Unsupervised and Semi-Supervised Learning
Series Editor: M. Emre Celebi

Nizar Bouguila
Wentao Fan
Manar Amayri *Editors*

Hidden Markov Models and Applications

Springer

Unsupervised and Semi-Supervised Learning

Series Editor
M. Emre Celebi, Computer Science Department, Conway, AR, USA

Springer's Unsupervised and Semi-Supervised Learning book series covers the latest theoretical and practical developments in unsupervised and semi-supervised learning. Titles – including monographs, contributed works, professional books, and textbooks – tackle various issues surrounding the proliferation of massive amounts of unlabeled data in many application domains and how unsupervised learning algorithms can automatically discover interesting and useful patterns in such data. The books discuss how these algorithms have found numerous applications including pattern recognition, market basket analysis, web mining, social network analysis, information retrieval, recommender systems, market research, intrusion detection, and fraud detection. Books also discuss semi-supervised algorithms, which can make use of both labeled and unlabeled data and can be useful in application domains where unlabeled data is abundant, yet it is possible to obtain a small amount of labeled data.

Topics of interest in include:

– Unsupervised/Semi-Supervised Discretization
– Unsupervised/Semi-Supervised Feature Extraction
– Unsupervised/Semi-Supervised Feature Selection
– Association Rule Learning
– Semi-Supervised Classification
– Semi-Supervised Regression
– Unsupervised/Semi-Supervised Clustering
– Unsupervised/Semi-Supervised Anomaly/Novelty/Outlier Detection
– Evaluation of Unsupervised/Semi-Supervised Learning Algorithms
– Applications of Unsupervised/Semi-Supervised Learning

While the series focuses on unsupervised and semi-supervised learning, outstanding contributions in the field of supervised learning will also be considered. The intended audience includes students, researchers, and practitioners.

** Indexing: The books of this series indexed in zbMATH **

Nizar Bouguila • Wentao Fan • Manar Amayri
Editors

Hidden Markov Models and Applications

Springer

Editors
Nizar Bouguila
Concordia Institute for
Information Systems Engineering
Concordia University
Montreal
QC
Canada

Manar Amayri
Grenoble Institute of Technology
Grenoble
France

Wentao Fan
Department of Computer Science
and Technology
Huaqiao University
Xiamen
China

ISSN 2522-848X ISSN 2522-8498 (electronic)
Unsupervised and Semi-Supervised Learning
ISBN 978-3-030-99141-8 ISBN 978-3-030-99142-5 (eBook)
https://doi.org/10.1007/978-3-030-99142-5

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The quantity and diversity of generated data (text, images, videos, audio files, etc.) continue to increase. This has been mainly fueled by the increasing use of social media platforms and the integration of low-cost sensors in a variety of digital systems. The extraction of useful knowledge from these data and its modeling has been the topic of extensive research in the past, and many advanced data mining and machine learning techniques and tools have been developed over the years. Since roughly 15 years, much has changed in the field of machine learning. The most spectacular developments have been in the area of deep learning which has shown state-of-the-art results in many challenging applications. Recent advances have been spectacular and have captured much popular imagination. At the same time other machine learning approaches continue to evolve. This is especially true in the case of learning models dedicated to sequential data. Hidden Markov Models (HMMs) are one of the most fundamental and largely applied statistical tools for modeling sequential data. They have been viewed for a longtime as the workhorse model for statistical time series analysis, have attracted interest in the machine learning community, have been widely used in many fields, and have provided excellent modeling results. HMMs provide solid statistical inference procedures in areas as diverse as computer vision, multimedia processing, speech recognition, genomics, machine translation, pattern recognition, energy and buildings, transportation systems, and finance. Even after more than five decades of research works on HMMs, significant research problems remain. Examples include inference, selection of the hidden state space's cardinality, model selection, feature selection, and online learning. The goal of this edited book is to present some recent works that try to tackle these problems while demonstrating the merits of HMMs in a variety of applications from different domains. The book contains 11 chapters tackling and discussing different but complementary challenging problems related to HMMs deployment in a variety of scenarios. The first chapter gives an introduction to HMMs in a simple manner by presenting various basic concepts, such as the well-known Baum Welch and Viterbi algorithms, via an interesting application that concerns occupancy estimation in smart buildings. In Chap. "Bounded Asymmetric Gaussian Mixture-Based Hidden Markov Models", Xian et al. describe a framework

