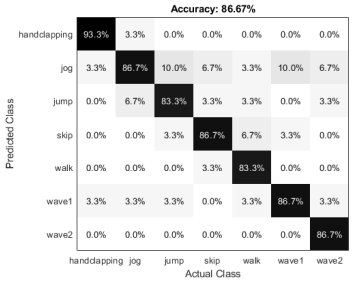
(b) Dirichlet HMM - horizontal MBH.



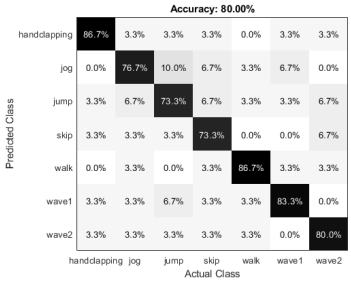
(c) Dirichlet HMM - vertical MBH.



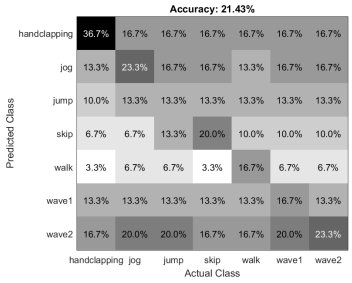
(d) GD HMM - HOF.



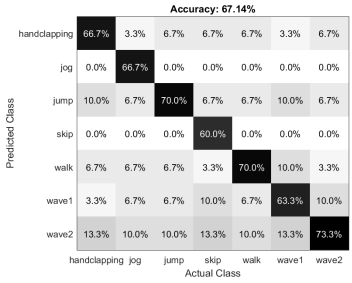
(e) GD HMM - horizontal MBH.



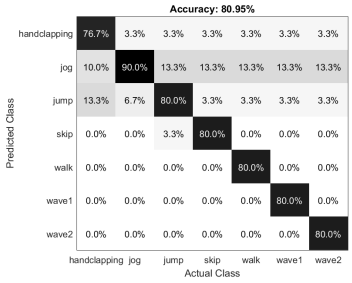
(f) GD HMM - vertical MBH.



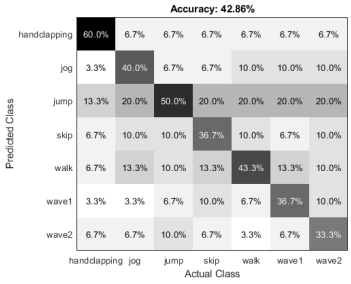
(g) Dirichlet HMM - HOF.



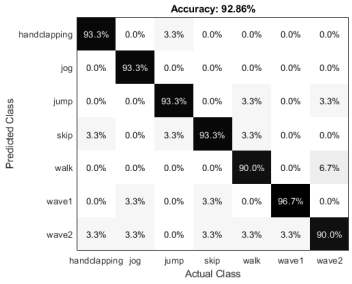
(h) Dirichlet HMM - horizontal MBH.



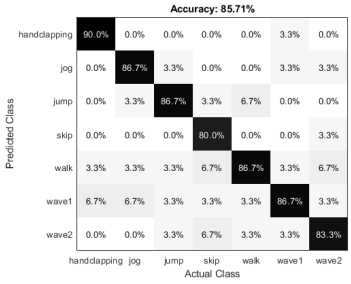
(i) Dirichlet HMM - vertical MBH.



(j) GD HMM - HOF.

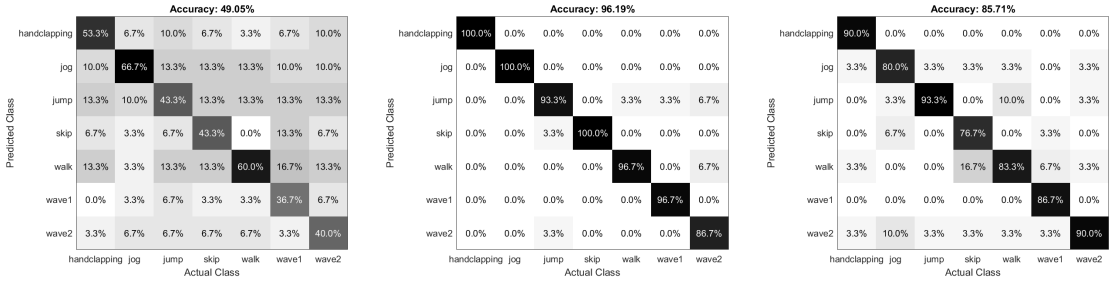


(k) GD HMM - horizontal MBH.



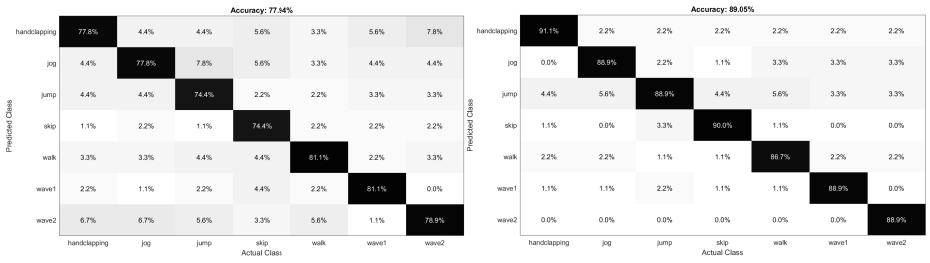
(l) GD HMM - vertical MBH.

Figure 6.5: IR AR confusion matrices of the trained HMMs. (a)-(f) are approximated with the Baum Welch method, while (g)-(l) are trained by the MAP technique presented.

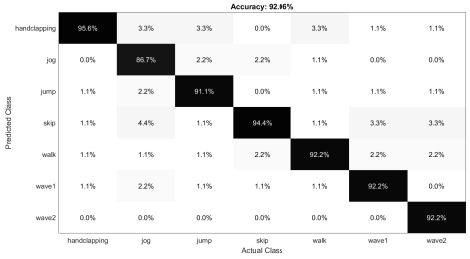


(a) GD HMM - HOF. (b) GD HMM - horizontal MBH. (c) GD HMM - vertical MBH.

Figure 6.6: IR AR confusion matrices of the Generalized Dirichlet (GD) HMMs estimated by the proposed MAP framework with simultaneous feature selection.



(a) BL HMM - HOF. (b) BL HMM - horizontal MBH.



(c) BL HMM - vertical MBH.

Figure 6.7: IR AR confusion matrices of the Beta Liouville (BL) HMMs approximated with variational inference.

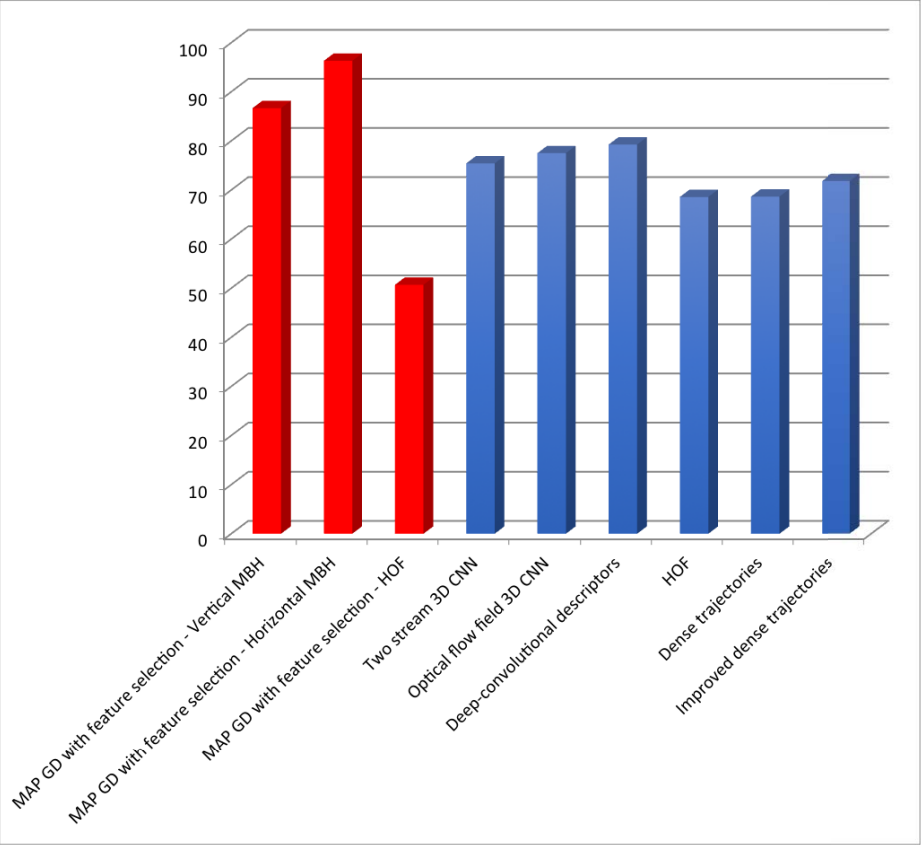


Figure 6.8: A contrast of proposed HMMs (red) for IR AR application on the InfAR dataset with other state-of-the-art methodologies (blue) referenced in the article. Model names are displayed over the horizontal axis and AP (in $\%$) are depicted across the vertical axis.

[34] for efficient power usage of IoT devices with HMMs, while the latter is used for addressing the detection of IoT power signature anomalies in [35]. Improvement for text to speech language models also serves as another application which the proposed model can further improve [40].

Chapter 7

Infinite Dirichlet and Beta Liouville Hidden Markov Model

> If you can DREAM it, you can do it.
>
> Walt Disney

In this chapter, we investigate the infinite extension of the Dirichlet and Beta-Liouville hidden Markov models (HMM) for proportional data. This work now addresses another computer vision based problem; i.e., anomaly detection for surveillance.

7.1 Introduction

Nowadays, the overwhelming need for data modeling is at an all time high. This is mainly driven by the vast amount of data, around 2.5 quintilian bytes, that is generated daily [167]. Modeling such data has several advantages. These include the potential removal of noise from the original signals, studying the data source or the generating process without the need of its explicit availability, and efficiently building powerful recognition and prediction systems [3]. Moreover, a consequent implication of successfully formulating a model for a data source is the ability to simulate as much data as needed to further study such a phenomenon without the need to carry out costly experiments.

Meanwhile, the prevalence of cameras and video surveillance has led to a significant increase in research interest in the field of automatic real time monitoring systems [146, 168, 147, 169]. Such

systems aim to assist human operators in the detection of potential threats; especially due to the need to oversee multiple feeds simultaneously for long periods of time. Hence, efficient automatic video monitoring plays an impertinent role in the detection of malicious threats and the avoidance of consequent incidents. While testing and development of these systems have been facilitated by the recent release of real-world data sets [170, 171], an anomaly, i.e. a threat, by its nature is a rare occurrence that is therefore seldom recorded and hence sufficient data for automatic detection training is unavailable [29].

A typical methodology to overcome this hindrance in the development of video anomaly detection systems is training on video sequences; i.e., data that is considered normal. This enables the system to report anomalies as outliers [146, 172, 147]. Indeed, this is especially important for the design and training of appropriate machine learning techniques. Indeed, this forms the basis of multiple approaches that are paving the way for new advancements in anomaly video detection [173].

In this paper, we propose to use variational learning of infinite HMM for the modeling of anomalies. A HMM is a probabilistic model that is appropriate for dynamic data [174, 175]. It is usually trained with the Baum-Welch method; a variation of the Expectation Maximization (EM) algorithm specially designed for HMM. However, employing a variational inference technique is advantageous due to the various drawbacks of the Baum-Welch method [176, 177]. The latter include over-fitting or sub-optimal generalization performance [46].

Furthermore, a primary area of HMM research lies in modeling emission probabilities of proportional data. They naturally result from numerous preprocessing procedures; such as the commonly used histograms. Applying a Gaussian-based HMM in such a case is not ideal. Indeed, recent studies have proven success utilizing distributions that correspond to the proportional nature of the data such as the Dirichlet [94, 48] and the Beta-Liouville (BL) [46] distributions.

Also, choosing the correct number of states to model the data with HMMs is another area that warrants further investigations. The number of states is usually determined as a result of an exhaustive search for the appropriate count. Nonetheless, we are inspired by works in the literature that extend the structure of HMMs to infinity [178]. Implementation of infinite HMMs can be achieved through the means of non-parametric Bayesian methods [179].

The novel contributions of this work can be summarized as:

- We propose a mathematical formulation for construction of infinite HMMs for continuous proportional data. We particularly focus on the use of Dirichlet and BL distributions given their proven effectiveness in modeling proportional sequential data. In comparison to deep learning techniques, this is an explainable approach that requires less training data and less computationally expensive.
- We present a variational inference learning approach for the proposed infinite HMMs. The technique guarantees convergence within a reasonable time frame and provides an accurate approximation of the posterior.
- We propose an end-to-end framework for realtime robust anomaly detection in videos. The framework performs competitively with state-of-the-art relevant methods and may be applied to any video data.

7.2 Infinite Hidden Markov Models

A HMM is characterized by an underlying stochastic process with K hidden states, each governed by an initial probability π, and the transition between the states $B = \{b_{ii'} = P(s_t = i'|s_{t-1} = i)\}$ at time t. In each state s_t, an observation is emitted corresponding to its respective parameters of a probability distribution $\varkappa$ with a mixing matrix $C = \{c_{ij} = P(m_t = j|s_t = i)\}$ where $j \in [1, M]$ such that M is the number of mixture components in set $L = \{m_1, ..., m_M\}$. Hence, a continuous HMM may be defined as $\lambda = \{B, C, \varkappa, \pi\}$. A graphical model of the latter HMM is depicted in Fig. 7.1.

The likelihood of a sequence with HMMs may be denoted by:

$$p(X|B, C, \varkappa, \pi) = \sum_{S} \sum_{L} \pi_{s_1} \left[\prod_{t=2}^{T} b_{s_{t-1}, s_t} \right] \left[\prod_{t=1}^{T} c_{s_t, m_t} p(X_t|\Lambda_{s_t, m_t}) \right] \tag{182}$$

where $\Lambda_{ij} = (\Lambda_{1ij}, ..., \Lambda_{Dij})$ with $\varkappa$ defined according to the Dirichlet and BL distributions for proportional data in this work. For simplification purposes, we derive the model for a unique sequence. A summation over sequences may then be added for inclusion of more sequences. Indeed,

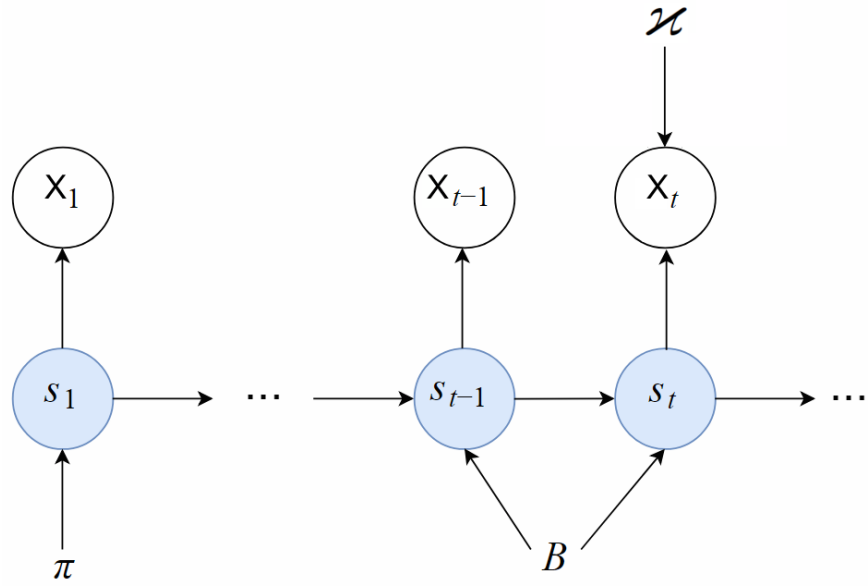


Figure 7.1: Graphical model representation of a continuous hidden Markov model. Symbols in unshaded circles denote the observed variables, symbols in shaded circles indicate the hidden states, and edges represent conditional dependencies between the states or the variables.

this is usually the case to prevent overfitting.

Formally, a D-dimensional Dirichlet distribution is denoted by:

$$DR(X|\vec{\varepsilon}) = \frac{\Gamma(\sum_{d=1}^{D} \varepsilon_d)}{\prod_{d=1}^{D} \Gamma(\varepsilon_d)} \prod_{d=1}^{D} x_d^{\varepsilon_d - 1} \tag{183}$$

where $\vec{\varepsilon} = (\varepsilon_1, ..., \varepsilon_D)$ is the real and strictly positive parameter of the distribution and $X \in \mathbb{R}_+^D$, $\sum_{d=1}^{D} x_d = 1$ corresponding to the D-dimension proportional vector that adds up to one. Consequently, the complete data log-likelihood with the Dirichlet mixture may be split with the logarithm sum-product property as follows:

$$\begin{aligned} \ln(p(X, Z|\lambda)) = \sum_{t=2}^{T} \ln(b_{s_{t-1}, s_t}) + \sum_{t=1}^{T} \ln(c_{s_t, m_t}) + \ln(\pi_{s_1}) + \sum_{t=1}^{T} \Bigg[\sum_{d=1}^{D} [\ln(x_d) + \\ \Psi(\sum_{d=1}^{D} \varepsilon_d) - \Psi(\varepsilon_d) - \ln(\sum_{d=1}^{D} x_d) \Bigg] \Bigg] \end{aligned} \tag{184}$$

A better model of proportional time series data has been proposed with the BL distribution in [46]. This distribution is closely related to the Dirichlet, but it relaxes the constraint of negative covariance

at the cost of two additional parameters. As a matter of fact, the Dirichlet distribution is a special case of the BL distribution. The latter is expressed by:

$$BL(X|\vec{\kappa},\alpha,\beta) = \frac{\Gamma(\sum_{d=1}^{D}\kappa_d)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\prod_{d=1}^{D}\frac{x_d^{\kappa_d-1}}{\Gamma(\kappa_d)}\left(\sum_{d=1}^{D}x_d\right)^{\alpha-\sum_{d=1}^{D}\alpha_d} \times \left(1-\sum_{d=1}^{D}x_d\right)^{\beta-1} \tag{185}$$

where $\vec{\kappa} = (\kappa_1, ..., \kappa_D)$, α, and β are real and strictly positive parameters of the BL distribution, $\Gamma(t) = \int_0^\infty X^{t-1}e^{-X}dX$ is the Gamma function, and X is a $D+1$ dimensional vector whereby $X \in \mathbb{R}_+^D$ and $\sum_{d=1}^{D} x_d < 1$. In this case, the complete data log-likelihood is expanded as:

$$\begin{aligned}\ln(p(X,Z|\lambda)) = &\sum_{t=2}^{T}\ln(b_{s_{t-1},s_t}) + \sum_{t=1}^{T}\ln(c_{s_t,m_t}) \\ &+ \ln(\pi_{s_1}) + \sum_{t=1}^{T}\Bigg[\Psi(\sum_{d=1}^{D}\kappa_d) + \Psi(\alpha+\beta) - \\ &\Psi(\alpha) - \Psi(\beta) + (\alpha - \sum_{d=1}^{D}\kappa_d)\ln(\sum_{d=1}^{D}x_d)) + \\ &(\beta-1)\ln(1-\sum_{d=1}^{D}x_d)) + \sum_{d=1}^{D}[(\kappa_d-1)\times \\ &\ln(x_d)) - \Psi(\kappa_d)]]\end{aligned} \tag{186}$$

7.2.1 The Dirichlet and stick-breaking processes

The DP is a parameterized stochastic process with a positive scalar precision ϵ and base distribution G_0 [180]. The DP forms a distribution over discrete distributions that place its mass on a countably infinite collection of atoms. The base distribution places location of atoms and the concentration variable controls the range of the mass spreading around atoms. This may be expressed for disjoint sets of $A = \{A_1, ..., A_D\}$ in measurable space Θ, where $\cup_i A_i = \Theta$, as:

$$(G(A_1), \ldots, G(A_D)) \sim \mathrm{DR}(\epsilon G_0(A_1), \ldots, \epsilon G_0(A_D)) \tag{187}$$

The DP has infinite dimensions, $D \to \infty$, as G_0 is a continuous distribution. G represents a draw from the DP that is denoted by $G \sim DP(\epsilon G_0)$ and may be written as:

$$G = \sum_{i=1}^{\infty} p_i \delta_{\theta_i} \tag{188}$$

where θ_i is a location drawn from G_0 and is associated with measure p_i. θ_i may then be interpreted as the emission distribution of an HMM at state i. A draw from the DP is then defined with a stick-breaking process as [181]:

$$\begin{aligned} &G = \textstyle\sum_{i=1}^{\infty} p_i \delta_{\theta_i}, \quad p_i = V_i \prod_{i'=1}^{i-1} (1 - V_{i'}) \\ &V_i \sim \text{Beta}(1, \epsilon), \quad \theta_i \sim G_0 \end{aligned} \tag{189}$$

where the influence of ϵ on a draw from DP is clearly impertinent. As $\epsilon \to 0$, a random component with location drawn from G_0 depicts a degenerate measure due to the breaking of the entire stick and consequent allocation to a single component. On the other hand, infinitely small breaks and convergence of G to the distribution of the draws occur when $\epsilon \to \infty$ so that G_0 itself is reproduced.