that integrates the bounded asymmetric Gaussian mixture model into HMMs. A detailed inference and parameters learning approach is proposed and applied to occupancy estimation as well as human activity recognition. In Chap. "Using HMM to Model Neural Dynamics and Decode Useful Signals for Neuroprosthetic Control", Diomedi et al. deploy HMM to model neural dynamics and decode useful signal for neuroprosthetic control. Detailed simulations and experiments are presented and discussed in that chapter. An interesting computer vision application, of discrete HMMs, that concerns fire detection in images is described in Chap. "Fire Detection in Images with Discrete Hidden Markov Models" by Ali et al. The authors in Chap. "Hidden Markov Models: Discrete Feature Selection in Activity Recognition" investigate the problem of indoor activities recognition, by considering only ambient sensors, using HMMs and feature selection. Extensive simulations and analysis are provided using a challenging data set. Unlike previous chapters in which HMMs learning is based mainly on frequentist approaches, Chap. "Bayesian Inference of Hidden Markov Models using Dirichlet Mixtures" provides a fully Bayesian approach in the context of a Dirichlet-based HMM. The proposed approach is based on reversible jump Markov chain Monte Carlo sampling and is validated using video and speech data and compared with several benchmark models. Chapter "Online Learning of Inverted Beta-Liouville HMMs for Anomaly Detection in Crowd Scenes" deals with the challenging problem of HMMs online learning where the authors considered the special case of semi-bounded positive vectors modeling. The proposed model considers inverted Beta-Liouville mixtures as emission probabilities and is learned using expectation maximization framework. Its merits are shown thanks to a challenging application, namely anomaly detection in crowd scenes. The same HMM model is estimated in Chap. "A Novel Continuous Hidden Markov Model for Modeling Positive Sequential Data" via a variational approach providing a compromise between purely Bayesian and frequentist learning. Several real data sets have been used to validate the proposed methodology. Like some of the previous chapters, Chap. "Multivariate Beta-Based Hidden Markov Models Applied to Human Activity Recognition" tackled the activity recognition problem, yet using a novel multivariate Beta-based HMM architecture. Two learning approaches have been proposed based on maximum likelihood estimation and variational learning. The same HMM model was considered in Chap. "Multivariate Beta-Based Hierarchical Dirichlet Process Hidden Markov Models in Medical Applications", but within a nonparametric Bayesian framework using hierarchical Dirichlet processes (HDPs). The resulting infinite model is applied to the activity recognition task. Chapter "Shifted-Scaled Dirichlet Based Hierarchical Dirichlet Process Hidden Markov Models with Variational Inference Learning" considers also a nonparametric Bayesian approach using HDPs in the context of shifted-scaled Dirichlet HMMs. The approach is applied successfully to activity recognition and texture modeling.

Montreal, QC, Canada — Nizar Bouguila
Xiamen, China — Wentao Fan
Grenoble, France — Manar Amayri

Contents

Contributors

Samr Ali Concordia University, Montreal, QC, Canada
Global Artificial Intelligence Accelerator (GAIA), Ericsson, Montreal, QC, Canada

Manar Amayri G-SCOP Laboratory, Grenoble Institute of Technology, Grenoble, France

Muhammad Azam Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

Ali Baghdadi Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

Nizar Bouguila Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

Oumayma Dalhoumi Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

Stefano Diomedi Department of Biomedical and Neuromotor Sciences, University of Bologna, Bologna, Italy

Wentao Fan Department of Computer Science and Technology, Huaqiao University, Xiamen, China

Patrizia Fattori Department of Biomedical and Neuromotor Sciences, University of Bologna, Bologna, Italy

Kostas Hadjidimitrakis Department of Biomedical and Neuromotor Sciences, University of Bologna, Bologna, Italy

Wenjuan Hou Instrumental Analysis Center, Huaqiao University, Xiamen, China

Narges Manouchehri Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

Rim Nasfi Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

Zachary Patterson Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
Alma Mater Research Institute for Human-Centered Artificial Intelligence, University of Bologna, Bologna, Italy

Md. Hafizur Rahman Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

Francesco Edoardo Vaccari Deptartment of Biomedical and Neuromotor Sciences, University of Bologna, Bologna, Italy

Ravi Teja Vemuri Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

Zixiang Xian Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

A Roadmap to Hidden Markov Models and a Review of Its Application in Occupancy Estimation

Samr Ali and Nizar Bouguila

1 Introduction

Hidden Markov models (HMMs) 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 [1]. 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 [2]. 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 model has its own properties and advantages that we summarize shortly.

Discriminative models usually achieve superior classification accuracy results due to their primary learning objective of the boundary between classes [3]. These

S. Ali
Concordia University, Montreal, QC, Canada

Global Artificial Intelligence Accelerator (GAIA), Ericsson, Montreal, QC, Canada
e-mail: al_samr.ali@ericsson.com; samr.ali@ericsson.com

N. Bouguila (✉)
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca

N. Bouguila et al. (eds.), *Hidden Markov Models and Applications*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-99142-5_1

include the famous support vector machines (SVMs) 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 [2, 4] for further discussions. Hybrid models with HMMs are also possible such as in [5, 6]; 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.

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 [7, 8].

HVACL (Heating, Ventilation, Air Conditioning, and Lighting) systems utilize about half of this amount in industrialized countries [9, 10]. 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 [11]. 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 [12].

For highest control efficiency, a real-time input of occupancy information to the systems is required [13]. Real-time occupancy estimation is essential in evacuation of buildings and other emergencies [14]. 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 [15, 16].