This may be conversely expressed as a DP mixture model with the following generative process when the interest is shifted to the parameters of a distribution rather than the data itself:

$$x_i \left| \theta_i \sim p\left(x | \theta_i\right), \quad \theta_i \right| G \sim G, \quad G | \epsilon G_0 \sim DP\left(\epsilon G_0\right) \tag{190}$$

HMMs are a special case of the mixture models which are state-dependent. That is each mixture has different weights but the same support. Consequently, we can define $\theta_i \equiv (\Lambda_{ij}, ..., \Lambda_{iM})$ and express the state-dependent mixture model of a continuous proportional HMM as:

$$x_t \left| \theta_{s_t} \sim Dist\left(\theta_{s_t}\right), \theta_{s_t} \right| s_{t-1} \sim G_{s_{t-1}}, G_i = \sum_{i'=1}^{D} b_{ii'} \delta_{\theta_{i'}} \tag{191}$$

where $Dist$ is defined according to the appropriate distribution to be applied. In this paper, this is either the Dirichlet or the BL distributions and the initial state has been assumed to be chosen from π. For an infinite HMM, it may then be inferred that each transition should be modeled as a DP. Nonetheless, an issue arises with this approach. Particularly, assume that each row i is drawn for

the infinite state transition matrix with:

$$\begin{aligned} &G_i = \textstyle\sum_{i'=1}^{\infty} b_{ii'}\delta_{\theta_{ii'}}, \quad b_{ii'} = V_{ii'}\prod_{f=1}^{i'-1}(1-V_{if}) \\ &V_{ii'} \sim \text{Beta}(1,\epsilon), \quad \theta_{ii'} \sim G_0 \end{aligned} \tag{192}$$

where $b_{ii'}$ is the i'th component of the infinite vector b_i. Note then that for each state indexing $\theta_{ii'}$, the probability of a transition to a previously visited state is zero when G_0 is continuous. This is because $p(\theta_\varsigma = \theta_\varrho) = 0$ for $\varsigma \neq \varrho$. This indicates the impracticality of such an approach for formulating an infinite HMM.

An extension of the DP is the hierarchical DP (HDP) which has been proposed to resolve such a problem. Indeed, the HDP is a two level approach whereby the base distribution is itself drawn from a DP resulting in an almost decidedly discrete G_0 [182]:

$$G_\varpi \sim DP\left(\Upsilon G_0\right), \quad G_0 \sim DP(\epsilon H) \tag{193}$$

Hence, substantial weight on the same set of states is impacted by multiple draws. If we truncate at K and write the top level DP in a stick-breaking form, we may then explicitly denote the second level DP as:

$$G_0 = \sum_{i=1}^{K} p_i \delta_{\boldsymbol{\theta}_i}, \quad p_i = V_i \prod_{i'=1}^{i-1}(1-V_{i'}) \tag{194}$$

$$V_i \sim \text{Beta}(1,\epsilon), \quad \theta_i \sim H \tag{195}$$

$$\begin{gathered} \left(G_\varpi(\theta_1), G_\varpi(\theta_2), \ldots, G_\varpi(\theta_K)\right) \sim \\ \text{DR}\left(\Upsilon p_1, \Upsilon p_2, \ldots, \Upsilon p_K\right) \end{gathered} \tag{196}$$

where $G(\theta_i)$ is a probability at location θ_i. However, the lack of conjugacy between the two levels (number of states and their emission parameters at the top level and the mixing weights as priors to draw the transition probabilities at the second level) means that a true variational solution does not exist [179]. Consequently, we utilize the priors of the form:

$$p_i = V_i \textstyle\prod_{i'=1}^{i-1}(1-V_{i'}), \quad V_i \sim \text{Beta}\left(\upsilon_i, \omega_i\right) \tag{197}$$

This formulation has a more flexible parametrization where the weights and locations are effectively detached. Though this stick-breaking process is infinite, v and ω terminate at a finite number K with $p_{K+1} \equiv 1 - \sum_{i=1}^{K} p_i$, so that the result is a draw from the Generalized Dirichlet (GD) distribution. This is necessary for the variational learning approach as discussed earlier. Hence, we must utilize a GD prior for the state transitions. Formally, for $\boldsymbol{V} = (V_1, ..., V_K)$:

$$F(\boldsymbol{V}) = \prod_{i=1}^{K} F(V_i) = \prod_{i=1}^{K} \frac{\Gamma(v_i + \omega_i)}{\Gamma(v_i)\Gamma(\omega_i)} V_i^{v_i - 1} (1 - V_i)^{\omega_i - 1} \tag{198}$$

The density of p may then be derived with a change of variables from $\boldsymbol{V}$:

$$\begin{aligned} &F(p) = \textstyle\prod_{i=1}^{K} \left(\frac{\Gamma(v_i+\omega_i)}{\Gamma(v_i)\Gamma(\omega_i)} p_i^{v_i - 1} \right) p_{K+1}^{\omega_K - 1} \times \cdots \\ &(1 - p_1)^{\omega_1 - (v_2 + \omega_2)} \times \cdots \\ &\times (1 - p_{K-1})^{\omega_{K-1} - 1 - (v_{K-1} + \omega_{K-1})} \end{aligned} \tag{199}$$

with each element p_i of mean and variance:

$$\begin{aligned} \mathbb{E}[p_i] &= \frac{v_{i'} \prod_{\ell=1}^{i'-1} \omega_\ell}{\prod_{\ell=1}^{i'} (v_\ell + \omega_\ell)} \\ \mathbb{V}[p_i] &= \frac{v_{i'}(v_{i'}+1) \prod_{\ell=1}^{i'-1} \omega_\ell(\omega_\ell+1)}{\prod_{\ell=1}^{i'} (v_\ell+\omega_\ell)(v_\ell+\omega_\ell+1)} \end{aligned} \tag{200}$$

We note here that the GD is a special case of the Dirichlet distribution if $\omega_i = \sum_{i'=i+1}^{K} v_{i'}$ for $i < K$ with $\omega_K = \omega_K$.

7.2.2 Infinite formulation of the hidden Markov model

Each row in the infinite state transition matrix is then modeled with a stick breaking prior and the state dependent parameters are drawn independently and with an identical distribution (iid) from G_0:

$$G_i = \textstyle\sum_{i'=1}^{\infty} a_{ii'} \delta_{\theta_{i'}}, \quad \boldsymbol{b}_i \overset{\text{iid}}{\sim} iHMM(\boldsymbol{v}, \boldsymbol{\omega}), \quad \theta_{i'} \overset{\text{iid}}{\sim} G_0 \tag{201}$$

Note that the state transitions are no longer required across the levels as in the DP and consequently required HDP. That is $V_{ii'} \sim \text{Beta}(v_{ii'}, \omega_{ii'})$ corresponds to the portion broken from the remainder of the unit length stick belonging to state i (defines the transition probability from state i to state i').

This is because each state-dependent parameters θ_i are drawn separately detaching the construction of the emission distributions from the construction of B. The construction of the initial states probabilities π is also performed similarly. The generative process below simplifies the required infinite parameterization [183]:

$$\begin{gathered} G_i = \sum_{i'=1}^{\infty} a_{ii'} \delta_{\theta_{i'}}, \quad a_{ii'} = V_{ii'} \prod_{f=1}^{i'-1} (1 - V_{if}) \\ V_{ii'} \sim \text{Beta}\left(1, \epsilon_{ii'}\right), \theta_{i'} \overset{iid}{\sim} G_0, \epsilon_{ii'} \overset{iid}{\sim} \text{Gamma}(\text{cc}, dd) \end{gathered} \tag{202}$$

where we have fixed $v_i = 1 \forall i$ and $\omega = \epsilon$ to highlight the similarity between the variable in this capacity and the Dirichlet distribution. This allows us to exploit the resultant conjugacy with the Gamma distribution for higher flexibility where hyperparameter setting plays a significant role in the model. For instance, the posterior of an $\epsilon_{ii'}$ is:

$$p\left(\epsilon_{ii'} | V_{ii'}, cc, dd\right) = \text{Gamma}\left(cc + 1, dd - \ln\left(1 - V_{ii'}\right)\right) \tag{203}$$

7.2.3 Variational inference learning

The exponential growth of the number of possible sequences to be summed as the length of the time series increases renders Eq. (182) computationally intractable [93]. However, an introduction of the approximate distribution $q(B, C, \pi, \Lambda, S, L)$ of the true posterior $p(B, C, \pi, \Lambda, S, L|X)$ enables us to derive a lower bound. When q is equal the true posterior, the inequality is tight. Hence,

$$\ln(p(X)) \quad = \quad \mathcal{L}(q) \quad - \quad KL(q(B, C, \pi, \Lambda, S, L)||p(B, C, \pi, \Lambda, S, L|X)) \tag{204}$$

where $\mathcal{L}(q)$ is the lower bound and KL is the Kullback-Leibler distance between the true posterior and the approximate distribution [92, 94] where q may be factorized; i.e., $q(B, C, \pi, \Lambda, S, L) = q(B)q(C)q(\pi)q(\Lambda)q(S, L)$ where $q(\Lambda) = q(\vec{\varepsilon})$ for the Dirichlet and $q(\Lambda) = q(\vec{\kappa})q(\alpha)q(\beta)$ for the BL distribution. This variational approximation is performed iteratively with expectation (E-step) and maximization (M-step). Let $\langle x_i \rangle$ be the expected number of observations from a component in

an iteration with a K-dimensional truncation, the variational equations can then be expressed as:

$$\begin{aligned}
\langle \ln V_i \rangle &= \Psi\left(1 + \langle x_i \rangle\right) - \Psi\left(1 + \epsilon_i + \sum_{i'=i}^{K} \langle x_{i'} \rangle\right) \\
\langle \ln\left(1 - V_i\right) \rangle &= \Psi\left(\epsilon_i + \sum_{i'=i+1}^{K} \langle x_{i'} \rangle\right) \\
&\quad - \Psi\left(1 + \epsilon_i + \sum_{i'=i}^{K} \langle x_{i'} \rangle\right) \\
\langle \ln p_1 \rangle &= \langle \ln V_i \rangle \\
\langle \ln p_k \rangle &= \langle \ln V_k \rangle + \sum_{i'=1}^{k-1} \langle \ln\left(1 - V_{i'}\right) \rangle \quad 2 \leq k < K \\
\langle \ln p_K \rangle &= \sum_{i'=1}^{K-1} \langle \ln\left(1 - V_{i'}\right) \rangle
\end{aligned} \tag{205}$$

We also note that $\langle \epsilon_{ii'} \rangle = cc_{ii'}/dd_{ii'}$ via the posterior parameters $cc_{ii'}$ and $dd_{ii'}$. We breakdown the distribution $BL(\vec{x}|\vec{\kappa}, \alpha, \beta)$ corresponding to the prior factorization assumption made to $q(\Lambda)$. Note that a similar procedure may be derived for the Dirichlet distribution with the exclusion of the α and β approximations. This yields the following evaluation:

$$\begin{aligned}
&\ln(p^*(X_t|\vec{\kappa}_{s_t,m_t}, \alpha_{s_t,m_t}, \beta_{s_t,m_t})) = \gamma_{ijt}^{C} \int q(\vec{\kappa}) q(\alpha, \beta) \ln(\nu(X_t|\vec{\kappa}_{s_t,m_t}) \\
&\quad \times \eta(X_t|\alpha_{s_t,m_t}, \beta_{s_t,m_t})) d\vec{\kappa} d\alpha d\beta = \gamma_{ijt}^{C} \left(\langle \ln(\nu(X_t|\vec{\kappa})) \rangle_{q(\vec{\kappa})} + \langle \ln(\eta(X_t|\alpha, \beta)) \rangle_{q(\alpha,\beta)} \right)
\end{aligned} \tag{206}$$

where $\gamma_{ijt}^{C} \triangleq q(s_t = i, m_t = j)$, $*$ superscript denotes an optimized parameter,

$$\begin{aligned}
\langle \ln(\nu(X_t|\vec{\kappa})) \rangle_{q(\vec{\kappa})} &= \left\langle \ln\left(\frac{\Gamma(\sum_{d=1}^{D} \alpha_{ijd})}{\prod_{d=1}^{D} \Gamma(\alpha_{ijd})} \right) \right\rangle_{q(\vec{\kappa})} \\
&\quad + \sum_{d=1}^{D} \ln(X_{td}) \langle \alpha_{ijd} - 1 \rangle_{q(\vec{\kappa})} - \ln\left(\sum_{d=1}^{D} X_{td} \right) \sum_{d=1}^{D} \langle \alpha_{ijd} \rangle_{q(\vec{\kappa})} \\
&= J(\alpha_{ijl}) + \sum_{d=1}^{D} \ln(X_{td}) \left(\frac{u_{ijd}}{v_{ijd}} - 1 \right) - \ln\left(\sum_{d=1}^{D} X_{td} \right) \sum_{d=1}^{D} \left(\frac{u_{ijd}}{v_{ijd}} \right)
\end{aligned} \tag{207}$$

and

$$\begin{aligned}\langle \ln(\eta(X_t|\alpha_{ij},\beta_{ij}))\rangle_{q(\alpha_{ij},\beta_{ij})} &= \left\langle \ln\left(\frac{\Gamma(\alpha_{ij}+\beta_{ij})}{\Gamma(\alpha_{ij})\Gamma(\beta_{ij})}\right)\right\rangle_{q(\alpha,\beta)} \\ &+ \ln\left(\sum_{d=1}^{D} X_{td}\right)\langle\alpha_{ij}\rangle_{q(\alpha,\beta)} + \ln\left(1-\sum_{d=1}^{D} X_{td}\right)\langle\beta_{ij}-1\rangle_{q(\alpha,\beta)} \\ &= J(\alpha_{ij},\beta_{ij}) + \ln\left(\sum_{d=1}^{D} X_{td}\right)\left(\frac{g_{ij}}{h_{ij}}\right) + \ln\left(1-\sum_{d=1}^{D} X_{td}\right)\left(\frac{e_{ij}}{r_{ij}}-1\right) \end{aligned} \quad (208)$$

$J(\alpha_{ijl})$ and $J(\alpha_{ij},\beta_{ij})$ are analytically intractable. Consequently, they are approximated by their lower bounds as derived in [46].

We then compute the sufficient statistics for determination of the posterior in the M-step:

$$\begin{aligned} q(B) &= \prod_{i=1}^{K} GD\left(v_i',\omega_i'\right) \\ q(\vec{\kappa}) &= \prod_{d=1}^{D}\prod_{i=1}^{K}\prod_{j=1}^{M} Gamma(\alpha_{ijd}|u_{ijd}^*,v_{ijd}^*) \\ q(\alpha) &= \prod_{i=1}^{K}\prod_{j=1}^{M} Gamma(\alpha_{ij}|g_{ij}^*,h_{ij}^*) \\ q(\beta) &= \prod_{i=1}^{K}\prod_{j=1}^{M} Gamma(\beta_{ij}|e_{ij}^*,r_{ij}^*) \\ q(\pi) &= \prod_{i=1}^{K} \mathrm{DR}\left(v_\pi',\omega_\pi'\right) \\ q(\boldsymbol{\epsilon}) &= \prod_{i=1}^{K}\prod_{i'=1}^{K-1} Gamma\left(c+1, d-\langle\ln\left(1-V_{ii'}\right)\rangle\right) \\ q\left(\epsilon_\pi\right) &= \prod_{i=1}^{K-1} Gamma\left(\tau_{\pi 1}+1, \tau_{\pi 2}-\langle\ln\left(1-V_{\pi i}\right)\rangle\right) \end{aligned} \quad (209)$$

where

$$u^*_{ijl} = u_{ijl} + \sum_{p=1}^{P} \langle Z_{pij} \rangle \bar{\alpha}_{ijl} \left[\Psi \left(\sum_{d=1}^{D} \bar{\alpha}_{ijd} \right) - \Psi(\bar{\alpha}_{ijl}) \right.$$
$$\left. + \sum_{d=1, d \neq l}^{D} \Psi' \left(\sum_{d=1}^{D} \bar{\alpha}_{ijd} \right) \bar{\alpha}_{ijd} (\langle \ln(\alpha_{ijd}) \rangle - \ln(\bar{\alpha}_{ijd})) \right] \tag{210}$$

$$v^*_{ijl} = v_{ijl} - \sum_{p=1}^{P} \langle Z_{pij} \rangle \left[\ln(X_{pl}) - \ln \left(\sum_{d=1}^{D} X_{pd} \right) \right] \tag{211}$$

$$g^*_{ij} = g_{ij} + \sum_{p=1}^{P} \langle Z_{pij} \rangle [\Psi(\bar{\alpha}_{ij} + \bar{\beta}_{ij}) - \Psi(\bar{\alpha}_{ij}) \tag{212}$$
$$+ \bar{\beta}_{ij} \Psi'(\bar{\alpha}_{ij} + \bar{\beta}_{ij}) (\langle \ln(\beta_{ij}) \rangle - \ln(\bar{\beta}_{ij}))] \bar{\alpha}_{ij}$$

$$h^*_{ij} = h_{ij} - \sum_{p=1}^{P} \langle Z_{pij} \rangle \ln \left(\sum_{d=1}^{D} X_{pd} \right) \tag{213}$$

$$e^*_{ij} = e_{ij} + \sum_{p=1}^{P} \langle Z_{pij} \rangle [\Psi(\bar{\alpha}_{ij} + \bar{\beta}_{ij}) - \Psi(\bar{\beta}_{ij}) \tag{214}$$
$$+ \bar{\alpha}_{ij} \Psi'(\bar{\alpha}_{ij} + \bar{\beta}_{ij}) (\langle \ln(\alpha_{ij}) \rangle - \ln(\bar{\alpha}_{ij}))] \bar{\beta}_{ij}$$

$$r^*_{ij} = r_{ij} - \sum_{p=1}^{P} \langle Z_{pij} \rangle \ln \left(1 - \sum_{d=1}^{D} X_{pd} \right) \tag{215}$$

where $\Psi'(.)$ is the trigamma function and Z_{pij} is an indicator function for X_{pt} belonging to state i and mixture component j. Hence, $\langle Z_{pij} \rangle = \sum_{t=1}^{T} \gamma^C_{pijt} = p(s = i, m = j|X)$ and the responsibilities are computed using the forward-backward algorithm [10]. The entire procedure repeats until convergence is reached.

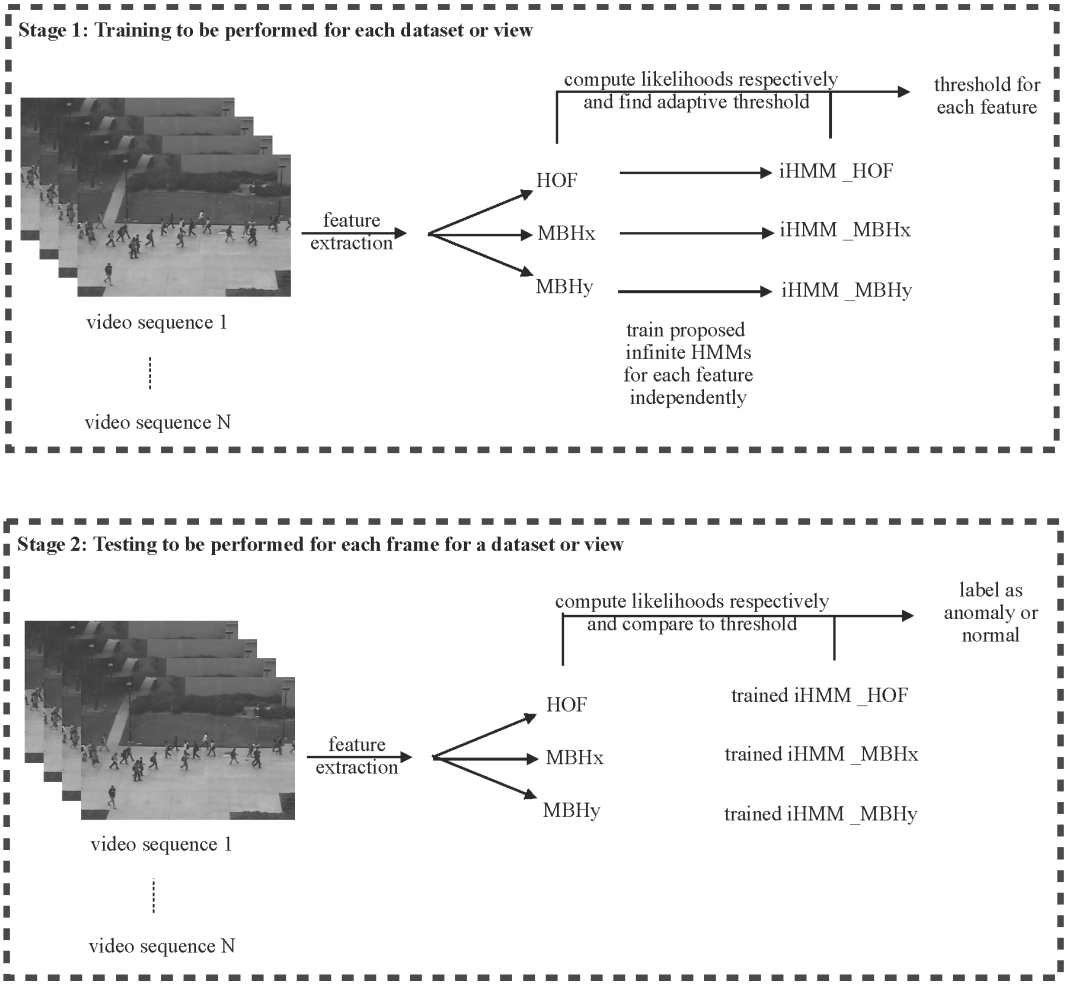


Figure 7.2: An overview of the proposed realtime robust anomaly detection framework. This applies to both the infinite Dirichlet and BL based HMMs.