1.1 Objectives

The first objective of this chapter is to assimilate an intuitive introduction into the insides and background of HMMs. Ergo, the aim is to simplify the concepts for an interested reader and make connections between the various aspects of this interesting technique. It is also imperative to mention that many excellent resources exist for HMMs and we draw from them collectively. Additionally, we also incorporate further directions of research and revolutionize the texts for a modern take on the subject. All in all, this assembled guide provides a thorough explanation of HMMs for beginners and practitioners alike. It is our aspiration that this chapter becomes a reference for the next generation of researchers in this field.

Another objective that this chapter tackles is to review the literature for the application of HMMs in occupancy estimation and prediction in smart buildings. Occupancy estimation refers to finding the true number of occupants in buildings. While papers exist that review the application of machine learning techniques in general for occupancy estimation in smart buildings [17–22], none undertake the specific HMM technique independently and hence a comprehensive review remains lacking. Hence, we endeavor to present an exhaustive discussion of relevant papers.

It is noteworthy to mention that we also survey the application of HMMs in occupancy detection. Occupancy detection is closely related to occupancy estimation. It is defined as identifying whether the area space is occupied by humans or not. As such, it generalizes the occupancy estimation problem to only two levels. This may sound trivial; however, it assuredly is a research-worthy problem with significant impact on the energy consumption of buildings. Indeed, it has been established that energy consumption in smart buildings can be reduced by 40% by only performing occupancy detection [23–25]. We also identify limitations in the employment of HMMs in occupancy estimation and potential general solutions for this interesting application.

1.2 Outline

This chapter is organized as follows. Section 2 introduces HMMs, describes its various model variations, and presents the mathematical formulations as well as some of the applications of HMMs. Section 3 presents the application of occupancy estimation and discusses the application of the models in the literature and its impact. Finally, Sect. 4 concludes the chapter.

2 Hidden Markov Models

In this section, we introduce the HMM and present its various aspects. We begin with an overview of the model in Sect. 2.1 and discuss its origin and assumptions. We then evolve our description to divulge the topologies of HMMs in Sect. 2.2. Next, we examine the Gaussian mixture model (GMM) and its famous Expectation–Maximization (EM) algorithm in Sect. 2.3 as a building block for the upcoming analysis of HMMs. In Sect. 2.4, we disclose the mathematical formulations for the learning of its parameters. Then, in Sect. 2.5, 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 Sect. 2.1). Finally, we also briefly explore applications of HMMs in Sect. 2.6. 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.

2.1 Overview

HMMs are one of the most popular statistical methods used in sequential and time series probabilistic modeling [26, 27]. A HMM is a well-received double stochastic model that uses a compact set of features to extract underlying statistics [1]. 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 twentieth century. Late 1960s and early 1970s then saw a boom of papers by *Leonard E. Baum* and other researchers who introduced and addressed its statistical techniques and modeling [26]. It allows us to maintain generality while relaxing the independent identically distributed assumption [28].

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. 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 [29] whose description is out of the scope of this report. 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 $\Xi = \{\Xi_{it}(m) = P(X_t = \xi_m \mid s_t = i)\}$, where $[m, t, i] \in [1, M] \times [1, T] \times [1, K]$, and the set of all possible discrete observations $\xi = \{\xi_1, \ldots, \xi_m, \ldots, \xi_M\}$. 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 $\varkappa = (\mu, \Sigma)$ [26, 30, 31]. 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

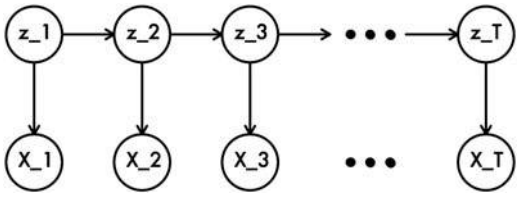


Fig. 1 A typical hidden Markov chain structure representation of a time series where z_1 denotes the first hidden state z_1 and X_1 denotes the corresponding observed state X_1. This is shown accordingly for a time series of length T

components in set $L = \{m_1, \ldots, m_M\}$. 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 [32]:

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

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

This is known as the *Limited Horizon* assumption such that state s_t has a sufficient representative summary of the past in order to predict the future.

Assumption 2 Given the t-th hidden variable, the t-th observation is independent of other variables such that:

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

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 [26]: evaluation or likelihood, estimation or decoding, and training or learning. These are described as follows:

1. **Evaluation problem:** It 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 $X = X_1, X_2, \ldots, X_T$ with length T given a HMM model Λ.
2. **Decoding problem:** It finds the optimum state sequence path $I = i_1, i_2, \ldots, i_T$ for an observation sequence X. This is mathematically $\mathbf{s}^* = \text{argmax}_{\mathbf{s}}\, P(\mathbf{s} \mid X, \Lambda)$.
3. **Learning problem:** It 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 $\Lambda^* = \text{argmax}_{\Lambda}\, P(X \mid \Lambda)$.