7.3 Proposed Anomaly Detection Framework

We represent each of the video sequences with a series of extracted histogram of optical flow (HOF) and motion boundary histogram (MBH) descriptors [101]. For the HOF, the orientations are quantized into 9 bins and normalized with the L_2 norm. Derivatives of the optical flow are evaluated separately along the horizontal (MBHx) and vertical (MBHy) components to compute the MBH. The latter effectively captures relative motion between pixels and suppresses constant motion information to mute noise from background motion. The HOF and MBH descriptors may be extricated using any interest point detector [100]. In this paper, we extract the points along the

motion trajectory for both the training and the testing video sequences [101].

In order to use HMMs for anomaly detection, the probability of a sequence given a model λ is computed with the forward algorithm and then compared to the predetermined threshold. Each set of feature histograms extracted for each of the datasets has a model λ whose parameters must be estimated. This training procedure is performed by maximizing the probability of a given set of training non-anomalous observations using the proposed variational inference learning approach for infinite proportional HMMs.

Once the likelihoods of the testing video sequences are computed with the corresponding trained HMMs, they must be compared to a threshold to determine the presence of an anomaly in a frame. We statistically choose such a threshold in order to enable our scheme to be adaptive to any features extracted and from any video data; i.e., the same framework can be directly applied for a different feature set as well as datasets via the proposed threshold setting process. In this work, we apply the Chebyshev's theorem that dictates that at least $1 - (1/\chi^2)$ of the data must lie within $\langle X \rangle \pm \chi$std where $\langle X \rangle$ represents the mean of the data and std its standard deviation. In our framework, we choose $\chi = 125$ in order to reduce the false alarm rate that many anomaly detection systems suffer from. Hence, this addresses the robustness requirements of our proposed framework. This yields in the detection of anomalies that are not within 99.9936% of the data distribution.

Although we have predetermined the value of χ, it is a variable that may be adjusted according to the system requirements for a higher level of anomaly detection as per the application of the framework. That is if applied in security video surveillance systems, for instance, the authorities concerned may choose to enforce a tighter threshold as required. Moreover, this setup also allows the threshold to adapt to variability in the perspective distortion as well as other intricacies according to the nature of the features extracted.

We also investigate a fusion scheme of the three final predictions made for each of the video frames by each of the infinite HMMs. The final anomaly detection decision in this case is made through the highest number of votes. This is carried out for the two different proposed models: the Dirichlet and BL-based infinite HMMs. The proposed anomaly detection framework can be observed in Fig. 7.2.

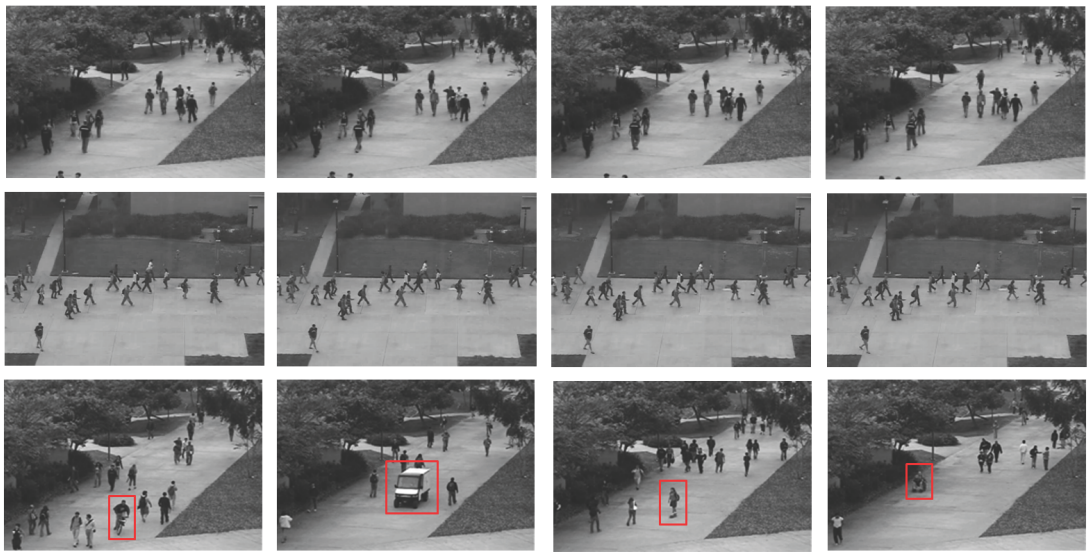


Figure 7.3: Samples of the UCSD ped1 normal sequences (top row), ped2 normal sequences (second row), and anomalous sequences from ped1 (third row - left to right - biker, cart, skater, and wheelchair).

7.4 Experimental Setup and Results

7.4.1 Datasets

The proposed framework is tested on the public real-world UCSD ped1 and ped2 datasets with different people densities and some extent of perspective distortion [184]. Each of the datasets is made up of a training video set (normal sequences with no anomalies) and a testing video set (normal and anomalous sequences) and represent different scenes. Normal sequences have only pedestrians, while abnormal sequences may contain people walking across a walkway, skaters, bikers, and small carts among others. Samples of the datasets are shown in Fig. 7.3.

Abnormalities were not staged and hence are naturally occurring. This allows us to test the proposed framework on real world data. The data also includes ground-truth of the anomalies. Nonetheless, we exclude ped1 training sequences 2, 23, and 25 where unexpected anomalies have been located in them [94].

7.4.2 Quantitative evaluation criteria

We compute the equal error rate (EER) for quantitative evaluation of our proposed model and comparison with various state-of-the-art methodologies on the UCSD datasets. The smaller the EER, the better the performance of the system. EER represents a compromise between the true positive rate (TPR) and false positive rate (FPR). TPR represents the rate of correctly detected frames to all abnormal frames in ground truth. This is mathematically denoted by $TPR = TP/(TP+FN)$ where TP is the number of true positive frames, and FN is the number of false negative frames. On the other hand, the rate of incorrectly detected frames to all normal frames in ground truth is the FPR. That is $FPR = FP/(FP + TN)$ where FP is the number of false positive frames, and TN is the number of true negative frames. We also measure the computational time required for testing sequences using the proposed framework. This evaluates the realtime capabilities of the system.

7.4.3 Results and comparison with state-of-the-art

We experimentally set the truncation level for both the infinite Dirichlet and BL HMMs at $K = 10$ with $v = 10e - 6$ and $\omega = 0.1$. In Table 7.1, we compare quantitatively the proposed method and its computational time with various relevant state-of-the-art anomaly detection methodologies. We report the EER, the system configuration, the frame processing time, and the implementation language used. Our proposed framework performs competitively with near real-time processing. Note that the processing times of the proposed method are dependent on the programming methods employed such as the use of parallel computing and optimization techniques at large, and hence may be further improved for production.

A simple classifier is built based on the distance of the nearest neighbor of the query feature to the features extracted in the training set and then compared to a threshold in [172]. This is an approach that does not require training and hence is non-parametric. This relates it to the non-parametric formulation of the proposed HMMs to extend to infinity. A Gaussian-based HMM approach is taken in [186] along with texture map and 3-D Harris features. This is related to our proposed model as they both utilize HMMs. However, our proposed HMMs are infinite and based

Table 7.1: Comparison of the proposed framework with state-of-the-art methods for anomaly detection.

Method	EER-ped1	EER-ped2	Processing time (sec/frame)	Configuration and language
[185]	31.0%	30.0%	0.1	CPU: 2.6GHz, RAM: 3GB
[186]	32.4%	28.5%	5.1	CPU: 2GHz (dual core), RAM: 4GB, MATLAB
[187]	19.9%	N/A	1.3	CPU: 3.4GHz, RAM: 4GB, MATLAB
[188]	2.9%	9.9%	N/A	N/A
[189]	27.0%	26.9%	1.2	CPU: 3.5GHz, RAM: 16GB, C++
[190]	17.8%	18.5%	1.2	CPU: 3.5GHz, RAM: 16GB, C++
[191]	24.0%	24.4%	0.4	CPU: 2.8GHz, RAM: 128GB
[192]	N/A	19.0%	0.04	CPU: 3.5GHz, RAM: 8GB, MATLAB
HMMD [29]	28.9%	18.5%	0.2	CPU: 3.4GHz, RAM: 5GB, MATLAB
HMMGD [29]	29.0%	22.0%	0.2	CPU: 3.4GHz, RAM: 5GB, MATLAB
HMMBL [29]	29.0%	16.6%	0.2	CPU: 3.4GHz, RAM: 5GB, MATLAB
VBHMMD [94]	31.4%	12.5%	0.2	CPU: 3.4GHz, RAM: 5GB, MATLAB
VBHMMGD [94]	29.0%	13.8%	0.2	CPU: 3.4GHz, RAM: 5GB, MATLAB
iHMMDR - HOF (proposed)	17.1%	50.9%	0.004	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB
iHMMDR - MBHx (proposed)	17.1%	79.4%	0.004	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB
iHMMDR - MBHy (proposed)	17.1%	79.4%	0.004	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB
iHMMDR - Fused (proposed)	18.0%	28.9%	0.01	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB
iHMMBL - HOF (proposed)	7.3%	22.3%	0.004	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB
iHMMBL - MBHx (proposed)	7.2%	22.3%	0.004	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB
iHMMBL - MBHy (proposed)	7.2%	22.2%	0.004	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB
iHMMBL - Fused (proposed)	7.8%	9.5%	0.01	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB

on the Dirichlet and BL distributions to better model proportional sequential data. Nonetheless, it serves to depict the influence of the choice of emission probability on global performance.

[187] presents the Gaussian process regression for the modeling of frequent geometric patterns between Spatial-Temporal Interest Points (STIP) and via 3-D-scale-invariant feature transforms. [188] is closely related whereby histograms of gradients and optical flow are computed for appearance and motion modeling via points of interest detected using 3-D Harris corner functions with a support vector machine (SVM) for the classification. Histograms of oriented swarms for dynamics modeling with HOG for appearance modeling are combined in [189] along with an SVM. A hierarchical approach via mixtures of dynamic textures and several spatial scales to build a normalcy model is proposed in [190]. Spatio-temporal convolutional neural networks are fed with raw data of small spatiotemporal video volumes selected using optical flow in [191] to capture appearance and motion information for anomaly detection.

On the other hand, a combination of two local, spatial and temporal, self-similarity descriptors with a global descriptor learned using autoencoders is utilized in [192]. A typical Baum-Welch algorithm trained HMM approach is proposed in [29]. However, the HMMs are proportional in nature, based on the Dirichlet (HMMD), GD (HMMGD), and Beta-Liouville (HMMBL) distributions, and build upon the features proposed in [185]. It is then intriguing to observe that the use of HMMs can radically improve the results as shown.

Finally, [94] presents an extension to [29] through the application of variational learning for the proportional Dirichlet and Generalized Dirichlet HMMs denoted by VBHMMD and VBHMMGD, respectively, in Table 7.1. The latter HMM methods are particularly relevant due to the use of proportional HMMs, especially with variational learning. It is then interesting to observe the improvement in time and EER by extending the model to infinity as well as the use of a different set of features.

Overall, it can be clearly observed that the proposed framework is efficient, robust, and realtime. While the proposed fusion is simple, it still significantly improves the results. This is especially apparent for the infinite Dirichlet and BL HMM models of the UCSD ped2 dataset. This depicts the complementary nature of the features chosen and reinforces the unity of the proposed framework. The results also clearly illustrate how the use of the BL distribution can drastically enhance the

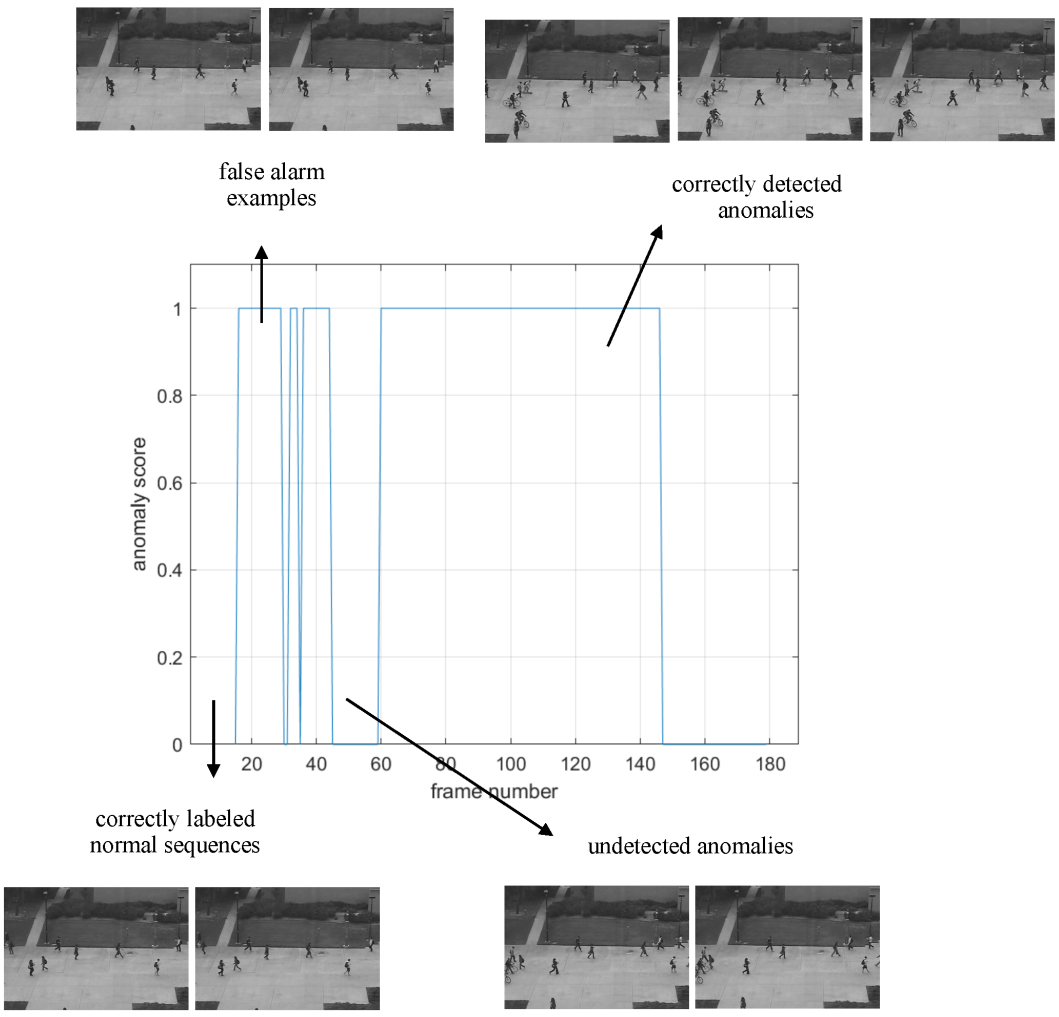


Figure 7.4: Qualitative results of our proposed realtime robust anomaly detection framework. Samples are shown from test sequence 7 from UCSD ped2 dataset modeled by the proposed infinite BL HMM trained on the extracted HOF features.

performance of the variational inference based infinite proportional HMMs. This is due to the more flexible covariance structure of the BL distribution in comparison to the enforced negative covariance in the Dirichlet.

We also observe that the EER is lower for the ped1 dataset for our proposed framework due to the higher number of training sequences available. This enables the framework to better capture the variability in normal events and hence reduces the false alarm rate. We also report the states which have been effectively removed in the proposed infinite HMMs. That is the optimum number of states have been determined automatically which addresses an area of active research in HMMs. For the HOF infinite HMMs, states 1, 5, 6, and 10 are inactive for ped1 and states 5, 9, and 10 are inactive for ped2. Only state 10 is inactive for the MBHx and MBHy. Note that this flexibility in the infinite HMM setup allows for seamless optimum model construction. We also depict a sample of our qualitative results as shown in Fig. 7.4.

Chapter 8

Infinite Generalized Dirichlet Hidden Markov Models with Simultaneous Feature Selection

> The ones who are crazy enough to think that they can change the world are the ones who do.
>
> Steve Jobs

A closely related model is the infinite extension of the Generalized Dirichlet hidden Markov models. An interesting characteristic of this distribution that has set it apart is its transformation to Beta distributions; hence, enabling the incorporation of a simultaneous feature selection paradigm. This is the topic of this chapter with a validation performed in frame-based video anomaly detection.

8.1 Introduction

Nowadays, the overwhelming need for data modeling is at an all time high. This is mainly driven by the vast amount of data, around 2.5 quintilian bytes, that is generated daily [167]. Modeling such data has several advantages. These include the potential removal of noise from the original signals, studying the data source or the generating process without the need of its explicit availability, and

efficiently building powerful recognition and prediction systems [3]. Moreover, a consequent implication of successfully formulating a model for a data source is the ability to simulate as much data as needed to further study such a phenomenon without the need to carry out costly experiments.

Meanwhile, the prevalence of cameras and video surveillance has led to a significant increase in research interest in the field of automatic real time monitoring systems [146, 172, 147, 169]. Such systems aim to assist human operators in the detection of potential threats; especially due to the need to oversee multiple feeds simultaneously for long periods of time. Hence, efficient automatic video monitoring plays an important role in the detection of malicious threats and the avoidance of consequent incidents. While the testing and development of these systems have been facilitated by the recent release of real-world data sets [170, 171], an anomaly; i.e., a threat, by its nature is a rare occurrence that is therefore seldom recorded. Hence, sufficient data for automatic detection training is unavailable [193].

A typical methodology to overcome this hindrance in the development of video anomaly detection systems is training on video sequences; i.e., data that is considered normal. This enables the system to report anomalies as outliers [146, 172, 147]. This is especially important for the design and training of appropriate machine learning techniques. This forms the basis of multiple approaches that are paving the way for new advancements in anomaly video detection [173].

In this paper, we propose variational learning of infinite Hidden Markov Models (HMM) for the modeling of anomalies. A HMM is a probabilistic model that is appropriate for dynamic data [174, 194, 175]. It is usually trained with the Baum-Welch method; a variation of the Expectation Maximization (EM) algorithm specially designed for HMMs. However, employing a variational inference technique is advantageous as it alleviates the various drawbacks of the Baum Welch method [176, 177]. The latter include over-fitting or sub-optimal generalization performance [46].

We also propose the incorporation of a simultaneous feature selection paradigm [151, 152]. Intuitively, the higher the number of features used to represent a given dataset, the higher the efficiency of the model is expected. However, some features can be noisy, redundant, or uninformative in practice [158]. Hence, these can hinder the modeling performance. In particular, the presence of many irrelevant features introduces a bias resulting in unreliable homogeneity measures. Feature selection is the process of reducing the number of collected features to a subset of relevant features.

In addition to improving the performance of the models, it also aids in ameliorating model interpretation and decreasing the risk of overfitting [154]. It is noteworthy to mention that this study is novel in its treatment of the feature selection paradigm in terms of the mathematical derivations required for the deployment of non-Gaussian HMMs in contrast to the traditional Gaussian based [155, 195, 156, 157].

Furthermore, we also investigate another primary area of HMM research that lies in modeling emission probabilities of proportional data. They naturally result from numerous preprocessing procedures; such as the commonly used histograms. Recent studies have proven successful utilizing distributions that correspond to proportional data such as the Dirichlet [94, 48] and the Beta-Liouville (BL) [46] distributions.