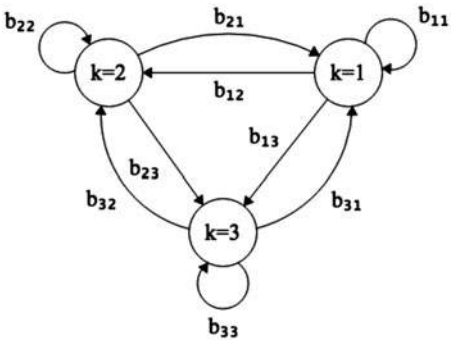


Fig. 2 A HMM transition diagram with three states

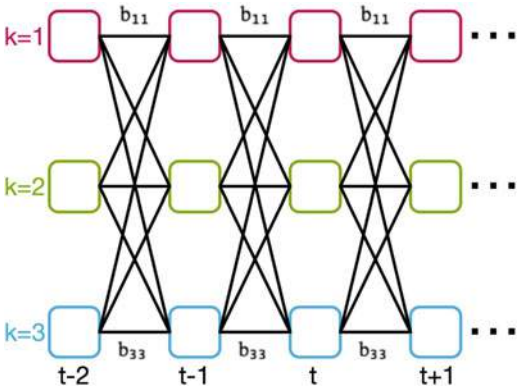


Fig. 3 Lattice or trellis HMM structure, which is a representation of the hidden states

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. 2. Figure 3 shows transitions then when they become trellis or lattice.

2.2 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 the following:

- **Hidden Markov Model (HMM):** It is introduced in Sect. 2.1, 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):** It 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.

- **Factorial Hidden Markov Model (FHMM):** It 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):** It 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):** It can explicitly model the longer-range correlations of sequential data by adding direct stochastic dependence among observations.
- **Non-Stationary HMM (NSHMM):** It captures state duration behavior by defining a set of dynamic transition probability parameters. It can model state duration probabilities explicitly as a function of time.
- **Hierarchical HMM (HHMM):** It has multi-level states that describe a sequence of inputs 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 nth-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:** In this model, 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 regard to the data for a data-based approach. We note a depiction of this in both Figs. 2 and 3 where any of the states can be visited from any other state.
2. **Left-to-Right HMM:** It requires that transitions can only be made from the current state to 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. 4.

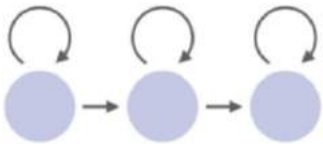


Fig. 4 A left-to-right HMM topology with three states

The structure of the HMM may also vary in regard 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. This is a design choice and in infinite models, which are outside the scope of this chapter, is of imperative significance.

Finally, we briefly bring to the attention of the reader a recent research direction that has focused on proposing new HMM models for a data-driven approach. In particular, emission distributions of the model are traditionally chosen to 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 $(-\infty, \infty)$. Indeed, other distributions have proven to be perform better in terms of fitted models in these instances [33–36].

It naturally follows that would be the circumstance in time-based probabilistic modeling using HMMs. This was proven to be true in multiple types of data such as: Dirichlet, Generalized Dirichlet, and Beta Liouville-based HMMs for proportional data [37, 38], inverted Dirichlet-based HMM for positive data [39], and von Mises–Fisher-based HMM for spherical data [40]. Furthermore, the case of mixed data (simultaneous continuous and discrete data) has also been recently investigated in [41].

2.3 *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 Sect. 2.6 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 analytic purposes. This in turn results in finding the optimum set of parameters, Θ^*, that best fit 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 dataset. 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 [42–46] 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,

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

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:

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

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

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

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

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

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 step that is part of the EM step, which is the maximization step.

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

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

(continued)

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:

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

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:

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

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

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

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

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

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

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

2.4 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 [47, 48]. In the context of HMMs, this algorithm is of extreme importance [26]. 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 [49].

This may be applied to the discrete as well as the continuous case. In this chapter, we focus on the latter and further develop Sect. 2.3 for the computation of such continuous emission distributions. The discrete case is a simplification of the continuous case 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 [50].

The forward algorithm solves the first problems that are posed for HMM as discussed in Sect. 2.1, i.e., the evaluation problem. 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 $\rho_t(i) = P(X_1, X_2, \ldots, X_t, i_t = s_i \mid \Lambda)$, 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, $i = 1, 2, \ldots, K$, at time t and sum the product over all the K possible states:

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

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 that is considerably less than a naive direct calculation approach. A graphical depiction of the forward algorithm can be observed in Fig. 5.

Figure 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

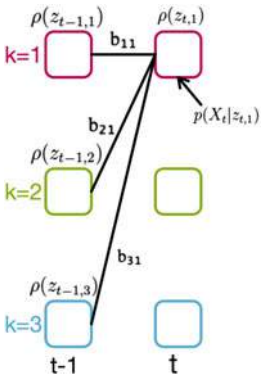


Fig. 5 Graphical representation of the evaluation of the ρ variable of the forward algorithm in a HMM lattice fragment

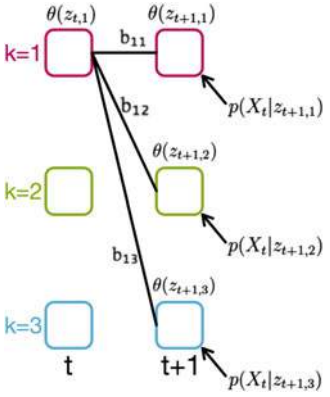


Fig. 6 Graphical representation of the evaluation of the β variable of the backward algorithm in a HMM lattice fragment