Also, choosing the correct number of states to model the data with HMMs is another area that warrants further investigations. HMMs can be finite or infinite [178]. In finite HMMs, the number of states is usually determined as a result of an exhaustive search for the appropriate count. This can be achieved through the implementation of infinite HMM models through the means of non-parametric Bayesian methods [179].

Hence, the novel contributions of this work can be summarized as:

- We propose a mathematical formulation for construction of infinite HMMs for continuous proportional data. We particularly focus on the use of Generalized Dirichlet (GD) distribution given its proven effectiveness in modeling proportional sequential data. This distribution is closely related to the Dirichlet, but it relaxes the constraint of negative covariance at the cost of additional parameters. In comparison to deep learning techniques, this is an explainable approach that requires less training data and is less computationally expensive.

- We present a variational inference learning approach for the proposed infinite HMMs. The technique guarantees convergence within a reasonable time frame and provides an accurate approximation of the posterior.

- We incorporate a simultaneous feature selection paradigm into the proposed infinite HMMs for improved performance and efficient deployment.

- We propose an end-to-end framework for realtime robust anomaly detection in videos. The

framework performs competitively with state-of-the-art relevant methods and may be applied to any video data.

8.2 Infinite Hidden Markov Models for Proportional Data

In this section, we discuss the main components of the adaptive algorithm for infinite GD-based HMMs with simultaneous feature selection. Section 8.2.1 examines the Dirichlet and stick-breaking processes, Section 8.2.2 discusses the infinite formulation of the HMM, Section 8.2.3 details the variational inference learning, and Section 8.2.4 presents the feature saliency model for the incorporation of the simultaneous feature selection approach. Moreover, a summary of the symbols utilized in this manuscript with their corresponding definitions can be observed in Table 8.1 for ease of mathematical reference.

A HMM is characterized by an underlying stochastic process with K hidden states whereby the transition between the states $B = \{b_{ii'} = P(s_t = i'|s_{t-1} = i)\}$ at time t. An initial probability π governs each state s_t, in which an observation is emitted corresponding to its respective parameters of a probability distribution $\varkappa$ with a mixing matrix $C = \{c_{ij} = P(m_t = j|s_t = i)\}$ where $j \in [1, M]$ such that M is the number of mixture components in set $L = \{m_1, ..., m_M\}$. Hence, a continuous HMM may be defined as $\lambda = \{B, C, \varkappa, \pi\}$. A graphical model of the latter HMM is depicted in Fig. 8.1.

The likelihood of a sequence with HMMs may be denoted by Eq. (64) where $\varkappa_{ij} = (\varkappa_{1ij}, ..., \varkappa_{Dij})$ with $\varkappa$ defined according to the GD distribution for proportional data in this work.

Formally, a D-dimensional GD distribution is denoted by:

$$GD(X|\vec{\iota}, \vec{\vartheta}) = \prod_{d=1}^{D} \frac{\Gamma(\iota_d + \vartheta_d)}{\Gamma(\iota_d)\Gamma(\vartheta_d)} X_d^{\iota_d - 1} \left(1 - \sum_{r=1}^{d} X_r\right)^{\zeta_d} \tag{216}$$

where $\vec{\iota} = (\iota_1, ..., \iota_D)$ and $\vec{\vartheta} = (\vartheta_1, ..., \vartheta_D)$ are the real and strictly positive parameters of the distribution and $X \in \mathbb{R}_+^D$ with $\sum_{d=1}^{D} X_d < 1$ corresponding to the $(D+1)$-dimensional proportional vector that adds up to one. Finally, ζ_d is computed using the parameters of the distribution as $\vartheta_d - \iota_{d+1} - \vartheta_{d+1}$, when $d \neq D$. Otherwise, $\zeta_d = \vartheta_D - 1$.

Table 8.1: Definitions of symbols utilized in the manuscript.

Symbol	Definition
t	time index
K	number of states
D	number of feature dimensions
B	transition matrix
i, i'	state index
d	dimension index
$b_{ii'}$	transition matrix index
s_t	state at time t
$\varkappa$	probability distribution for HMM emission
π	initial probability
C	mixing matrix
M	number of mixture components
j	mixture component index
L	set of mixtures
S	set of states
m_M	component of a mixture in a state
c_{ij}	index of the mixing weight of the mixture in a state
$p(X\|B, C, \varkappa, \pi)$	likelihood of a sequence
$\Lambda_{ij} = (\Lambda_{1ij}, ..., \Lambda_{Dij})$	parameters of the GD
c_{s_t,m_t}	HMM mixing weight of a mixture component in a state
b_{s_{t-1},s_t}	a transition weight between states of a HMM
X	time series data
$\rho_t(i)$	forward variable
$\theta_t(i)$	backward variable
$\gamma^t_{s_t,m_t}, \eta$	forward-backward algorithm resultant variables
Z	hidden variables
$\iota, \vartheta, \zeta_d$	parameters of GD distribution
ϱ, ζ, u, and v	hyperparameters of the GD distribution
λ	A HMM model
ϵ	positive scalar precision of Dirichlet process
G_0	base distribution of Dirichlet process
$G \sim DP(\epsilon G_0)$	a draw from Dirichlet process
θ_i	location drawn from G_0 associated with measure p_i
p_i and V_i	parameters for the stick breaking process
v and ω	parameters of V_i that are drawn from a GD
$V_{ii'} \sim \text{Beta}\,(v_{ii'}, \omega_{ii'})$	transition probability from state i to state i'
$\text{Beta}\,(X_{dt}\|\epsilon_d, \tau_d)$	the distribution of irrelevant feature(s)
φ	feature saliency
z_d	feature assignment
$p(B, C, \pi, \Lambda, S, L\|X)$	true posterior
$q(B, C, \pi, \Lambda, S, L)$	approximate distribution of the true posterior
$\mathcal{L}(q)$	the approximate lower bound of the posterior
KL	the Kullback-Leibler distance between the true posterior and the approximate distribution
$\langle x_i \rangle$	the expected number of observations from a component in an iteration
Y	projection of X into a transformed space
$\vec{\alpha}$	parameters of the DR distribution
$\mathbb{E}$	expectation of a variable
TP	true positives
TN	true negatives

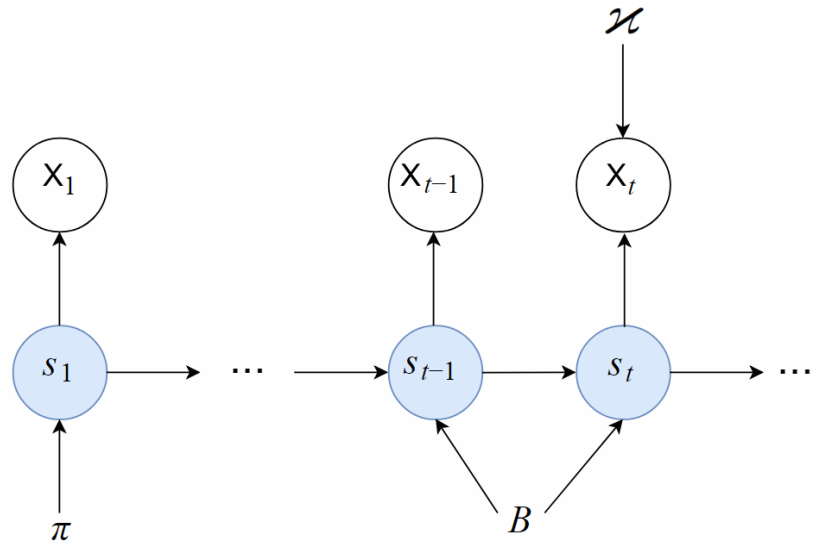


Figure 8.1: Graphical model representation of a continuous hidden Markov model. Symbols in unshaded circles denote the observed variables, symbols in shaded circles indicate the hidden states, and edges represent conditional dependencies between the states or the variables.

8.2.1 The Dirichlet and stick-breaking processes

The Dirichlet process (DP) is a parameterized stochastic process with a positive scalar precision ϵ and base distribution G_0 [180]. It forms a distribution over discrete distributions that place its mass on a countably infinite collection of atoms. The base distribution places location of atoms and the concentration variable controls the range of the mass spreading around atoms. This may be expressed for disjoint sets of $A = \{A_1, ..., A_D\}$ in measurable space Θ, where $\cup_i A_i = \Theta$, as:

$$(G(A_1), \ldots, G(A_D)) \sim \mathrm{DR}(\epsilon G_0(A_1), \ldots, \epsilon G_0(A_D)) \tag{217}$$

The DP has infinite dimensions, $D \to \infty$, as G_0 is a continuous distribution. G represents a draw from the DP that is denoted by $G \sim DP(\epsilon G_0)$ and may be written as:

$$G = \sum_{i=1}^{\infty} p_i \delta_{\theta_i} \tag{218}$$

where θ_i is a location drawn from G_0 and is associated with measure p_i. θ_i may then be interpreted as the emission distribution of an HMM at state i. A draw from the DP is then defined with a

stick-breaking process as [181]:

$$\begin{aligned} &G = \textstyle\sum_{i=1}^{\infty} p_i \delta_{\theta_i}, \quad p_i = V_i \prod_{i'=1}^{i-1} (1 - V_{i'}) \\ &V_i \sim \text{Beta}(1, \epsilon), \quad \theta_i \sim G_0 \end{aligned} \tag{219}$$

where the influence of ϵ on a draw from DP is clearly impertinent. As $\epsilon \to 0$, a random component with location drawn from G_0 depicts a degenerate measure due to the breaking of the entire stick and consequent allocation to a single component. On the other hand, infinitely small breaks and convergence of G to the distribution of the draws occur when $\epsilon \to \infty$ so that G_0 itself is reproduced.

This may be conversely expressed as a DP mixture model with the following generative process when the interest is shifted to the parameters of a distribution rather than the data itself:

$$x_i \left| \theta_i \sim p\left(x | \theta_i\right), \quad \theta_i \right| G \sim G, \quad G | \epsilon G_0 \sim DP\left(\epsilon G_0\right) \tag{220}$$

HMMs are a special case of the mixture models which are state-dependent. That is each mixture has different weights but the same support. Consequently, we can define $\theta_i \equiv (\Lambda_{ij}, ..., \Lambda_{iM})$ and express the state-dependent mixture model of a continuous proportional HMM as:

$$x_t \left| \theta_{s_t} \sim Dist\left(\theta_{s_t}\right), \theta_{s_t} \right| s_{t-1} \sim G_{s_{t-1}}, G_i = \sum_{i'=1}^{D} b_{ii'} \delta_{\theta_{i'}} \tag{221}$$

where $Dist$ is defined according to the appropriate distribution to be applied. In this paper, this is the GD distribution and the initial state has been assumed to be chosen from π. For an infinite HMM, it may then be inferred that each transition should be modeled as a DP. Nonetheless, an issue arises with this approach. Particularly, assume that each row i is drawn for the infinite state transition matrix with:

$$\begin{aligned} &G_i = \textstyle\sum_{i'=1}^{\infty} b_{ii'} \delta_{\theta_{ii'}}, \quad b_{ii'} = V_{ii'} \prod_{f=1}^{i'-1} (1 - V_{if}) \\ &V_{ii'} \sim \text{Beta}(1, \epsilon), \quad \theta_{ii'} \sim G_0 \end{aligned} \tag{222}$$

where $b_{ii'}$ is the i'th component of the infinite vector b_i. Note then that for each state indexing $\theta_{ii'}$, the probability of a transition to a previously visited state is zero when G_0 is continuous. This

is because $p(\theta_\varsigma = \theta_\varrho) = 0$ for $\varsigma \neq \varrho$. This indicates the impracticality of such an approach for formulating an infinite HMM.

An extension of the DP is the hierarchical DP (HDP) which has been proposed to resolve such a problem. Indeed, the HDP is a two level approach whereby the base distribution is itself drawn from a DP resulting in an almost decidedly discrete G_0 [182]:

$$G_\varpi \sim DP\left(\Upsilon G_0\right), \quad G_0 \sim DP(\epsilon H) \tag{223}$$

Hence, substantial weight on the same set of states is impacted by multiple draws. If we truncate at K and write the top level DP in a stick-breaking form, we may then explicitly denote the second level DP as:

$$G_0 = \sum_{i=1}^{K} p_i \delta_{\boldsymbol{\theta}_i}, \quad p_i = V_i \prod_{i'=1}^{i-1} \left(1 - V_{i'}\right) \tag{224}$$

$$V_i \sim \text{Beta}(1, \epsilon), \quad \theta_i \sim H \tag{225}$$

$$\begin{gathered} \left(G_\varpi\left(\theta_1\right), G_\varpi\left(\theta_2\right), \ldots, G_\varpi\left(\theta_K\right)\right) \sim \\ \text{DR}\left(\Upsilon p_1, \Upsilon p_2, \ldots, \Upsilon p_K\right) \end{gathered} \tag{226}$$

where $G(\theta_i)$ is a probability at location θ_i. However, the lack of conjugacy between the two levels (number of states and their emission parameters at the top level and the mixing weights as priors to draw the transition probabilities at the second level) means that a true variational solution does not exist [179]. Consequently, we utilize the priors of the form:

$$p_i = V_i \textstyle\prod_{i'=1}^{i-1} \left(1 - V_{i'}\right), \quad V_i \sim \text{Beta}\left(\upsilon_i, \omega_i\right) \tag{227}$$

This formulation has a more flexible parametrization where the weights and locations are effectively detached. Though this stick-breaking process is infinite, υ and ω terminate at a finite number K with $p_{K+1} \equiv 1 - \sum_{i=1}^{K} p_i$, so that the result is a draw from the GD distribution. This is necessary for the variational learning approach as discussed earlier. Hence, we must utilize a GD prior for the state transitions. Formally, for $\boldsymbol{V} = (V_1, ..., V_K)$:

$$F(\boldsymbol{V}) = \prod_{i=1}^{K} F(V_i) = \prod_{i=1}^{K} \frac{\Gamma(v_i + \omega_i)}{\Gamma(v_i)\Gamma(\omega_i)} V_i^{v_i - 1} (1 - V_i)^{\omega_i - 1} \tag{228}$$

The density of p may then be derived with a change of variables from $\boldsymbol{V}$:

$$\begin{aligned} F(p) &= \prod_{i=1}^{K} \left(\frac{\Gamma(v_i+\omega_i)}{\Gamma(v_i)\Gamma(\omega_i)} p_i^{v_i - 1} \right) p_{K+1}^{\omega_K - 1} \times \cdots \\ &(1 - p_1)^{\omega_1 - (v_2 + \omega_2)} \times \cdots \\ &\times (1 - p_{K-1})^{\omega_{K-1} - 1 - (v_{K-1} + \omega_{K-1})} \end{aligned} \tag{229}$$

with each element p_i of mean and variance:

$$\begin{aligned} \mathbb{E}[p_i] &= \frac{v_{i'} \prod_{\ell=1}^{i'-1} \omega_\ell}{\prod_{\ell=1}^{i'} (v_\ell + \omega_\ell)} \\ \mathbb{V}[p_i] &= \frac{v_{i'}(v_{i'}+1) \prod_{\ell=1}^{i'-1} \omega_\ell(\omega_\ell+1)}{\prod_{\ell=1}^{i'} (v_\ell+\omega_\ell)(v_\ell+\omega_\ell+1)} \end{aligned} \tag{230}$$

8.2.2 Infinite formulation of the hidden Markov model

Each row in the infinite state transition matrix is then modeled with a stick breaking prior and the state dependent parameters are drawn independently and with an identical distribution (iid) from G_0:

$$G_i = \textstyle\sum_{i'=1}^{\infty} a_{ii'} \delta_{\theta_{i'}}, \quad \boldsymbol{b}_i \overset{\text{iid}}{\sim} iHMM(\boldsymbol{v}, \boldsymbol{\omega}), \quad \theta_{i'} \overset{\text{iid}}{\sim} G_0 \tag{231}$$

Note that the state transitions are no longer required across the levels as in the DP and consequently required HDP. That is $V_{ii'} \sim \text{Beta}(v_{ii'}, \omega_{ii'})$ corresponds to the portion broken from the remainder of the unit length stick belonging to state i (defines the transition probability from state i to state i'). This is because each state-dependent parameters θ_i are drawn separately detaching the construction of the emission distributions from the construction of B. The construction of the initial states probabilities π is also performed similarly. The generative process below simplifies the required infinite parameterization [183]:

$$\begin{aligned} G_i &= \sum_{i'=1}^{\infty} a_{ii'} \delta_{\theta_{i'}}, \quad a_{ii'} = V_{ii'} \prod_{f=1}^{i'-1} (1 - V_{if}) \\ V_{ii'} &\sim \text{Beta}(1, \epsilon_{ii'}), \theta_{i'} \overset{iid}{\sim} G_0, \epsilon_{ii'} \overset{iid}{\sim} \text{Gamma}(\text{cc}, dd) \end{aligned} \tag{232}$$

where we have fixed $v_i = 1 \; \forall i$ and $\omega = \epsilon$ to highlight the similarity between the variable in this capacity and the Dirichlet distribution. This allows us to exploit the resultant conjugacy with the Gamma distribution for higher flexibility where hyperparameter setting plays a significant role in the model. For instance, the posterior of an $\epsilon_{ii'}$ is:

$$p\left(\epsilon_{ii'}|V_{ii'}, cc, dd\right) = \text{Gamma}\left(cc + 1, dd - \ln\left(1 - V_{ii'}\right)\right) \tag{233}$$

8.2.3 Variational inference learning

We derive a variational inference approach in order to find the parameters of the proposed iHMM. The exponential growth of the number of possible sequences to be summed as the length of the time series increases renders Eq. (64) computationally intractable [93]. However, an introduction of the approximate distribution $q(B, C, \pi, \Lambda, S, L)$ of the true posterior $p(B, C, \pi, \Lambda, S, L|X)$ enables us to derive a lower bound. When q is equal the true posterior, the inequality is tight. Hence,

$$\ln(p(X)) \quad = \quad \mathcal{L}(q) \quad - \quad KL(q(B, C, \pi, \Lambda, S, L)||p(B, C, \pi, \Lambda, S, L|X)) \tag{234}$$

where $\mathcal{L}(q)$ is the lower bound and KL is the Kullback-Leibler distance between the true posterior and the approximate distribution [92, 94] where q may be factorized, i.e. $q(B, C, \pi, \Lambda, S, L) = q(B)q(C)q(\pi)q(\Lambda)q(S, L)$. This variational approximation is performed iteratively with expectation (E-step) and maximization (M-step). Let $\langle x_i \rangle$ be the expected number of observations from a component in an iteration with a K-dimensional truncation, the variational equations can then be

expressed as:

$$\begin{aligned}
\langle \ln V_i \rangle &= \Psi\left(1 + \langle x_i \rangle\right) - \Psi\left(1 + \epsilon_i + \sum_{i'=i}^{K} \langle x_{i'} \rangle\right) \\
\langle \ln\left(1 - V_i\right) \rangle &= \Psi\left(\epsilon_i + \sum_{i'=i+1}^{K} \langle x_{i'} \rangle\right) \\
&\quad - \Psi\left(1 + \epsilon_i + \sum_{i'=i}^{K} \langle x_{i'} \rangle\right) \\
\langle \ln p_1 \rangle &= \langle \ln V_i \rangle \\
\langle \ln p_k \rangle &= \langle \ln V_k \rangle + \sum_{i'=1}^{k-1} \langle \ln\left(1 - V_{i'}\right) \rangle \quad 2 \leq k < K \\
\langle \ln p_K \rangle &= \sum_{i'=1}^{K-1} \langle \ln\left(1 - V_{i'}\right) \rangle
\end{aligned} \tag{235}$$

We also note that $\langle \epsilon_{ii'} \rangle = cc_{ii'}/dd_{ii'}$ via the posterior parameters $cc_{ii'}$ and $dd_{ii'}$.

The posterior probability or the responsibilities for the GD distribution may be expressed as mixture of Beta distributions through first projecting the data into a transformed space [193]. This allows for better precision since an error in one of the estimations does not propagate to the other parameters. Moreover, with the reduced dimensionality, precision of the solution is naturally improved. In particular, independence between the features is now no longer merely an assumption but a fact. This more efficient methodology is defined as:

$$GD(X|\vec{\iota}, \vec{\vartheta}) \propto \prod_{d=1}^{D} \text{Beta}(Y_d|\iota_d, \vartheta_d) \tag{236}$$

where

$$Y_d = \begin{cases} X_d, & \text{for } d = 1 \\ X_d / \left(1 - \sum_{i=1}^{d-1} X_i\right), & \text{for } d \in [2, D]. \end{cases} \tag{237}$$

and $\text{Beta}(Y_d|\iota_d, \vartheta_d)$ is the Beta distribution that is denoted by:

$$\text{Beta}(Y_d|\iota_d, \vartheta_d) = \frac{\Gamma(\iota_d + \vartheta_d)}{\Gamma(\iota_d)\Gamma(\vartheta_d)} Y_d^{\iota_d - 1}(1 - Y_d)^{\vartheta_d - 1} \tag{238}$$

One can mathematically show that $\forall d, Y_d \sim \text{Beta}(\iota_d, \vartheta_d) = \text{DR}(\alpha_1 = \iota_d, \alpha_2 = \vartheta_d)$. Hence, the

estimation problem of $q(\Lambda)$ is abridged as $D-1$ smallest estimation problems of unidimensional Dirichlet distributions defined by:

$$DR(X|\vec{\alpha}) = \frac{\Gamma(\sum_{d=1}^{D}\alpha_d)}{\prod_{d=1}^{D}\Gamma(\alpha_d)}\prod_{d=1}^{D}X_d^{\alpha_d-1} \tag{239}$$

where $\vec{\alpha} = (\alpha_1, ..., \alpha_D)$ is the real and strictly positive parameter of the distribution.

This yields the following evaluation:

$$\ln(p^*(X_t|\vec{\alpha}_{s_t,m_t})) = \gamma_{ijt}^C \int q(\vec{\alpha})\ln(\nu(X_t|\vec{\alpha}_{s_t,m_t}))d\vec{\alpha} \tag{240}$$

$$= \gamma_{ijt}^C \left(\langle \ln(\nu(X_t|\vec{\alpha}))\rangle_{q(\vec{\alpha})}\right) \tag{241}$$

where $\gamma_{ijt}^C \triangleq q(s_t = i, m_t = j)$, $*$ superscript denotes an optimized parameter and

$$\begin{aligned}\langle \ln(\nu(X_t|\vec{\alpha}))\rangle_{q(\vec{\alpha})} &= \left\langle \ln\left(\frac{\Gamma(\sum_{d=1}^{D}\alpha_{ijd})}{\prod_{d=1}^{D}\Gamma(\alpha_{ijd})}\right)\right\rangle_{q(\vec{\alpha})} \\ &+ \sum_{d=1}^{D}\ln(X_{td})\langle\alpha_{ijd}-1\rangle_{q(\vec{\alpha})} - \ln\left(\sum_{d=1}^{D}X_{td}\right)\sum_{d=1}^{D}\langle\alpha_{ijd}\rangle_{q(\vec{\alpha})} \\ &= J(\alpha_{ijl}) + \sum_{d=1}^{D}\ln(X_{td})\left(\frac{u_{ijd}}{v_{ijd}}-1\right) - \ln\left(\sum_{d=1}^{D}X_{td}\right)\times \\ &\qquad \sum_{d=1}^{D}\left(\frac{u_{ijd}}{v_{ijd}}\right)\end{aligned} \tag{242}$$

$J(\alpha_{ijl})$ is analytically intractable. Consequently, it is approximated by its lower bound as derived

in [46]. We then compute the sufficient statistics for determination of the posterior in the M-step:

$$
\begin{aligned}
q(B) &= \prod_{i=1}^{K} \text{GD}\left(v_i', \omega_i'\right) \\
q(\vec{\alpha}) &= \prod_{d=1}^{D} \prod_{i=1}^{K} \prod_{j=1}^{M} \text{Gamma}(\alpha_{ijd} | u_{ijd}^*, v_{ijd}^*) \\
q(\pi) &= \prod_{i=1}^{K} \text{DR}\left(v_\pi', \omega_\pi'\right) \\
q(\boldsymbol{\epsilon}) &= \prod_{i=1}^{K} \prod_{i'=1}^{K-1} \text{Gamma}\left(c+1, d - \langle \ln\left(1 - V_{ii'}\right)\rangle\right) \\
q\left(\epsilon_\pi\right) &= \prod_{i=1}^{K-1} \text{Gamma}\left(\tau_{\pi 1}+1, \tau_{\pi 2} - \langle \ln\left(1 - V_{\pi i}\right)\rangle\right)
\end{aligned}
\tag{243}
$$

where

$$
\begin{aligned}
u_{ijl}^* = u_{ijl} + \sum_{p=1}^{P} \langle Z_{pij}\rangle \bar{\alpha}_{ijl} \Bigg[\Psi\left(\sum_{d=1}^{D} \bar{\alpha}_{ijd}\right) - \Psi(\bar{\alpha}_{ijl}) \\
+ \sum_{d=1, d\neq l}^{D} \Psi'\left(\sum_{d=1}^{D} \bar{\alpha}_{ijd}\right) \bar{\alpha}_{ijd}(\langle \ln(\alpha_{ijd})\rangle - \ln(\bar{\alpha}_{ijd})) \Bigg]
\end{aligned}
\tag{244}
$$

$$
v_{ijl}^* = v_{ijl} - \sum_{p=1}^{P} \langle Z_{pij}\rangle \left[\ln(X_{pl}) - \ln\left(\sum_{d=1}^{D} X_{pd}\right) \right] \tag{245}
$$

and $\Psi'(.)$ is the trigamma function and Z_{pij} is an indicator function for X_{pt} belonging to state i and mixture component j. Hence, $\langle Z_{pij}\rangle = \sum_{t=1}^{T} \gamma_{pijt}^{C} = p(s = i, m = j|X)$ and the responsibilities are computed using the forward-backward algorithm [10]. The entire procedure repeats until convergence is reached. An important aspect when applying variational inference is the convergence assessment. We trace the convergence systematically by monitoring the update difference in the estimated parameters of λ. This is set with an adaptive threshold which we have set at 10^{-3} between the iterations or reaching a maximum number of iterations set at 300.

8.2.4 Feature selection

We define whether a feature is relevant or not using a feature saliency technique. Feature saliency recasts feature selection as a parameter estimation problem [155]. New parameters, known as feature saliencies, are added to the latent variable model and used to find clusters embedded in the feature subspace. Mathematically, given a certain state, assume that each of the dimensions of the features is independent with latent indicator variable z_d, $z = (z_1, \ldots, z_D)$, indicates which component the dth observation belongs to, $z_d = (z_{d1}, \ldots, z_{dM})$ and each element z_{dj} is assigned value 1 when the observation X_i is associated with component j; otherwise, it is 0. Then:

$$\begin{aligned} &p\left(X_t|z, s_t = i, \lambda\right) \\ &\qquad = \prod_{d=1}^{D} p\left(X_{dt}|\Lambda_{id}\right)^{z_d} \text{Beta}\left(X_{dt}|\epsilon_d, \tau_d\right)^{1-z_d} \end{aligned} \tag{246}$$

where Beta is the conditional Beta distribution that is used to model irrelevant features and defined as:

$$\text{Beta}(X_{dt}|\epsilon_d, \tau_d) = \prod_{d=1}^{D} \frac{\Gamma(\epsilon_d + \tau_d)}{\Gamma(\epsilon)\Gamma(\tau_d)} X_d^{\epsilon_d - 1} \left(1 - X_d\right)^{\tau_d - 1} \tag{247}$$

The joint distribution of X_t and z given s is:

$$\begin{aligned} &p\left(X_t, z|s_t = i, \lambda\right) \\ &= \textstyle\prod_{d=1}^{D} \left[\varphi_d p\left(X_{dt}|\Lambda_{id}\right)\right]^{z_d} \left[\left(1 - \varphi_d\right) \text{Beta}(X_{dt}|\epsilon_d, \tau_d)\right]^{1-z_d} \end{aligned} \tag{248}$$

where the marginal probability of z and X_t given s are given by:

$$P(z|\lambda) = \prod_{d=1}^{D} \varphi_d^{z_d} \left(1 - \varphi_d\right)^{1-z_d} \tag{249}$$

$$\begin{aligned} p(X_t|\Lambda_{s_t,m_t}) = \prod_{d=1}^{D} &\left[\varphi_d p\left(X_{dt}|\Lambda_{id}\right)\right] \left[\left(1 - \varphi_d\right)\right. \\ &\left. \times \text{Beta}(X_{dt}|\epsilon_d, \tau_d)\right] \end{aligned} \tag{250}$$

respectively. This may then be used for the computation of the complete data likelihood in Eq. (64) accordingly. A graphical representation of the proposed iHMM can be observed in Fig. 8.2.

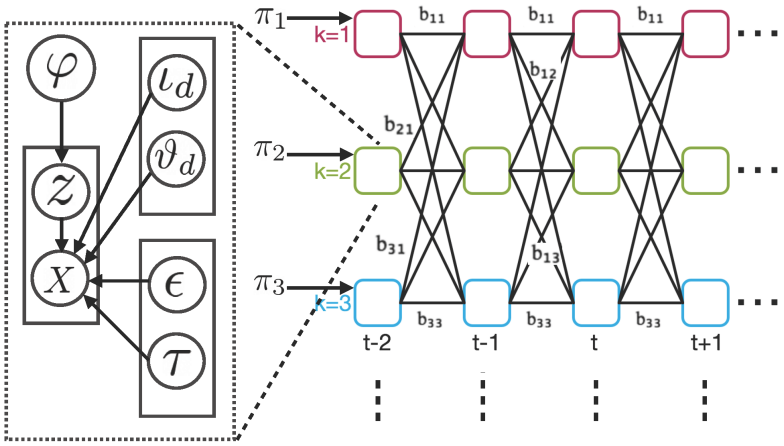


Figure 8.2: Graphical model representation of the proposed infinite GD-based hidden Markov model with simultaneous feature selection.

8.3 Proposed Anomaly Detection Framework

We represent each of the video sequences with a series of extracted histogram of optical flow (HOF) and motion boundary histogram (MBH) descriptors [101]. For the HOF, the orientations are quantized into 9 bins and normalized with the L_2 norm. Derivatives of the optical flow are evaluated separately along the horizontal (MBHx) and vertical (MBHy) components to compute the MBH. The latter effectively captures relative motion between pixels and suppresses constant motion information to mute noise from background motion. The HOF and MBH descriptors may be extracted using any interest point detector [100]. In this paper, we extract the points along the motion trajectory for both the training and the testing video sequences [101].

In order to use HMMs for anomaly detection, the probability of a sequence given a model λ is computed with the forward algorithm and then compared to the predetermined threshold. Each set of feature histograms extracted for each of the datasets has a model λ whose parameters must be estimated. This training procedure is performed by maximizing the probability of a given set of training non-anomalous observations using the proposed variational inference learning approach for infinite proportional HMMs.

Once the likelihoods of the testing video sequences are computed with the corresponding trained HMMs, they must be compared to a threshold to determine the presence of an anomaly in a frame. We statistically choose such a threshold in order to enable our scheme to be adaptive to any features

extracted and from any video data, i.e. the same framework can be directly applied for a different feature set as well as datasets via the proposed threshold setting process. In this work, we apply the Chebyshev's theorem that dictates that at least $1 - (1/\chi^2)$ of the data must lie within $\langle X \rangle \pm \chi$std where $\langle X \rangle$ represents the mean of the data and std its standard deviation. In our framework, we choose $\chi = 125$ in order to reduce the false alarm rate that many anomaly detection systems suffer from. Hence, this addresses the robustness requirements of our proposed framework. This yields in the detection of anomalies that are not within 99.9936% of the data distribution.

Although we have predetermined the value of χ, it is a variable that may be adjusted according to the system requirements for a higher level of anomaly detection as per the application of the framework. That is if applied in security video surveillance systems, for instance, the authorities concerned may choose to enforce a tighter threshold as required. Moreover, this setup also allows the threshold to adapt to variability in the perspective distortion as well as other intricacies according to the nature of the features extracted.

We also investigate a fusion scheme of the three final predictions made for each of the video frames by each of the infinite HMMs. The final anomaly detection decision in this case is made through the highest number of votes. The proposed anomaly detection framework[1] can be observed in Fig. 7.2.

8.4 Experimental Setup and Results

8.4.1 Datasets

The proposed framework is tested on the public real-world UCSD ped1 and ped2 datasets with different people densities and some extent of perspective distortion [184]. Each of the datasets is made up of a training video set (normal sequences with no anomalies) and a testing video set (normal and anomalous sequences) and represent different scenes. Normal sequences have only pedestrians, while abnormal sequences may contain people walking across a walkway, skaters, bikers, and small carts among others. Samples of the datasets are shown in Fig. 7.3. These training video sequences are 34 and 16 videos for each of the UCSD ped1 and ped2 datasets. Nonetheless, we exclude ped1

[1]The complete source code of this paper is available upon request.

training sequences 2, 23, and 25 where unexpected anomalies have been located in them [94].

On the other hand, each of the UCSD ped1 and ped2 datasets also contain testing datasets which are made up of 36 and 12 testing video samples respectively. Abnormalities were not staged and hence are naturally occurring. This allows us to test the proposed framework on real world data. The data also includes ground-truth of the anomalies.

8.4.2 Quantitative evaluation criteria

We compute the equal error rate (EER) on the frame-level for quantitative evaluation of our proposed model and comparison with various state-of-the-art methodologies on the UCSD datasets. The smaller the EER, the better the performance of the system. EER represents a compromise between the true positive rate (TPR) and false positive rate (FPR). TPR represents the rate of correctly detected frames to all abnormal frames in ground truth. This is mathematically denoted by $TPR = TP/(TP + FN)$ where TP is the number of true positive frames, and FN is the number of false negative frames. On the other hand, the rate of incorrectly detected frames to all normal frames in ground truth is the FPR. That is $FPR = FP/(FP + TN)$ where FP is the number of false positive frames, and TN is the number of true negative frames. We also measure the computational time required for testing sequences using the proposed framework. This evaluates the realtime capabilities of the system.

8.4.3 Results and comparison with state-of-the-art

We experimentally set the truncation level for both the infinite GD HMMs and the infinite GD HMMs with simultaneous feature selection at $K = 100$ with $v = 10e-6$ and $\omega = 0.1$. In Table 3.1, we compare quantitatively the proposed method and its computational time with various relevant state-of-the-art anomaly detection methodologies. We report the EER, the system configuration, the frame processing time, and the implementation language used. Our proposed framework performs competitively with near real-time processing. Note that the processing times of the proposed method are dependent on the programming methods employed such as the use of parallel computing and optimization techniques at large, and hence may be further improved for production.

A simple classifier is built based on the distance of the nearest neighbor of the query feature

Table 8.2: Comparison of the proposed framework with state-of-the-art methods for anomaly detection.

Method	EER-ped1	EER-ped2	Processing time (sec/frame)	Configuration and language
[185]	31.0%	30.0%	0.1	CPU: 2.6GHz, RAM: 3GB
[186]	32.4%	28.5%	5.1	CPU: 2GHz (dual core), RAM: 4GB, MATLAB
[187]	19.9%	N/A	1.3	CPU: 3.4GHz, RAM: 4GB, MATLAB
[188]	2.9%	9.9%	N/A	N/A
[189]	27.0%	26.9%	1.2	CPU: 3.5GHz, RAM: 16GB, C++
[190]	17.8%	18.5%	1.2	CPU: 3.5GHz, RAM: 16GB, C++
[191]	24.0%	24.4%	0.4	CPU: 2.8GHz, RAM: 128GB
[192]	N/A	19.0%	0.04	CPU: 3.5GHz, RAM: 8GB, MATLAB
HMMD [29]	28.9%	18.5%	0.2	CPU: 3.4GHz, RAM: 5GB, MATLAB
HMMGD [29]	29.0%	22.0%	0.2	CPU: 3.4GHz, RAM: 5GB, MATLAB
HMMBL [29]	29.0%	16.6%	0.2	CPU: 3.4GHz, RAM: 5GB, MATLAB
VBHMMD [94]	31.4%	12.5%	0.2	CPU: 3.4GHz, RAM: 5GB, MATLAB
VBHMMGD [94]	29.0%	13.8%	0.2	CPU: 3.4GHz, RAM: 5GB, MATLAB
iHMMGD - HOF (proposed)	17.1%	79.4%	0.004	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB
iHMMGD - MBHx (proposed)	17.1%	79.4%	0.005	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB
iHMMGD - MBHy (proposed)	17.1%	79.4%	0.005	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB
iHMMGD - Fused (proposed)	18.0%	35.1%	0.006	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB
iHMMGD (feature selection) - HOF (proposed)	17.7%	52.6%	0.004	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB
iHMMGD (feature selection) - MBHx (proposed)	17.1%	72.2%	0.005	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB
iHMMGD (feature selection) - MBHy (proposed)	17.1%	76.0%	0.005	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB
iHMMGD (feature selection) - Fused (proposed)	28.4%	24.8%	0.006	CPU: 3.6GHz, RAM: 32GB, C++/MATLAB

to the features extracted in the training set and then compared to a threshold in [185]. This is an approach that does not require training and hence is non-parametric. This relates it to the non-parametric formulation of the proposed HMMs to extend to infinity. A Gaussian-based HMM approach is taken in [186] along with texture map and 3-D Harris features. This is related to our proposed model as they both utilize HMMs. However, our proposed HMMs are infinite and based on the GD distribution to better model proportional sequential data. Nonetheless, it serves to depict the influence of the choice of emission probability on global performance.

[187] presents the Gaussian process regression for the modeling of frequent geometric patterns between Spatial-Temporal Interest Points (STIP) and via 3-D-scale-invariant feature transforms. [188] is closely related whereby graph features are computed for appearance and motion modeling via points of interest detected using 3-D Harris corner functions with a support vector machine (SVM) for the classification. Nonetheless, the features are not exactly the same but this represents an opportunity for potential future work whereby extracting graph features can be investigated for better representation of the input video data and may hence improve the proposed model performance.

Furthermore, the classification is performed with a discriminative model; i.e., support vector machines (SVM). This explains its low EER. On the other hand, no mention of the time taken of the experiments was recorded. Histograms of oriented swarms for dynamic modeling with HOG for appearance modeling are combined in [189] along with a SVM. A hierarchical approach via mixtures of dynamic textures and several spatial scales to build a model for normal event is proposed in [190]. Spatio-temporal convolutional neural networks are fed with raw data of small spatiotemporal video volumes selected using optical flow in [191] to capture appearance and motion information for anomaly detection.

On the other hand, a combination of two local, spatial and temporal, self-similarity descriptors with a global descriptor learned using autoencoders is utilized in [192]. A typical Baum-Welch algorithm trained HMM approach is proposed in [29]. However, the HMMs are proportional in nature, based on the Dirichlet (HMMD), GD (HMMGD), and Beta-Liouville (HMMBL) distributions, and build upon the features proposed in [185]. It is then intriguing to observe that the use of HMMs can radically improve the results as shown.

Finally, [94] presents an extension to [29] through the application of variational learning for the proportional Dirichlet and GD HMMs denoted by VBHMMD and VBHMMGD, respectively, in Table 8.2. The latter HMM methods are particularly relevant due to the use of proportional HMMs, especially with variational learning. It is then interesting to contemplate the improvement in time and EER by extending the model to infinity as well as the use of a different set of features.

Overall, it can be clearly observed that the proposed framework is efficient, robust, and nearly realtime. While the proposed fusion is simple, it still significantly improves the results. This is especially apparent for the infinite proposed GD HMM models of the UCSD ped2 dataset. This depicts the complementary nature of the features chosen and reinforces the unity of the proposed framework. The results also clearly illustrate how the use of the GD distribution can drastically enhance the performance of the variational inference based infinite proportional HMMs. This is due to the more flexible covariance structure of the GD distribution in comparison to the enforced negative covariance in the Dirichlet.

The influence of the fusion algorithm is particularly desired in degraded circumstances whereby the EER of each of the independent features is relatively high. However, when the results are

acceptable as in the case of the ped1, use of the fusion technique is not advised given its lower resultant EER. Nonetheless, the use of the infinite GD HMM approach improves the results.

We also observe that the EER is lower for the ped1 dataset for our proposed framework due to the longer time recorded for each of the video samples and the total available sequences. This enables the framework to better capture the variability in normal events and hence reduces the false alarm rate. On the other hand, incorporation of the simultaneous feature selection approach significantly ameliorates performance on the ped2 dataset. This is expected given the improvement consequences of removing noisy and reduntant features from the data. However, this is surprisingly not the case for the ped1 dataset. We find this might be due to the population sample of the features themselves. Indeed, some features may have proven to be significant in the training set whereas not so for the testing set.

We also report the states which have been effectively removed in the proposed infinite HMMs. That is the optimum number of states have been determined automatically which addresses an area of active research in HMMs. For the HOF infinite HMMs, five states (93, 94, 97, 98, and 100) were required for the modeling of the data for both datasets with the rest are inactive. Moreover, only two states (24 and 25) were needed for the MBHx and MBHy features. This is explained by the high modeling capabilities of the proposed iHMMs given the transformation of the GD into Beta distributions for actualization of feature independence which is merely an assumption for other distributions. Note that this flexibility in the iHMM setup allows for seamless optimum model construction.