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 $\beta_t(i) = P(X_{t+1}, X_{t+2}, \ldots, X_T, i_t = s_i \mid \Lambda)$ 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:

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

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 in state s_i at time t, given Λ and X. Also, because of the Markovian conditional assumption, we can denote the following:

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

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 defined as a GMM, then we need to define another probability of the generation of X_t from the ℓth component of the ith GMM 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 GMM in Sect. 2.3 enables us to easily derive the update equations needed, which 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 oth sequence of length T_o, then the update equations are the summation across all sequences. This may be denoted by the following:

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

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

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

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

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

2.5 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. 7.

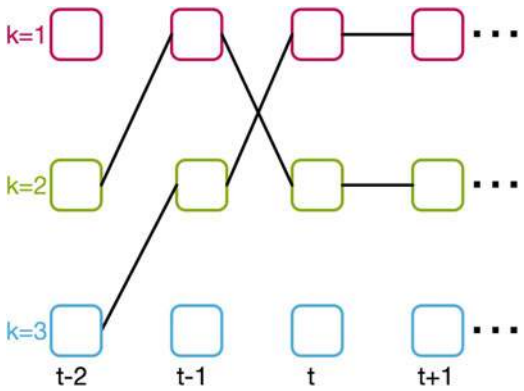


Fig. 7 Graphical representation of two probable pathways in a HMM lattice fragment. The objective of the Viterbi algorithm is to find the most likely one

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

$$\text{For } 2 \leqslant t \leqslant T, 1 \leqslant i' \leqslant K \tag{52}$$

$$\delta_t(i') = \max_{1 \leqslant i \leqslant K} [\delta_{t-1}(i) b_{ii'}] \, \Xi_{i'}(X_t) \tag{53}$$

$$\psi_t(i') = \operatorname{argmax}_{1 \leqslant i \leqslant K} [\delta_{t-1}(i) b_{ii'}] \tag{54}$$

3. Termination

$$P^* = \max_{1 \leqslant i \leqslant K} [\delta_T(i)] \, i_T^* = \operatorname{argmax}_{1 \leqslant i \leqslant K} [\delta_T(i)] \tag{55}$$

4. State sequence path backtracking

$$i_t^* = \psi_{t+1}(i_{t+1}^*), \text{ for } t = T-1, T-2, \ldots, 1 \tag{56}$$

This finalizes our mathematical discussions of the famous HMMs.

2.6 Applications

Early applications of this powerful model were in speech-related application 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 [51] for a study of HMMs in a variety of biological applications. Forecasting weather may also be performed utilizing HMMs such as in [52].

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 [53, 54] or even to detect fraud in bank transactions [55]. HMMs are also applicable in gesture recognition. An example is artificially intelligent cockpit control in [56]. You may then infer that HMMs would also shine whenever spatio-temporal analysis is carried out due to the nature of its composition.

As we will also discuss in Sect. 3, HMMs are 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 [57]. A method for efficient power usage is also proposed in [58] and another for power signature anomaly detection in [59].

Similar to speech recognition, HMMs are highly preferred in natural language processing and its subfields. Examples include recognition of handwritten characters [60], writer identification and verification systems [61, 62], and speech synthesis for the English language [63] and recently for Tamil [64]. We also refer an interested reader to [65] for a systematic survey of the applications of HMMs.

As a concluding remark, we have already covered multiple research venues within our discussion; nonetheless, many remain. For instance, 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. An interested reader is referred to [66] for a recent investigation of such a setup for HMMs.

Another potential expansion 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. Latest published articles include the variational inference such as in [38, 40] as well as Maximum A Posteriori framework in [67, 68].

3 Survey of the Employment of Hidden Markov Models in Occupancy Estimation

In this section, we address the second question that we posed in the Introduction. In particular, we review the application of HMMs in occupancy estimation of smart buildings. Table 1 summarizes the list of papers that we survey. We also briefly address the limitations of HMM deployment in occupancy estimation as well as potential future areas of research and improvements in Sect. 3.1.

One of the early papers to report large-scale deployment of a sensor network for occupancy estimation is [75]. The testbed consisted of an extensive set of sensors: CO_2, CO, TVOC, PM2.5, temperature, humidity, light, PIR, and sound that have been deployed in nodes. The ground truth labels were collected with cameras, whereas HMMs were trained for the estimation of occupancy with an average accuracy of 73%. The data collection took place in 2008 and feature selection was also taken into consideration. An important conclusion that this study also presented is the realistic modeling of occupancy estimation with HMM in comparison with other machine learning techniques such as the investigated support vector machines and artificial neural networks. The smaller scale version of the same study reported an 80% accuracy for HMM where acoustic features were not found as significant by the feature selection algorithm [76].

Later on, 100 sensor points were utilized in [77] for building a real-time model predictive control for building heating and cooling systems. This was again based on the occupancy estimation of the smart building as well as the ambient environment measurements. This is a recommended paper for feature extraction from the sensors for the training of machine learning techniques and has extensive details regarding the network design in the solar house where the study was carried out.

A HMM was utilized for occupancy estimation, while its extension the semi-hidden Markov model was employed for the duration estimation of occupants. A closely related investigation concluded that such an experimental setup results in a 17.8% reduction in the measured energy in the experiment test bed [78]. Further analysis of the energy consumption was also incorporated in this study.

A PIR is utilized in [71] for real-time occupancy estimation with HMMs in smart systems. The parameters of the HMM are learned via a simple EM algorithm within an online deployment. This enables a better estimation of the parameters with an increased efficiency over time.

In [73], authors investigate the incorporation of the location of the occupant as well as his/her motion patterns. The designed location-aware HMM in the study indeed improves the performance over conventional HMMs (up to 10%). Motion, temperature, humidity, lighting, CO_2, pressure, and sound level sensors were used to collect the environmental conditions and the motion activities. These represent the feature space that is then dynamically adapted by the proposed location-aware HMM. A leave-one-out cross validation schema was used (in relation to days, though the sampling was performed with a 3-minute time quantum). Feature selection was also carried out with the Pearson correlation coefficient.

Table 1 A list of the papers detailed in this chapter for occupancy detection and estimation in smart buildings with the respective sensors utilized and the variant(s) of the hidden Markov model (HMM). In the table, HMM represents the traditionally applied Gaussian Mixture Model-based HMM

Paper	Sensors	HMM algorithm(s)
[69]	Passive infrared (PIR)	Inhomogeneous hidden Markov models with softmax regression model
[70]	PIR	Markov chain
[71]	PIR	HMM
[72]	Pressure, temperature, humidity, and light levels	Occupant-based deployed HMM
[73]	Motion, temperature, humidity, lighting, CO_2, pressure, and sound levels	Location-aware and conventional hidden Markov models with feature selection
[74]	CO_2, electricity consumption, and light	HMM
[75]	CO_2, CO, TVOC, PM2.5, temperature, humidity, light, PIR, and sound	HMM with feature selection
[76]	CO_2, CO, TVOC, PM2.5, temperature, humidity, light, PIR, and sound	HMM with feature selection
[77]	CO_2, acoustics, motion, light, and local weather	HMM and hidden semi-Markov model
[78]	CO_2, acoustics, motion, light, and local weather	HMM and hidden semi-Markov model
[79]	Reed switches, pressure, PIR, mercury contacts, flush	Dynamic hidden semi-Markov model
[80]	Ultrasonic	HMM
[81]	Ultrasonic	Inhomogeneous HMM
[82]	CO_2, dew point temperature, and power consumption	HMM
[83]	Temperature, humidity, light, and CO_2	HMM
[84]	CO_2, temperature, relative humidity, acoustics, light, and motion	HMM and hidden semi-Markov model
[39]	Luminance, CO_2 concentration, relative humidity, temperature, motion, power consumption, window and door position, and acoustic pressure	Inverted Dirichlet-based HMM
[85]	Acoustics, light, motion, CO_2, temperature, and relative humidity	HMM and hidden semi-Markov model

An interesting model is proposed by Gomez Ortega et al. [79] where a dynamic hidden semi-Markov model is utilized for the problem of occupancy detection. However, in contrast to the traditional hidden semi-Markov model, the state duration is dynamic and the model is also capable of handling partially available observations. Overall, the performance of the proposed model was found be of significant higher accuracy (98%) than the conventional HMM and hidden semi-Markov model techniques (65.6% and 91.7%, respectively). It is also noteworthy to

mention that the incorporated weighted sensor approach appears to be of merit in relation to inclusion of sensors of various natures.

In [80], ultrasonic sensors are used for occupancy detection. HMMs were found to be robust even to unfavorable situations in real-time applications. Another occupancy detection system with HMM is proposed in [81]. The system was trained with the prior of occupants' behavior input for the computation of the posterior probability. It was tested in a real-world scenario where it performed well. Nonetheless, it is important to mention that such computation discredits a large portion of the inherent structure of HMMs as it confines its states by user intervention rather than allowing the dynamic flexibility of the mathematical formulations of the Baum Welch algorithm to shine through.

On the other hand, another approach has been recently suggested for the occupancy detection and estimation as a general case using HMMs. In particular, [72] proposes a methodology whereby the states no longer represent a particular number of occupants or a state of occupancy, but rather an entire model does. That is a model is trained for each level of occupancy in the estimation or the detection problems. This occupant-based deployment of HMMs opens up the field for scalable and flexible frameworks where an addition of level of occupancy will not interfere with pre-existing ones. Occupancy detection is also now explicitly formulated 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.

It is also noteworthy to mention the particular suitability of HMM within time-aware model for occupancy detection. This claim has been proven in [83], in which the authors compared the performance of HMM with linear regression, K-nearest neighbor, classification and regression tree, random forest, and stochastic gradient descent. The latter represents another suitable model.