Chapter 9

Online Learning for Dirichlet and Beta-Liouville Hidden Markov Models

> The two most powerful warriors are patience and time.
>
> Leo Tolstoy, *War and Peace*

In this chapter, we address the deployment of the proposed models. In particular, we investigate the problem of online learning and present it for the Dirichlet and the Beta-Liouville hidden Markov models. The validation is performed on the action recognition application.

9.1 Introduction

Many ubiquitous applications rely on automatic action recognition (AR). These include video surveillance [74], video retrieval [73], and video labeling [74]. Consequently, research attention in AR has increased in recent years. Typically, classification of a given video or image sequence or its assignment to a set of predefined classes is the objective of automatic AR [75]. The task is then based on lower level processing stages such as tracking and segmentation [76].

Different approaches for AR have been studied throughout the years with significant advances made in the past decades [77]. Most of the developed AR approaches are tested and implemented for the visible spectrum due to its popularity and availability [78]. Moreover, an abundant number of visible spectrum AR datasets is available such as UCF101 [80], KTH [81], and Weizmann [82].

Indeed, AR in general is fairly well-studied in the visible light spectrum with multiple successful applications [79]. Nonetheless, many challenges persist that limit its accuracy due to the need for the recognition of the exact action carried out. In surveillance systems particularly where both violent as well as non-violent actions should be taken into account.

A hidden Markov model (HMM) [3] is one of the machine learning approaches that may be used for AR. It is one of the most well-established mathematical formulations for time series modeling [3]. Its structure is formed primarily from a Markov chain of latent variables with each corresponding to the conditioned observation. A Markov chain is one of the least complicated ways to model sequential patterns in time series data. It allows us to maintain generality while relaxing the independent identically distributed assumption [12].

Early works mostly focused on the use of HMMs for discrete and Gaussian data [10]. A primary area of HMM research lies in modeling state emission probabilities of proportional data, i.e. strictly positive data that sum up to one. Multivariate proportional time series data naturally result from numerous preprocessing procedures, such as the commonly used histograms, and occur in various pattern recognition domains. Hence, in this paper, we utilize the Beta-Liouville (BL) distribution which has been proven to consistently outperform the Dirichlet distribution; i.e., the most commonly used distribution for the modelling of proportional data [48]. Furthermore, a HMM is usually trained with the Baum-Welch method; a variation of the Expectation Maximization algorithm. The proposed HMM is trained using a variational learning approach which incorporates prior knowledge into the training process [92]. Employing a variational Bayesian inference technique is advantageous as it overcomes the drawbacks of the Baum Welch algorithm. These include over-fitting or underfitting and sub-optimal generalization performance [46].

The main contributions of our work can be summarized as: (i) we propose the first variational learning based online HMM framework, to the best of our knowledge, for proportional data modeling of video data for continuous adaptation of the model to better fit the data and take into account all instances of a class; (ii) we implement the proposed online framework for the first time on video data, in particular on action recognition data for video surveillance; (iii) we compare the proposed method to the batch setup as well as the proportional data baseline; i.e., using the Dirichlet distribution with both the proposed online and the batch frameworks. This includes the first evaluation of

the online Dirichlet-based HMMs on the IOSB dataset.

9.2 Methods

9.2.1 Hidden Markov Models

A HMM is characterized by an underlying stochastic process with K hidden states that form a Markov chain. Each of the states is governed by an initial probability π, and the transition between the states at time t can be visualized with a transition matrix $B = \{b_{ii'} = P(s_t = i'|s_{t-1} = i)\}$. In each state s_t, an observation is emitted corresponding to its distribution which may be discrete or continuous. This is the observable stochastic process set.

The emission matrix of the discrete observations can be denoted by $\Xi = \{\Xi_i(m) = P(O_t = \xi_m|s_t = i)\}$ where $[m, t, i] \in [1, M]\times[1, T]\times[1, K]$, and the set of all possible discrete observations $\Xi = \{\xi_1, ..., \xi_m, ..., \xi_M\}$. On the other hand, the respective parameters of a probability distribution define the observation emission for a continuous observed symbol sequence. The Gaussian distribution is the most commonly used and is defined by its mean and covariance matrix $\varkappa = (\mu, \Sigma)$ [10, 14, 15]. Consequently, a mixing matrix must be defined $C = \{c_{ij} = P(m_t = j|s_t = i)\}$ in the case of continuous HMM emission probability distribution where $j \in [1, M]$ such that M is the number of mixture components in set $L = \{m_1, ..., m_M\}$. Hence, a discrete or continuous HMM may be defined with the following respective parameters $\lambda = \{B, \Xi, \pi\}$ or $\{B, C, \varkappa, \pi\}$. In this work, we consider the latter case which is defined as a proportional mixture model of BL distribution.

In D dimensions, a BL distribution is defined as Eq. (185) where $\vec{\alpha} = (\alpha_1, ..., \alpha_D)$, α, and β are the real and strictly positive parameters of the BL distribution, $\Gamma(t) = \int_0^\infty x^{t-1}e^{-x}dx$ is the Gamma function, and $\vec{x}$ is a D dimensional vector whereby $\vec{x} \in \mathbb{R}_+^D$ and $\sum_{d=1}^D x_d < 1$. For simplification, we also denote $\Lambda = [\vec{\alpha}, \alpha, \beta]$; the parameters of the BL distribution.

9.2.2 Online Setup for Variational Learning of Hidden Markov Models

In order to establish a fully adaptable surveillance system for maximum performance, an online framework has to be setup for HMM. This allows the system to retrain automatically as new data

becomes available; hence, taking into account all the various variations of the class in time as well as increasing the number of training instances for the HMM. This allows seamless employment of the proposed model for highly adaptable security and surveillance system.

The proposed online framework constitutes of two stages. The first stage is referred to as batch training whereby the HMM is trained with pre-existing training data. These parameters may be computed with the equations in an offline manner. The next stage then revolves around incremental training of the existing model to take into account new data that becomes available in time. In action recognition application for surveillance, this involves realtime videos that are recorded then fed into the system for classification of the action.

The online phase consists first of calculating the likelihood of a given length of incoming video (for instance from a surveillance camera) to classify the sequence. Once a class has been determined, the data is then used to train a separate BL HMM whose parameters are used to update the pretrained BL HMM of the corresponding class with a weighted average. The weight of the newly trained parameters is assigned according to the length, τ, of the newly available video sequence that is classified and is now incorporated into the training set after classification. On the other hand, the old parameters have a weight corresponding to the training data that has been used thus far. This formulation has the advantage of continuously reducing the weight of incoming data to the original training data. This effectively maintains the integrity of the model in case of a misclassified new entry or an anomaly data. Finally, the architecture of our proposed online HMM framework can be observed in Fig. 9.1.

The estimation of the HMM derived with the variational Bayesian approach uses the posterior probabilities through the assignment of parameter priors for integrating out the marginal likelihood of the data. This translates into regarding all the model parameters as random variables. The complete data likelihood of the HMM is then mathematically expressed as:

$$p(X) = \int d\pi dB dC d\varkappa \sum_{S,L} p(B, C, \pi, \varkappa) p(X, S, L | B, C, \pi, \varkappa) \tag{251}$$

However, this equation is intractable, so we introduce an approximate distribution $q(B, C, \pi, \varkappa, S, L)$

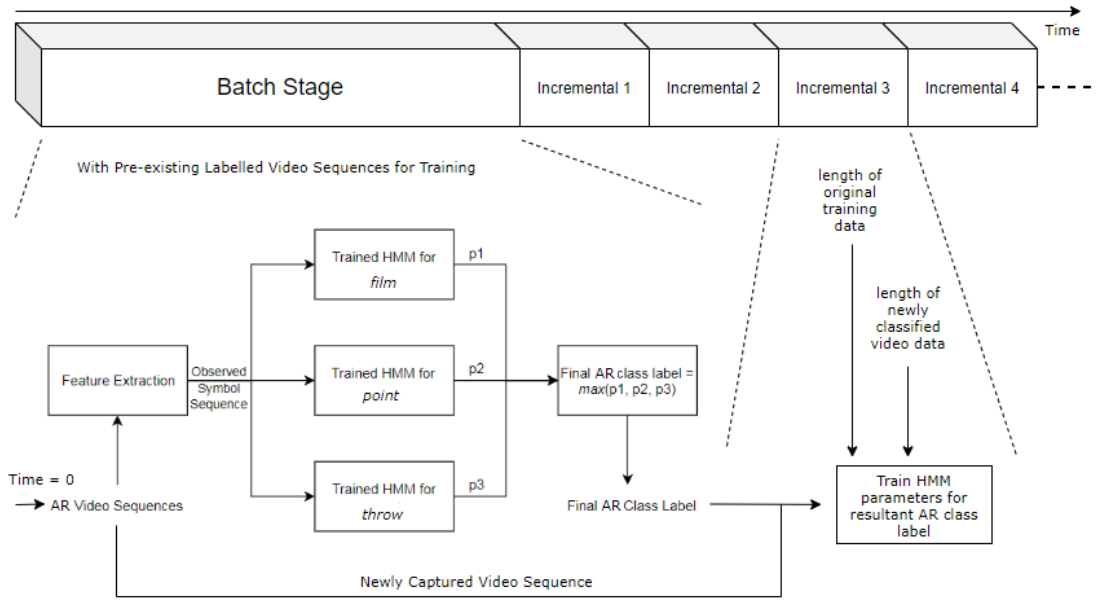


Figure 9.1: Proposed online framework for proportional hidden Markov modeling of action recognition videos for surveillance applications.

of the true posterior $p(B, C, \pi, \varkappa, S, L|X)$ for a lower bound, $\mathcal{L}(q)$, of KL Kullback-Leibler distance between the true posterior and the approximate distribution [92, 46].

The computation of the exact posterior distribution is intractable, so we only account for a certain family of distributions. As per the studied assumptions in [46, 95, 93, 96], q may be factorized; i.e., $q(B, C, \pi, \varkappa, S, L) = q(B)q(C)q(\pi)q(\varkappa)q(S, L)$ where $q(\varkappa) = q(\vec{\alpha})q(\alpha)q(\beta)$, with a similar factorization applying to p; i.e., the true distribution. Since the coefficients of the parameters π, B, and C are all less than one, strictly positive, and with a sum result equal to one for each row summation, their priors are chosen as Dirichlet distributions. For instance $p(\pi) = \mathcal{D}(\pi|\phi^{\pi}) = \mathcal{D}(\pi_1, ..., \pi_K|\phi_1^{\pi}, ..., \phi_K^{\pi})$. Similarly, a conjugate prior must also be defined over the emission distribution; the BL parameters $\vec{\alpha}$, α, and β. We adopt the Gamma distribution $\mathcal{G}(.)$ for positive conjugate prior approximations of the latter parameters, as we previously investigated by [46]. In the online approach, the corresponding pair hyperparameters u, g, h, e, r, and v

that are strictly positive are repeatedly evaluated in each incremental stage. These are denoted by:

$$u_{ijl}^{*}=\frac{\mathcal{L}}{\tau\theta\mathcal{L}}\left(u_{ijl}+\sum_{p=1}^{P}\langle Z_{pij}\rangle\bar{\alpha}_{ijl}(\Psi\left(\sum_{d=1}^{D}\bar{\alpha}_{ijd}\right)-\Psi(\bar{\alpha}_{ijl})+\right.$$
$$\left.\sum_{d=1,d\neq l}^{D}\Psi'\left(\sum_{d=1}^{D}\bar{\alpha}_{ijd}\right)\bar{\alpha}_{ijd}(\langle\ln(\alpha_{ijd})\rangle-\ln(\bar{\alpha}_{ijd})))\right)_{t} \tag{252}$$
$$+\frac{\tau\theta}{\tau\theta\mathcal{L}}\left(u_{ijl}^{*}\right)_{t+1}$$

$$v_{ijl}^{*}=\frac{\mathcal{L}}{\tau\theta\mathcal{L}}\left(v_{ijl}-\sum_{p=1}^{P}\langle Z_{pij}\rangle\left(\ln(X_{pl})-\ln\left(\sum_{d=1}^{D}X_{pd}\right)\right)\right)_{t} \tag{253}$$
$$+\frac{\tau\theta}{\tau\theta\mathcal{L}}\left(v_{ijl}^{*}\right)_{t+1}$$

$$g_{ij}^{*}=\frac{\mathcal{L}}{\tau\theta\mathcal{L}}\left(g_{ij}+\sum_{p=1}^{P}\langle Z_{pij}\rangle(\Psi(\bar{\alpha}_{ij}+\bar{\beta}_{ij})-\Psi(\bar{\alpha}_{ij})+\bar{\beta}_{ij}\right.$$
$$\left.\Psi'(\bar{\alpha}_{ij}+\bar{\beta}_{ij})(\langle\ln(\beta_{ij})\rangle-\ln(\bar{\beta}_{ij})))\bar{\alpha}_{ij}\right)_{t}+\frac{\tau\theta}{\tau\theta\mathcal{L}}\left(g_{ij}^{*}\right)_{t+1} \tag{254}$$

$$h_{ij}^{*}=\frac{\mathcal{L}}{\tau\theta\mathcal{L}}\left(h_{ij}-\sum_{p=1}^{P}\langle Z_{pij}\rangle\ln\left(\sum_{d=1}^{D}X_{pd}\right)\right)_{t}+\frac{\tau\theta}{\tau\theta\mathcal{L}}\left(h_{ij}^{*}\right)_{t+1} \tag{255}$$

$$e_{ij}^{*}=\frac{\mathcal{L}}{\tau\theta\mathcal{L}}\left(e_{ij}+\sum_{p=1}^{P}\langle Z_{pij}\rangle(\Psi(\bar{\alpha}_{ij}+\bar{\beta}_{ij})-\Psi(\bar{\beta}_{ij})+\bar{\alpha}_{ij}\right.$$
$$\left.\Psi'(\bar{\alpha}_{ij}+\bar{\beta}_{ij})(\langle\ln(\alpha_{ij})\rangle-\ln(\bar{\alpha}_{ij})))\bar{\beta}_{ij}\right)_{t}+\frac{\tau\theta}{\tau\theta\mathcal{L}}\left(e_{ij}^{*}\right)_{t+1} \tag{256}$$

$$r_{ij}^{*} = \frac{\mathcal{L}}{\tau\theta\mathcal{L}}\left(r_{ij} - \sum_{p=1}^{P}\langle Z_{pij}\rangle \ln\left(1 - \sum_{d=1}^{D} X_{pd}\right)\right)_{t} + \frac{\tau\theta}{\tau\theta\mathcal{L}}\left(r_{ij}^{*}\right)_{t+1} \tag{257}$$

where θ denotes the position of the current feed of data in reference to the start time of the realtime feed which we consider to be the median frame number, $Z_{pij} = 1$ if X_{pd} belongs to state i and mixture component j and $Z_{pij} = 0$ otherwise; i.e., it is an indicator function. Then, the weights of the data samples with respect to each mixture component are defined within the HMM framework. Consequently, $\langle Z_{pij}\rangle = \sum_{t=1}^{T}\gamma_{pijt}^{C} = p(s = i, m = j|X)$ and the responsibilities are computed via the forward-backward algorithm [10]. i and j are fixed for P observation vectors where $l \in [1, D]$, $i \in [1, K]$, and $j \in [1, M]$. $\Psi(.)$ is the digamma function, and $\Psi'(.)$ is the trigamma function; the logarithmic first and second derivatives of the Gamma function respectively. $\mathcal{L}$ is the number of pre-existing frames used for the batch training. The $*$ superscript implies the optimization of each of the corresponding parameters that the symbol is presented upon and $\langle.\rangle$ denotes the expectation with respect to the optimized parameter. It is also noteworthy to mention that the online update of the parameters expressed by the second additive partition in each of the equations is only executed for the update of the corresponding HMM of the selected label of the current feed of data. The definitions of the expected values of the aforementioned parameters are as follows:

$$\bar{\alpha}_{ijl} = \frac{u_{ijl}^{*}}{v_{ijl}^{*}}, \bar{\alpha}_{ij} = \frac{g_{ij}^{*}}{h_{ij}^{*}}, \bar{\beta}_{ij} = \frac{e_{ij}^{*}}{r_{ij}^{*}}, \langle \ln(\alpha_{ijl})\rangle = \Psi(u_{ijl}^{*}) - \ln(v_{ijl}^{*}) \tag{258}$$

$$\langle \ln(\alpha_{ij})\rangle = \Psi(g_{ij}^{*}) - \ln(h_{ij}^{*}), \langle \ln(\beta_{ij})\rangle = \Psi(e_{ij}^{*}) - \ln(r_{ij}^{*}) \tag{259}$$

The optimizations of $q(B)$, $q(C)$, and $q(\pi)$ are applicable to other continuous HMMs as they are independent of the emission distribution used. Therefore, these have already been studied in [46, 95, 90]. As such, the reader is referred to the aforementioned references for further details. $q(S, L)$ is then estimated in the E-step with the previously evaluated parameters now fixed and

utilizing the following definitions:

$$\pi_i^* \triangleq \exp\left[\langle \ln(\pi_i)\rangle_{q(\pi)}\right], \pi_i^* = \exp\left[\Psi(\omega_i^\pi) - \Psi(\sum_i \omega_i^\pi)\right],$$
$$b_{jj'}^* \triangleq \exp\left[\langle \ln(b_{jj'})\rangle_{q(B)}\right], b_{jj'}^* = \exp\left[\Psi(\omega_{jj'}^B) - \Psi(\sum_{j'} \omega_{jj'}^B)\right], \tag{260}$$
$$c_{ij}^* \triangleq \exp\left[\langle \ln(c_{ij})\rangle_{q(C)}\right], c_{ij}^* = \exp\left[\Psi(\omega_{ij}^C) - \Psi(\sum_j \omega_{ij}^C)\right]$$

For brevity's sake, we refer the reader to [46] for our previously studied derivations of the variational approximation of the BL-based HMM.

9.3 Experimental Setup and Results

The visible IOSB dataset consists of action videos that have been recorded at a sunny summer day of ten people; eight males and two female in the age range of 31.2±5.7 [99]. We test our proposed algorithm on three classes of the dataset; namely, film, point, and throw. Each of the classes has ten videos with sample frames shown in Fig. 3.3. A 25 frames per second frame rate of the captured actions in visible spectrum is generated by AXIS Q5534 and AXIS Q1755 cameras with a resolution of 800×600 pixels.

We represent each of the AR videos with a series of extracted histogram of optical flow (HOF) and motion boundary histogram (MBH) descriptors which may be detected using any interest point detector [100]. In our experiments, we extract the points along the motion trajectory as in [101]. This set of extracted features represent the training and testing data with a leave-one-out cross validation scheme. A HMM is then trained for each class using the aforementioned data. For the testing stage, the likelihood of each testing video sequence is calculated by the respective three trained HMMs and the class label is assigned according to the maximum resulting likelihood.

We train each HMM with each set of training features for each of the classes nine times in order to ensure robustness of the methodology on the IOSB dataset. We report our results as an average across the training times for the offline BL-based trained HMMs for benchmarking purposes. It is noteworthy to mention that the number of states and the respective number of mixture components

Table 9.1: Comparison of the accuracy of the Dirichlet (Dir), Beta-Liouville (BL), and the proposed online HMMs for the action recognition video data. Results of the proposed models are highlighted in italics.

Method	Accuracy
BL HMM (HOF)	*33.33%*
BL HMM (Horizontal MBH)	*35.00%*
BL HMM (Vertical MBH)	*38.33%*
Online Dir HMM (HOF)	*77.03%*
Online Dir HMM (Horizontal MBH)	*33.40%*
Online Dir HMM (Vertical MBH)	*64.53%*
Online BL HMM (HOF)	*62.80%*
Online BL HMM (Horizontal MBH)	*36.04%*
Online BL HMM (Vertical MBH)	*71.57%*

of the proposed BL HMM for this application have been set experimentally to $K = 2$ and $M = 2$ respectively. In the proposed online approach, we train the HMMs on 70% of the data in the batch stage before launching the incremental phase.

The results of the trained offline BL HMM models on the IOSB visible spectrum frames may be observed in Fig. 9.2 for the HOF, horizontal and vertical MBH features. Accuracy of the models is not optimum due to the characteristics of the visible spectrum that include high sensitivity to shadow, background clutter, occlusion, and changes in illumination. Nonetheless, these results are improved dramatically in the online setup due to the gradual adjustment of the parameters that allows for better fitting of the data by the proposed model. However, the BL-based HMM performs slightly better than the Dirichlet due to its superior modelling capabilities that overcome the negative covariance constraint that is enforced by the Dirichlet. Overall, the proposed system shows promise for efficient deployment of explainable machine learning models for surveillance, especially given the improvements that occur as more data is added.