Chaney et al. [82] presents a methodology for handling the challenges of feature extraction as well as sensor fusion within the occupancy estimation problem. The paper uses HMMs for occupancy detection and then infers the behavior of the occupant(s) for profiling the power consumption. Further analysis of load shifts as well as the benefits of occupants' active engagement in demand response toward a wholesome smart grid vision are also discussed. Overall, this paper serves to incorporate the HMM within the larger image of more efficient energy consumption in smart buildings as well as present the relatively untapped potential of real-time occupancy detection systems.

This is further supported by the findings of [84] where a HMM was used for occupancy estimation 18.5% energy savings were achieved in the space upon incorporation of the proposed model within the conducted simulation. Hidden semi-Markov model with exponential distribution functions also proved to be effective in modeling the associated durations. A closely related work suggests a 30% saving in energy though the investigation was carried out in a conference room [85].

In [39], the authors investigate more efficient representation of the emission distribution through a data-driven approach. In particular, they propose an inverted Dirichlet-based HMM and investigate its performance across various applications;

one of which is occupancy estimation. Within the task, they compare the performance of their proposed model versus other traditional HMMs. This represents an interesting venue of research as this setup indeed improves the performance, even within the classically deployed framework where states represent the level of occupancy.

Predicting occupancy is another task where HMMs are also applicable. This task refers to a futuristic classification or making an educated guess of the future value based on the trained model. Authors of [74] focused on utilizing only environmental data for an indirect approach of occupancy estimation with machine learning approaches. This is in contrast to direct approaches such as PIR motion detectors, video cameras, or radio-frequency identification (RFID) technologies for monitoring the occupants. Hence, all the latter were excluded and the study concluded that HMMs were suitable for the prediction of occupancy, whereas they chose decision trees for the estimation with an information gain criterion.

In another instance, a new Markov chain was developed with superior results in comparison to other machine learning models [70]. These included artificial neural network and support vector regression in various spatial scenarios. This was at both room level and house level with temporal levels: 15-minute, 30-minute, 1-hour, and 24-hour ahead forecasts.

As discussed thus far, HMMs are ubiquitous in the field of occupancy estimation and prediction [70, 71, 86]. Nonetheless, they are based on implicit assumption of time-invariant transition probability matrix of the hidden states in the HMM. However, this is not necessarily applicable to the dynamic changes that characterize indoor occupancy [16, 87]. Hence, in [69], authors investigate the time-dependent transitions between the different states of a HMM for the modelling of occupancy estimation. In particular, this character of real-time indoor occupancy system response is modelled with an inhomogeneous HMM with a softmax regression-based emission probability. The features are extracted from the collected data of fast-sampling infrared array sensors for both online and offline estimation.

Furthermore, and for the completeness of this review, we also briefly examine other relevant surveys in the literature that touch upon similar topics. For instance, [65] presents a systematic review of hidden Markov models in various applications. The survey briefly describes the various variants of HMMs and then discusses the respective prevalence of the model in applications as well as relevant papers. An occupancy estimation paper is mentioned.

Shen et al. [88] also covers occupancy detection in its various faces. The paper describes both conventional occupancy detection approaches and the modern ambient sensor-based ones. It does also incorporate several HMM papers. However, as is often the case in depth analysis of the topic is lacking due to the broad scope of the paper.

In contrast, [17] is an occupancy detection dedicated survey that touches upon various topics in that subject. This is a highly beneficial resource for those seeking an introduction to occupancy detection field. The most common sensors and their limitations are even described. The mention of HMMs is confined within

the *Occupancy estimation methods* section, in particular, the estimation through prediction techniques.

Even a more specific survey in the topic is [18] that focuses on data analytics approaches. This translates to algorithms for feature extraction and data preprocessing as well as review of machine learning techniques in general for occupancy detection. Benchmarking and performance evaluation metrics are also introduced. Hidden Markov models are only mentioned in terms of some papers in the literature under the *Probabilistic graphical models* section.

To face the futuristic challenges of minimal intrusive occupancy monitoring and effective data fusion techniques, authors of [89] survey existing papers for smart buildings of which HMMs are a key player. Interesting side track topics include future fields of research in this topic such as utilizing the existing infrastructure in terms of WiFi as well as localization with an emphasis on multimodal data fusion and privacy preservation.

Indeed, privacy concerns are such a major concern in this topic as previously mentioned that authors of [19] propose a novel method in addition to their survey of the topic. The paper deliberates the various techniques used for occupancy detection in terms of the actual means to do so. For instance, that includes WiFi, Bluetooth, PIR sensors, sensor fusion, etc. In relation to its HMM mention, it only occurs once very briefly in relation to a paper in the PIR section titled *Occupancy measurement via passive infra-red (PIR) sensor*. Sensors are also the major theme in [22] with two papers in relation to HMMs.

Markov chains are dedicated a subsection in [20], which aims to review models for the prediction of occupancy and window-opening behaviors in smart buildings. Several papers concerning the employment of hidden Markov models are mentioned in the paper, but no thorough discussion is covered due to the inclusion of other machine learning techniques.