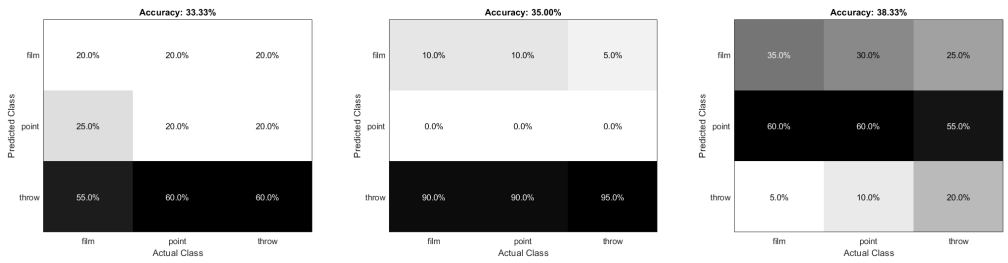


Figure 9.2: Confusion matrices of batch BL HMM trained with HOF (left), horizontal MBH (middle), and vertical MBH (right) features extracted from the IOSB visible spectrum frames.

Chapter 10

Conclusion and Future Work

One never notices what has been done; one can only see what remains to be done.

Marie Curie

10.1 Summary

A HMM is a double stochastic generative model that is appropriate for sequence data or time series modeling. It is characterized by a number of hidden states, the initial probabilities to start in each, and the associated discrete or continuous distributions of the emitted observations. It is highly suited for modeling dynamic data such as videos and for spatiotemporal object modeling.

Given the recent increased research interest in HMMs, we aimed to address five of the main modern HMM research challenges and state-of-the-art techniques to address them: (i) methods for accurate estimation of the model; (ii) choice of the appropriate emission distributions based on the nature of the data, especially proportional; (iii) dynamic determination of the structure of the HMM based in a data-driven manner for best fit; (iv) simultaneous feature selection paradigm for finite and infinite proportional HMMs; (v) incremental learning of HMM parameters for online deployment. In this thesis, we developed and implemented various novel proportional HMMs to tackle these issues successfully. Validation of the proposed models was carried out on multiple computer vision tasks. Namely, infrared action recognition, visible spectrum action recognition, multimodal action recognition, and anomaly detection in videos. We also successfully impacted

the occupancy estimation and detection research community through the introduction of a novel experimental setup for HMMs.

10.2 Conclusions

The study in this research led to several interesting new research investigations and consequent conclusions:

- We tackled the estimation and emission distribution choice problems through the derivation and implementation of a variational Bayesian learning technique that utilized the BL distribution. The use of the BL distribution for proportional time series or sequential data modeling with variational Bayesian-based HMMs was a promising expansion to the state-of-the-art. Indeed, in addition to our proposed research, this has also been proven in [94] for the incorporation of other proportional distributions into HMMs with variational learning.

- Next, we focused on developing a Maximum A Posteriori (MAP)-based approach for effective parameter estimation. The advantage of using MAP approximation instead of the traditional Baum Welch algorithm lies in its improved performance sans the computational overhead that other approaches, such as variational learning, impose. We validate the proposed models on dynamic texture classification and infrared action recognition. We compare our results with the Baum Welch approach as well as benchmark against state-of-the-art methods. Accuracy improvements clearly highlight the significance of deriving and applying the MAP approximation as well as the use of an appropriate distribution corresponding to the nature of the data.

- We tackled the difficult challenge of choosing the number of states. HMMs can be finite or infinite [178]. In finite HMMs, the number of states is usually determined as a result of an exhaustive search for the appropriate count. This can be achieved through the implementation of infinite HMMs through the means of non-parametric Bayesian methods [179].

- From experimental results, we have shown that applying proportional HMMs corresponding to the nature of the input data leads to improved model performance. That is when the

Gaussian distribution and its mixture model are no longer the ideal choice for the emission distribution modeling of the sequential data at hand. Motivated by these facts, we investigated the Dirichlet, the Beta-Liouville, and the Generalized Dirichlet distributions in infinite HMMs. Furthermore, development and testing of the proposed real-time unusual event detection also serve as interesting investigations particularly in the domains of public security and safety.

- The majority of research in HMMs has been primarily concerned with the parameter learning of the model. However, such approaches have several limitations whereby all the features are apriori assumed to have the same weight across the various mixture components as well as the HMM states. Intuitively, the higher the number of features used to represent a given dataset, the higher the expected efficiency of the model. However, some features can be noisy, redundant, or uninformative in practice and hence can hinder the clustering performance [153]. The presence of many irrelevant features introduces a bias resulting in unreliable homogeneity measures. Feature selection is the process of reducing the number of collected features to a subset of relevant ones. In addition to increasing the performance of the models, it also aids in improving model interpretation and decreasing the risk of overfitting [154]. Hence, we suggest the incorporation of a feature selection paradigm [151, 152]. This is a doubly impactful conclusion as we have incorporated it in both finite and infinite proportional HMMs.

- Most of the research on HMMs is reported in an offline setting. That is once the training of the model is completed, any new testing data is only classified with the model but the parameters do not update to benefit from the availability of new training data. On the other hand, online learning takes into account such a scenario whereby the existing model is capable of incorporating the newly classified data without having to completely retrain the model from scratch.

10.3 Future Work

Various multiple directions can be carried out as future works building on this thesis. The following is a list of possibilities:

(1) Online deployment of Generalized Dirichlet based HMMs and the incorporation of feature selection.

(2) Carrying out the framework investigations performed in this thesis for proportional data on other data types. That would require the change of the emission distribution to fit the nature and statistical properties of the data accordingly.

(3) Investigating the proposed approaches and models on other applications. For instance, applying it in some of the mentioned domains in Chapter 1.

(4) Utilizing deep learning techniques in order to enable the modelling of a large amount of data. An interesting idea would be a combination of both the proposed approaches with these techniques.

(5) Though computationally expensive, sampling methods such as the Markov Chain Monte Carlo technique are a novel venue of investigation using the proposed models.

Bibliography

[1] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models," *Energy and Buildings*, vol. 112, pp. 28 – 39, 2016.

[2] S. Ali and N. Bouguila, "Maximum a posteriori approximation of dirichlet and beta-liouville hidden markov models for proportional sequential data modeling," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 4081–4087.

[3] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[4] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., pp. 841–848. MIT Press, 2002.

[5] N. Bouguila, "Hybrid generative/discriminative approaches for proportional data modeling and classification," *IEEE Trans. on Knowl. and Data Eng.*, vol. 24, no. 12, pp. 2184–2202, Dec. 2012.

[6] J. A. Lasserre, C. M. Bishop, and T. P. Minka, "Principled hybrids of generative and discriminative models," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, Washington, DC, USA, 2006, CVPR '06, pp. 87–94, IEEE Computer Society.

[7] S. Ali and N. Bouguila, "Hybrid generative-discriminative generalized dirichlet-based hidden markov models with support vector machines," in *2019 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2019, pp. 231–2311.

[8] S. Ali and N. Bouguila, "Dynamic texture recognition using a hybrid generative-discriminative approach with hidden markov models and support vector machines," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2019, pp. 1–5.

[9] S. L. Ho and M. Xie, "The use of arima models for reliability forecasting and analysis," *Comput. Ind. Eng.*, vol. 35, no. 1–2, pp. 213–216, Oct. 1998.

[10] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, Jan 1986.

[11] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, "Fine-grained activity recognition by aggregating abstract object usage," in *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*, Oct 2005, pp. 44–51.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, Berlin, Heidelberg, 2006.

[13] R. E. Kalman, "Mathematical description of linear dynamical systems," *Journal of the Society for Industrial and Applied Mathematics Series A Control*, vol. 1, no. 2, pp. 152–192, 1963.

[14] M. Rodriguez, C. Orrite, C. Medrano, and D. Makris, "One-shot learning of human activity with an map adapted gmm and simplex-hmm," *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1769–1780, July 2017.

[15] M. Wang, S. Abdelfattah, N. Moustafa, and J. Hu, "Deep gaussian mixture-hidden markov model for classification of eeg signals," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 4, pp. 278–287, Aug 2018.

[16] J. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," *International Computer Science Institute*, vol. 4, pp. 126, 1998.

[17] S.-Z. Yu, "Hidden semi-markov models," *Artificial Intelligence*, vol. 174, no. 2, pp. 215–243, 2010, Special Review Issue.

[18] S. K. Ng, T. Krishnan, and G. J. McLachlan, *The EM Algorithm*, pp. 139–172, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[19] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the em algorithm," *SIAM review*, vol. 26, no. 2, pp. 195–239, 1984.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[21] R. D. Bock and M. Aitkin, "Marginal maximum likelihood estimation of item parameters: Application of an em algorithm," *Psychometrika*, vol. 46, no. 4, pp. 443–459, Dec 1981.

[22] B. S. Everitt, "Maximum likelihood estimation of the parameters in a mixture of two univariate normal distributions; a comparison of different algorithms," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 33, no. 2, pp. 205–215, 1984.

[23] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition," *Bell System Technical Journal*, vol. 62, no. 4, pp. 1035–1074, 1983.

[24] S. Levinson, L. Rabiner, and M. Sondhi, "Speaker independent isolated digit recognition using hidden markov models," in *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1983, vol. 8, pp. 1049–1052.

[25] J. Li, J. Y. Lee, and L. Liao, "A novel algorithm for training hidden markov models with positive and negative examples," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2020, pp. 305–310.

[26] J. A. Hartigan, *Clustering Algorithms*, John Wiley & Sons, Inc., New York, NY, USA, 99th edition, 1975.

[27] B.-J. Yoon, "Hidden markov models and their applications in biological sequence analysis," *Current genomics*, vol. 10, no. 6, pp. 402–415, 2009.

[28] D. Khiatani and U. Ghose, "Weather forecasting using hidden markov model," in *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, Oct 2017, pp. 220–225.

[29] E. Epaillard and N. Bouguila, "Proportional data modeling with hidden markov models based on generalized dirichlet and beta-liouville mixtures applied to anomaly detection in public areas," *Pattern Recognition*, vol. 55, pp. 125 – 136, 2016.

[30] E. Epaillard and N. Bouguila, "Hidden markov models based on generalized dirichlet mixtures for proportional data modeling," in *Artificial Neural Networks in Pattern Recognition*, N. El Gayar, F. Schwenker, and C. Suen, Eds., Cham, 2014, pp. 71–82, Springer International Publishing.

[31] X. Wang, H. Wu, and Z. Yi, "Research on bank anti-fraud model based on k-means and hidden markov model," in *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, June 2018, pp. 780–784.

[32] M. Haid, B. Budaker, M. Geiger, D. Husfeldt, M. Hartmann, and N. Berezowski, "Inertial-based gesture recognition for artificial intelligent cockpit control using hidden markov models," in *2019 IEEE International Conference on Consumer Electronics (ICCE)*, Jan 2019, pp. 1–4.

[33] S. Wolf, J. K. Møller, M. A. Bitsch, J. Krogstie, and H. Madsen, "A markov-switching model for building occupant activity estimation," *Energy and Buildings*, vol. 183, pp. 672 – 683, 2019.

[34] P. Kumar and M. D'Souza, "Design a power aware methodology in iot based on hidden markov model," in *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, Jan 2017, pp. 580–581.

[35] M. A. Fouad and A. T. Abdel-Hamid, "On detecting iot power signature anomalies using hidden markov model (hmm)," in *2019 31st International Conference on Microelectronics (ICM)*, Dec 2019, pp. 108–112.

[36] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid hmm/ann models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 4, pp. 767–779, 2010.

[37] A. Schlapbach and H. Bunke, "A writer identification and verification system using hmm based recognizers," *Pattern analysis and applications*, vol. 10, no. 1, pp. 33–43, 2007.

[38] A. Schlapbach and H. Bunke, "Using hmm based recognizers for writer identification and verification," in *Ninth International Workshop on Frontiers in Handwriting recognition*. IEEE, 2004, pp. 167–172.

[39] K. Tokuda, H. Zen, and A. W. Black, "An hmm-based speech synthesis system applied to english," in *IEEE Speech Synthesis Workshop*, 2002, pp. 227–230.

[40] J. Jayakumari and A. F. Jalin, "An improved text to speech technique for tamil language using hidden markov model," in *2019 7th International Conference on Smart Computing Communications (ICSCC)*, June 2019, pp. 1–5.

[41] B. Mor, S. Garhwal, and A. Kumar, "A systematic review of hidden markov models and their applications," *Archives of Computational Methods in Engineering*, May 2020.

[42] S. Amudala, S. Ali, and N. Bouguila, "Variational inference of infinite generalized gaussian mixture models with feature selection," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 120–127.

[43] Z. Song, S. Ali, N. Bouguila, and W. Fan, "Nonparametric hierarchical mixture models based on asymmetric gaussian distribution," *Digital Signal Processing*, vol. 106, pp. 102829, 2020.

[44] K. Maanicshah, S. Ali, W. Fan, and N. Bouguila, "Unsupervised variational learning of finite generalized inverted dirichlet mixture models with feature selection and component splitting," in *International Conference on Image Analysis and Recognition*. Springer, 2019, pp. 94–105.

[45] Z. Song, S. Ali, and N. Bouguila, "Bayesian learning of infinite asymmetric gaussian mixture models for background subtraction," in *International Conference on Image Analysis and Recognition*. Springer, 2019, pp. 264–274.

[46] S. Ali and N. Bouguila, "Variational learning of beta-liouville hidden markov models for infrared action recognition," in *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[47] S. Ali and N. Bouguila, "Multimodal action recognition using variational-based beta-liouville hidden markov models," *IET Image Processing*, vol. 14, pp. 4785–4794(9), December 2020.

[48] S. Ali and N. Bouguila, "Dynamic texture recognition using a hybrid generative-discriminative approach with hidden markov models and support vector machines," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Nov 2019, pp. 1–5.

[49] S. Ali and N. Bouguila, "Hybrid generative-discriminative generalized dirichlet-based hidden markov models with support vector machines," in *2019 IEEE International Symposium on Multimedia (ISM)*, Dec 2019, pp. 231–2311.

[50] S. Adams, P. A. Beling, and R. Cogill, "Feature selection for hidden markov models and hidden semi-markov models," *IEEE Access*, vol. 4, pp. 1642–1657, 2016.

[51] S. Ali and N. Bouguila, "On maximum a posteriori approximation of hidden markov models for proportional data," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, 2020, pp. 1–6.

[52] S. Ali and N. Bouguila, "Online learning for beta-liouville hidden markov models: Incremental variational learning for video surveillance and action recognition," in *27th IEEE International Conference on Image Processing (ICIP 2020)*, 2020.

[53] A. Toleikyte, L. Kranzl, and A. Müller, "Cost curves of energy efficiency investments in buildings – methodologies and a case study of lithuania," *Energy Policy*, vol. 115, pp. 148 – 157, 2018.

[54] M. Evans, S. Yu, A. Staniszewski, L. Jin, and A. Denysenko, "The international implications of national and local coordination on building energy codes: Case studies in six cities," *Journal of Cleaner Production*, vol. 191, pp. 127 – 134, 2018.

[55] J. Brooks, S. Kumar, S. Goyal, R. Subramany, and P. Barooah, "Energy-efficient control of under-actuated hvac zones in commercial buildings," *Energy and Buildings*, vol. 93, pp. 160 – 168, 2015.

[56] G. Y. Yun, H. J. Kong, H. Kim, and J. T. Kim, "A field survey of visual comfort and lighting energy consumption in open plan offices," *Energy and Buildings*, vol. 46, pp. 146 – 151, 2012, Sustainable and healthy buildings.

[57] C. Liao, Y. Lin, and P. Barooah, "Agent-based and graphical modelling of building occupancy," *Journal of Building Performance Simulation*, vol. 5, no. 1, pp. 5–25, 2012.

[58] Z. Liu, Y. Xie, K. Y. Chan, K. Ma, and X. Guan, "Chance-constrained optimization in d2d-based vehicular communication network," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5045–5058, 2019.

[59] F. Oldewurtel, D. Sturzenegger, and M. Morari, "Importance of occupancy information for building climate control," *Applied Energy*, vol. 101, pp. 521 – 532, 2013, Sustainable Development of Energy, Water and Environment Systems.

[60] X. Pan, C. S. Han, K. Dauber, and K. H. Law, "A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations," *AI & SOCIETY*, vol. 22, no. 2, pp. 113–132, Nov 2007.

[61] Y. Yamaguchi, Y. Shimoda, and M. Mizuno, "Transition to a sustainable urban energy system from a long-term perspective: Case study in a japanese business district," *Energy and Buildings*, vol. 39, no. 1, pp. 1 – 12, 2007.

[62] A. Roetzel, "Occupant behaviour simulation for cellular offices in early design stages—architectural and modelling considerations," *Building Simulation*, vol. 8, no. 2, pp. 211–224, Apr 2015.

[63] V. L. Erickson, M. . Carreira-Perpiñán, and A. E. Cerpa, "Observe: Occupancy-based system for efficient reduction of hvac energy," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2011, pp. 258–269.

[64] V. L. Erickson, M. A. Carreira-Perpiñán, and A. E. Cerpa, "Occupancy modeling and prediction for building energy management," *ACM Trans. Sen. Netw.*, vol. 10, no. 3, May 2014.

[65] B. Dong and B. Andrews, "Sensor-based occupancy behavioral pattern recognition for energy and comfort management in intelligent buildings," 2009, pp. 1444–1451, 11th International IBPSA Conference - Building Simulation 2009, BS 2009 ; Conference date: 27-07-2007 Through 30-07-2007.

[66] T. Vafeiadis, S. Zikos, G. Stavropoulos, D. Ioannidis, S. Krinidis, D. Tzovaras, and K. Moustakas, "Machine learning based occupancy detection via the use of smart meters," in *2017 International Symposium on Computer Science and Intelligent Controls (ISCSIC)*, 2017, pp. 6–12.

[67] A. Khan, J. Nicholson, S. Mellor, D. Jackson, K. Ladha, C. Ladha, J. Hand, J. Clarke, P. Olivier, and T. Ploetz, "Occupancy monitoring using environmental and context sensors and a hierarchical analysis framework," in *BuildSys '14*, M. Srivastava, Ed., United States, Nov. 2014, pp. 90–99, ACM.

[68] B. Qolomany, A. Al-Fuqaha, A. Gupta, D. Benhaddou, S. Alwajidi, J. Qadir, and A. C. Fong, "Leveraging machine learning and big data for smart buildings: A comprehensive survey," *IEEE Access*, vol. 7, pp. 90316–90356, 2019.

[69] B. Dong, D. Yan, Z. Li, Y. Jin, X. Feng, and H. Fontenot, "Modeling occupancy and behavior for better building design and operation—a critical review," *Building Simulation*, vol. 11, no. 5, pp. 899–921, Oct 2018.

[70] L. M. Candanedo, V. Feldheim, and D. Deramaix, "A methodology based on hidden markov models for occupancy detection and a case study in a low energy residential building," *Energy and Buildings*, vol. 148, pp. 327 – 341, 2017.

[71] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," *Energy and Buildings*, vol. 140, pp. 81 – 97, 2017.

[72] J. Pohle, R. Langrock, F. M. van Beest, and N. M. Schmidt, "Selecting the number of states in hidden markov models: Pragmatic solutions illustrated using animal movement," *Journal of Agricultural, Biological and Environmental Statistics*, vol. 22, no. 3, pp. 270–293, 2017.

[73] M. Ramezani and F. Yaghmaee, "A review on human action analysis in videos for retrieval applications," *Artif. Intell. Rev.*, vol. 46, no. 4, pp. 485–514, Dec. 2016.

[74] C.-B. Jin, S. Li, T. D. Do, and H. Kim, "Real-time human action recognition using cnn over temporal images for static video surveillance cameras," in *Advances in Multimedia Information Processing – PCM 2015*, Y.-S. Ho, J. Sang, Y. M. Ro, J. Kim, and F. Wu, Eds., Cham, 2015, pp. 330–339, Springer International Publishing.

[75] Z. Moghaddam and M. Piccardi, "Training initialization of hidden markov models in human action recognition," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 394–408, April 2014.

[76] M. H. Kabir, M. R. Hoque, K. Thapa, and S.-H. Yang, "Two-layer hidden markov model for human activity recognition in home environments," *International Journal of Distributed Sensor Networks*, vol. 12, no. 1, pp. 4560365, 2016.

[77] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image and Vision Computing*, vol. 60, pp. 4 – 21, 2017, Regularization Techniques for High-Dimensional Data Analysis.

[78] M. Vrigkas, C. Nikou, and I. A. Kakadiaris, "A review of human activity recognition methods," *Frontiers in Robotics and AI*, vol. 2, pp. 28, 2015.

[79] Y. Liu, Z. Lu, J. Li, C. Yao, and Y. Deng, "Transferable feature representation for visible-to-infrared cross-dataset human action recognition," *Complexity*, vol. 2018, 2018.

[80] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[81] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03*, Washington, DC, USA, 2004, ICPR '04, pp. 32–36, IEEE Computer Society.

[82] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, December 2007.

[83] C. Gao, Y. Du, J. Liu, L. Yang, and D. Meng, "A new dataset and evaluation for infrared action recognition," in *CCF Chinese Conference on Computer Vision*. Springer, 2015, pp. 302–312.

[84] F. El Baf, T. Bouwmans, and B. Vachon, "Fuzzy statistical modeling of dynamic backgrounds for moving object detection in infrared videos," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 2009, pp. 60–65.

[85] W. Gong, X. Zhang, J. Gonzàlez, A. Sobral, T. Bouwmans, C. Tu, and E.-h. Zahzah, "Human pose estimation from monocular images: A comprehensive survey," *Sensors*, vol. 16, no. 12, 2016.

[86] N. Tomašev and M. Radovanović, "Clustering evaluation in high-dimensional data," in *Unsupervised Learning Algorithms*, pp. 71–107. Springer, 2016.

[87] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Systems with Applications*, vol. 40, no. 1, pp. 200 – 210, 2013.

[88] W. Fan and N. Bouguila, "Learning finite beta-liouville mixture models via variational bayes for proportional data clustering," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. 2013, IJCAI '13, pp. 1323–1329, AAAI Press.

[89] L. Chen, D. Barber, and J.-M. Odobez, "Dynamical dirichlet mixture model," Idiap-RR Idiap-RR-02-2007, IDIAP, 2007.

[90] E. Epaillard and N. Bouguila, "Variational bayesian learning of generalized dirichlet-based hidden markov models applied to unusual events detection," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2018.

[91] S. P. Chatzis, D. I. Kosmopoulos, and T. A. Varvarigou, "Robust sequential data modeling using an outlier tolerant hidden markov model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1657–1669, Sept 2009.

[92] W. Fan and N. Bouguila, "Online learning of a dirichlet process mixture of beta-liouville distributions via variational inference," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 11, pp. 1850–1862, Nov 2013.

[93] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, no. 2, pp. 183–233, Nov. 1999.

[94] E. Epaillard and N. Bouguila, "Variational bayesian learning of generalized dirichlet-based hidden markov models applied to unusual events detection," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2018.

[95] S. P. Chatzis and D. I. Kosmopoulos, "A variational bayesian methodology for hidden markov models utilizing student's-t mixtures," *Pattern Recognition*, vol. 44, no. 2, pp. 295–306, 2011.

[96] D. J. C. MacKay, "Ensemble Learning for Hidden Markov Models," *Technical Report*, , no. 1995, pp. 0–6, 1997.

[97] Z. Ma and A. Leijon, "Bayesian estimation of Beta mixture models with variational inference," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 33, no. 11, pp. 2160–2173, 2011.

[98] C. Gao, Y. Du, J. Liu, J. Lv, L. Yang, D. Meng, and A. G. Hauptmann, "Infar dataset: Infrared action recognition at different times," *Neurocomputing*, vol. 212, pp. 36 – 47, 2016, Chinese Conference on Computer Vision 2015 (CCCV 2015).

[99] B. Hilsenbeck, D. Münch, A.-K. Grosselfinger, W. Hübner, and M. Arens, "Action recognition in the longwave infrared and the visible spectrum using hough forests," in *2016 IEEE International Symposium on Multimedia (ISM)*, Dec 2016, pp. 329–332.

[100] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *Int. J. Comput. Vision*, vol. 37, no. 2, pp. 151–172, June 2000.

[101] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2011, CVPR '11, pp. 3169–3176, IEEE Computer Society.

[102] Z. Jiang, V. Rozgic, and S. Adali, "Learning spatiotemporal features for infrared action recognition with 3d convolutional neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 309–317.

[103] Y. Liu, Z. Lu, J. Li, T. Yang, and C. Yao, "Global temporal representation based cnns for infrared action recognition," *IEEE Signal Processing Letters*, vol. 25, no. 6, pp. 848–852, June 2018.

[104] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International journal of computer vision*, vol. 103, no. 1, pp. 60–79, 2013.

[105] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, Washington, DC, USA, 2013, ICCV '13, pp. 3551–3558, IEEE Computer Society.

[106] D. Ruta and B. Gabrys, "An overview of classifier fusion methods," *Computing and Information systems*, vol. 7, no. 1, pp. 1–10, 2000.

[107] Y. Qiao and L. Weng, "Hidden markov model based dynamic texture classification," *IEEE Signal Processing Letters*, vol. 22, no. 4, pp. 509–512, April 2015.

[108] P. Saisan, G. Doretto, Ying Nian Wu, and S. Soatto, "Dynamic texture recognition," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Dec 2001, vol. 2, pp. II–II.

[109] A. B. Chan and N. Vasconcelos, "Probabilistic kernels for the classification of auto-regressive visual processes," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, June 2005, vol. 1, pp. 846–851 vol. 1.

[110] S. V. Vishwanathan, A. J. Smola, and R. Vidal, "Binet-cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes," *Int. J. Comput. Vision*, vol. 73, no. 1, pp. 95–119, June 2007.

[111] G. Doretto, D. Cremers, P. Favaro, and S. Soatto, "Dynamic texture segmentation," in *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, Washington, DC, USA, 2003, ICCV '03, pp. 1236–, IEEE Computer Society.

[112] Rene Vidal and Avinash Ravichandran, "Optical flow estimation amp; segmentation of multiple moving dynamic textures," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, June 2005, vol. 2, pp. 516–521 vol. 2.

[113] A. B. Chan and N. Vasconcelos, "Mixtures of dynamic textures," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Oct 2005, vol. 1, pp. 641–647 Vol. 1.

[114] A. B. Chan and N. Vasconcelos, "Layered dynamic textures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1862–1879, Oct 2009.

[115] V. Kellokumpu, G. Zhao, and M. Pietikinen, "Recognition of human actions using texture descriptors," *Mach. Vision Appl.*, vol. 22, no. 5, pp. 767–780, Sept. 2011.

[116] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *Int. J. Comput. Vision*, vol. 51, no. 2, pp. 91–109, Feb. 2003.

[117] A. Chan, *Beyond Dynamic Textures: A Family of Stochastic Dynamical Models for Video with Applications to Computer Vision*, Ph.D. thesis, La Jolla, CA, USA, 2008, AAI3331461.

[118] G. Atluri, A. Karpatne, and V. Kumar, "Spatio-temporal data mining: A survey of problems and methods," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 83:1–83:41, Aug. 2018.

[119] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915–928, June 2007.

[120] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "Object trajectory-based activity classification and recognition using hidden markov models," *IEEE Transactions on Image Processing*, vol. 16, no. 7, pp. 1912–1919, July 2007.

[121] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, June 1998.

[122] Y. Bai, Z. Sun, B. Zeng, J. Long, L. Li, J. V. Oliveira, and C. Li, "A comparison of dimension reduction techniques for support vector machine modeling of multi-parameter manufacturing quality prediction," *J. Intell. Manuf.*, vol. 30, no. 5, pp. 2245–2256, June 2019.

[123] S. Wang, Q. Liu, E. Zhu, F. Porikli, and J. Yin, "Hyperparameter selection of one-class support vector machine by self-adaptive data shifting," *Pattern Recognition*, vol. 74, pp. 198 – 211, 2018.

[124] S. Yue, P. Li, and P. Hao, "Svm classification:its contents and challenges," *Applied Mathematics-A Journal of Chinese Universities*, vol. 18, no. 3, pp. 332–342, Sep 2003.

[125] T. Bdiri and N. Bouguila, "Bayesian learning of inverted dirichlet mixtures for svm kernels generation," *Neural Computing and Applications*, vol. 23, no. 5, pp. 1443–1458, Oct 2013.

[126] D. R. Hardoon, C. Saunders, and J. Shawe-Taylor, "Using fisher kernels and hidden markov models for the identification of famous composers from their sheet music," *notes*, vol. 16, no. 1, pp. 16, 2005.

[127] Y. Goutsu, W. Takano, and Y. Nakamura, "Gesture recognition using hybrid generative-discriminative approach with fisher vector," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 3024–3031.

[128] W. Fan and N. Bouguila, "Dynamic textures clustering using a hierarchical pitman-yor process mixture of dirichlet distributions," in *2015 IEEE International Conference on Image Processing (ICIP)*, Sep. 2015, pp. 296–300.

[129] W. Fan, N. Bouguila, S. Bourouis, and Y. Laalaoui, "Entropy-based variational bayes learning framework for data clustering," *IET Image Processing*, vol. 12, no. 10, pp. 1762–1772, 2018.

[130] T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*, Cambridge, MA, USA, 1999, pp. 487–493, MIT Press.

[131] L. Chen, H. Man, and A. V. Nefian, "Face recognition based on multi-class mapping of fisher scores," *Pattern Recogn.*, vol. 38, no. 6, pp. 799–811, June 2005.

[132] S. Bourouis, A. Zaguia, N. Bouguila, and R. Alroobaea, "Deriving probabilistic SVM kernels from flexible statistical mixture models and its application to retinal images classification," *IEEE Access*, vol. 7, pp. 1107–1117, 2019.

[133] R. Péteri, S. Fazekas, and M. J. Huiskes, "DynTex : a Comprehensive Database of Dynamic Textures," *Pattern Recognition Letters*, vol. doi: 10.1016/j.patrec.2010.05.009, http://projects.cwi.nl/dyntex/.

[134] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, July 2002.

[135] T. Elguebaly and N. Bouguila, "Background subtraction using finite mixtures of asymmetric gaussian distributions and shadow detection," *Mach. Vis. Appl.*, vol. 25, no. 5, pp. 1145–1162, 2014.

[136] A. Sefidpour and N. Bouguila, "Spatial color image segmentation based on finite non-gaussian mixture models," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 8993–9001, 2012.

[137] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 909–926, 2008.

[138] S. R. Arashloo and J. Kittler, "Dynamic texture recognition using multiscale binarized statistical image features," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2099–2109, 2014.

[139] Q. Lou, S. Sarkhel, S. Mitra, and V. Swaminathan, "Content-based effectiveness prediction of video advertisements," in *2018 IEEE International Symposium on Multimedia (ISM)*, Dec 2018, pp. 69–72.

[140] W. Xu, Z. Miao, X. Zhang, and Y. Tian, "A hierarchical spatio-temporal model for human activity recognition," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1494–1509, July 2017.

[141] P. Heracleous, V. Tran, T. Nagai, and K. Shikano, "Analysis and recognition of nam speech using hmm distances and visual information," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1528–1538, Aug 2010.

[142] L. Du, M. Chen, J. Lucas, and L. Carin, "Sticky hidden markov modeling of comparative genomic hybridization," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5353–5368, Oct 2010.

[143] A. Giménez and A. Juan, "Embedded bernoulli mixture hmms for handwritten word recognition," in *2009 10th International Conference on Document Analysis and Recognition*, July 2009, pp. 896–900.

[144] Y. Bengio, V. . Lauzon, and R. Ducharme, "Experiments on the application of iohmms to model financial returns series," *IEEE Transactions on Neural Networks*, vol. 12, no. 1, pp. 113–123, Jan 2001.

[145] M. Cholewa and P. Głomb, "Estimation of the number of states for gesture recognition with hidden markov models based on the number of critical points in time sequence," *Pattern Recognition Letters*, vol. 34, no. 5, pp. 574 – 579, 2013.

[146] E. L. Andrade, S. Blunsden, and R. B. Fisher, "Hidden markov models for optical flow analysis in crowds," in *18th International Conference on Pattern Recognition (ICPR'06)*, 2006, vol. 1, pp. 460–463.

[147] F. Jiang, Y. Wu, and A. K. Katsaggelos, "Abnormal event detection from surveillance video by dynamic hierarchical clustering," in *2007 IEEE International Conference on Image Processing*, Sept 2007, vol. 5, pp. V – 145–V – 148.

[148] J. W. Orr, P. Tadepalli, J. R. Doppa, X. Fern, and T. G. Dietterich, "Learning scripts as hidden markov models," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[149] E. Epaillard and N. Bouguila, "Proportional data modeling with hidden markov models based on generalized dirichlet and beta-liouville mixtures applied to anomaly detection in public areas," *Pattern Recognition*, vol. 55, pp. 125 – 136, 2016.

[150] Y. Xu, Yuhui Quan, H. Ling, and H. Ji, "Dynamic texture classification using dynamic fractal analysis," in *2011 International Conference on Computer Vision*, 2011, pp. 1219–1226.

[151] T. Do, T. Hoang, V. Pomponiu, Y. Zhou, Z. Chen, N. Cheung, D. Koh, A. Tan, and S. Tan, "Accessible melanoma detection using smartphones and mobile image analysis," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2849–2864, Oct 2018.

[152] E. Yu, J. Sun, J. Li, X. Chang, X. Han, and A. G. Hauptmann, "Adaptive semi-supervised feature selection for cross-modal retrieval," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1276–1288, May 2019.

[153] S. Boutemedjet, N. Bouguila, and D. Ziou, "A hybrid feature extraction selection approach for high-dimensional non-gaussian data clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 1429–1443, 2009.

[154] W. Fan, N. Bouguila, and X. Liu, "A nonparametric bayesian learning model using accelerated variational inference and feature selection," *Pattern Analysis and Applications*, vol. 22, no. 1, pp. 63–74, Feb 2019.

[155] M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain, "Simultaneous feature selection and clustering using mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1154–1166, 2004.

[156] S. Adams and P. A. Beling, "Feature selection for hidden markov models with discrete features," in *Intelligent Systems and Applications*, Y. Bi, R. Bhatia, and S. Kapoor, Eds., Cham, 2020, pp. 67–82, Springer International Publishing.

[157] S. Adams and P. A. Beling, "A survey of feature selection methods for gaussian mixture models and hidden markov models," *Artificial Intelligence Review*, vol. 52, no. 3, pp. 1739–1779, Oct 2019.

[158] Z. Song, S. Ali, and N. Bouguila, "Background subtraction using infinite asymmetric gaussian mixture models with simultaneous feature selection," *IET Image Processing*, vol. 14, no. 11, pp. 2321–2332, 2020.

[159] X. Zhao, Y. Lin, and J. Heikkilä, "Dynamic texture recognition using volume local binary count patterns with an application to 2d face spoofing detection," *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 552–566, March 2018.

[160] X. Zhao, Y. Lin, L. Liu, J. Heikkilä, and W. Zheng, "Dynamic texture classification using unsupervised 3d filter learning and local binary encoding," *IEEE Transactions on Multimedia*, vol. 21, no. 7, pp. 1694–1708, July 2019.

[161] S. Dubois, R. Péteri, and M. Ménard, "Characterization and recognition of dynamic textures based on the 2d+t curvelet transform," *Signal, Image and Video Processing*, vol. 9, no. 4, pp. 819–830, 2015.

[162] Y. Quan, Y. Huang, and H. Ji, "Dynamic texture recognition via orthogonal tensor dictionary learning," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 73–81.

[163] S. Hong, J. Ryu, and H. S. Yang, "Not all frames are equal: aggregating salient features for dynamic texture classification," *Multidimensional Systems and Signal Processing*, vol. 29, no. 1, pp. 279–298, 2018.

[164] X. Zhao, Y. Lin, and J. Heikkilä, "Dynamic texture recognition using multiscale pca-learned filters," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 4152–4156.

[165] P. Wang, W. Li, Z. Gao, C. Tang, and P. O. Ogunbona, "Depth pooling based large-scale 3-d action recognition with convolutional neural networks," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1051–1061, May 2018.

[166] Y. Jiang, Z. Wu, J. Tang, Z. Li, X. Xue, and S. Chang, "Modeling multimodal clues in a hybrid deep learning framework for video classification," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3137–3147, Nov 2018.

[167] B. Marr, "How much data do we create every day? the mind-blowing stats everyone should read," 2019.

[168] M. Bertini, A. D. Bimbo, and L. Seidenari, "Multi-scale and real-time non-parametric approach for anomaly detection and localization," *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 320 – 329, 2012, Special issue on Semantic Understanding of Human Behaviors in Image Sequences.

[169] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 18–32, Jan 2014.

[170] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 1975–1981.

[171] A. Zaharescu and R. Wildes, "Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing," in *Proceedings of the 11th European Conference on Computer Vision: Part I*, Berlin, Heidelberg, 2010, ECCV'10, pp. 563–576, Springer-Verlag.

[172] M. Bertini, A. D. Bimbo, and L. Seidenari, "Multi-scale and real-time non-parametric approach for anomaly detection and localization," *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 320 – 329, 2012, Special issue on Semantic Understanding of Human Behaviors in Image Sequences.

[173] A. A. Sodemann, M. P. Ross, and B. J. Borghetti, "A review of anomaly detection in automated surveillance," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1257–1272, Nov 2012.

[174] J. Cheng, J. H. Park, J. Cao, and W. Qi, "Hidden markov model-based nonfragile state estimation of switched neural network with probabilistic quantized outputs," *IEEE Transactions on Cybernetics*, pp. 1–10, 2019.

[175] S. Dong, Z. Wu, P. Shi, H. Su, and T. Huang, "Quantized control of markov jump nonlinear systems based on fuzzy hidden markov model," *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2420–2430, July 2019.

[176] Jie Yang, Yangsheng Xu, and C. S. Chen, "Human action learning via hidden markov model," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 27, no. 1, pp. 34–44, Jan 1997.

[177] S. Huda, J. Yearwood, and R. Togneri, "Hybrid metaheuristic approaches to the expectation maximization for estimation of the hidden markov model for signal modeling," *IEEE Transactions on Cybernetics*, vol. 44, no. 10, pp. 1962–1977, Oct 2014.

[178] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen, "The infinite hidden markov model," in *Advances in neural information processing systems*, 2002, pp. 577–584.

[179] M. Beal, "Variational algorithms for approximate Bayesian inference," *PhD Thesis*, pp. 1–281, 2003.

[180] T. S. Ferguson, "A bayesian analysis of some nonparametric problems," *The annals of statistics*, pp. 209–230, 1973.

[181] J. Sethuraman, "A constructive definition of the dirichlet prior," *Statistica Sinica*, vol. 4, pp. 639–650, 01 1994.

[182] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[183] J. Paisley and L. Carin, "Hidden markov models with stick-breaking priors," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3905–3917, Oct 2009.

[184] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 1975–1981.

[185] M. Bertini, A. Del Bimbo, and L. Seidenari, "Multi-scale and real-time non-parametric approach for anomaly detection and localization," *Comput. Vis. Image Underst.*, vol. 116, no. 3, pp. 320–329, Mar. 2012.

[186] F. B. Lung, M. H. Jaward, and J. Parkkinen, "Spatio-temporal descriptor for abnormal human activity detection," in *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, May 2015, pp. 471–474.

[187] K. Cheng, Y. Chen, and W. Fang, "Gaussian process regression-based video anomaly detection and localization with hierarchical feature representation," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5288–5301, Dec 2015.

[188] D. Singh and C. K. Mohan, "Graph formulation of video activities for abnormal activity recognition," *Pattern Recognition*, vol. 65, pp. 265 – 272, 2017.

[189] V. Kaltsa, A. Briassouli, I. Kompatsiaris, L. J. Hadjileontiadis, and M. G. Strintzis, "Swarm intelligence for detecting interesting events in crowded environments," *IEEE Transactions on Image Processing*, vol. 24, no. 7, pp. 2153–2166, July 2015.

[190] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 18–32, 2013.

[191] S. Zhou, W. Shen, D. Zeng, M. Fang, Y. Wei, and Z. Zhang, "Spatial-temporal convolutional neural networks for anomaly detection and localization in crowded scenes," *Signal Processing: Image Communication*, vol. 47, pp. 358 – 368, 2016.

[192] M. Sabokrou, M. Fathy, M. Hoseini, and R. Klette, "Real-time anomaly detection and localization in crowded scenes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2015, pp. 56–62.

[193] E. Epaillard and N. Bouguila, "Proportional data modeling with hidden markov models based on generalized dirichlet and beta-liouville mixtures applied to anomaly detection in public areas," *Pattern Recognition*, vol. 55, pp. 125 – 136, 2016.

[194] S. Ali and N. Bouguila, "Hybrid generative-discriminative generalized dirichlet-based hidden markov models with support vector machines," in *2019 IEEE International Symposium on Multimedia (ISM)*, Dec 2019, pp. 231–2311.

[195] S. Adams, P. A. Beling, and R. Cogill, "Feature selection for hidden markov models and hidden semi-markov models," *IEEE Access*, vol. 4, pp. 1642–1657, 2016.