Salimi and Hammad [21] expands on the topic of occupancy modeling and includes also a review of the control systems. The latter are a subsequent of the former with occupancy modeling techniques advancing them greatly. This is due to the increased efficiency and precision of such systems as well as its highly beneficial impact in relation to energy and its consumption. This is an expansive paper and as you would imagine highly involved in regard of the application rather the machine learning technique at hand, i.e., HMMs. It is only mentioned in terms of approaches used in papers.

3.1 Limitations and Future Venues of Improvement

In this section, we briefly discuss limitations and future venues of improvements in relation to HMMs employment in occupancy estimation. This is in addition to the first limitation that we covered whereby the structure of the HMM is inherently assumed to correspond to the physical system. As discussed, as well as researched, this is not always the case.

In [90], pressure, temperature, humidity, and light levels were used for an occupancy detection system. The raw features were utilized for the investigation of multiple classical machine learning algorithms. The latter included Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), and Random Forest (RF) models with high accuracies reaching to 99%. Important conclusions were drawn from this paper and enforce a significant aspect that is to be considered in this field. That is, an appropriate selection of features and model is integral to the overall accuracy performance of the system. This is also further improved by incorporating the time information of the sensor data that is collected.

Another interesting venue that garners further investigation are simulation tools that have the capability of generating data for model training with available ground truth labels at a large scale. For instance, [87] proposes an algorithm that is capable of characterizing an occupant's presence whose resultant simulations can be then used for building models. Indeed, this proposed technique utilizes inhomogeneous HMMs for the generation of an occupancy detection time series. This system has proven efficient when tested and represents a field that would have great impact due to the limited availability of large-scale datasets, especially ones whose simulations are closely emulating the real-life scenarios.

Indeed, an interested reader is referred to [16, 91] for further discussions on various aspects of such simulation systems. Liao et al. [11] is another simulation paper with graphical models for an interested reader. The gravity of such efforts cannot be overstated due to also the lack of labelled data at a large scale. Data augmentation is also another promising prospect that addresses this challenge [86].

4 Conclusion

In conclusion, we have presented a holistic treatment of the HMM topic and an introduction to its insides. The chapter addresses the diverse aspects of the model and provides the reader with insights into its structure and mathematical formulations. We also survey its application in the field of occupancy estimation. To the best of our knowledge, this is the first review of HMMs in that field. All in all, this assembled guide provides a thorough explanation of HMMs for beginners and practitioners alike. It is our aspiration that this chapter becomes a reference for the next generation of researchers in this field.

Acknowledgment This work was funded and supported by Ericsson—Global Artificial Intelligence Accelerator in Montreal and a Mitacs Accelerate fellowship.

References

1. L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
2. A.Y. Ng, M.I. Jordan, On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes, in *Advances in Neural Information Processing Systems 14*, ed. by T.G. Dietterich, S. Becker, Z. Ghahramani (MIT Press, 2002), pp. 841–848. http://papers.nips.cc/paper/2020-on-discriminative-vs-generative-classifiers-a-comparison-of-logistic-regression-and-naive-bayes.pdf
3. N. Bouguila, Hybrid generative/discriminative approaches for proportional data modeling and classification. IEEE Trans. Knowl. Data Eng. **24**(12), 2184–2202 (2012). https://doi.org/10.1109/TKDE.2011.162
4. J.A. Lasserre, C.M. Bishop, 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, ser. CVPR '06* (IEEE Computer Society, Washington, DC, 2006), pp. 87–94. https://doi.org/10.1109/CVPR.2006.227
5. S. Ali, N. Bouguila, Hybrid generative-discriminative generalized Dirichlet-based hidden Markov models with support vector machines, in *2019 IEEE International Symposium on Multimedia (ISM)* (IEEE, Piscataway, 2019), pp. 231–2311
6. S. Ali, 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, Piscataway, 2019), pp. 1–5
7. A. Toleikyte, L. Kranzl, A. Müller, Cost curves of energy efficiency investments in buildings - methodologies and a case study of Lithuania. Energy Policy **115**, 148–157 (2018)
8. M. Evans, S. Yu, A. Staniszewski, L. Jin, A. Denysenko, The international implications of national and local coordination on building energy codes: case studies in six cities. J. Clean. Prod. **191**, 127–134 (2018)
9. J. Brooks, S. Kumar, S. Goyal, R. Subramany, P. Barooah, Energy-efficient control of under-actuated HVAC zones in commercial buildings. Energy Build. **93**, 160–168 (2015)
10. G.Y. Yun, H.J. Kong, H. Kim, J.T. Kim, A field survey of visual comfort and lighting energy consumption in open plan offices. Energy Build. **46**, 146–151 (2012). Sustainable and healthy buildings
11. C. Liao, Y. Lin, P. Barooah, Agent-based and graphical modelling of building occupancy. J. Build. Perform. Simul. **5**(1), 5–25 (2012)
12. Z. Liu, Y. Xie, K. Y. Chan, K. Ma, X. Guan, Chance-constrained optimization in D2D-based vehicular communication network. IEEE Tran. Veh. Technol. **68**(5), 5045–5058 (2019)
13. F. Oldewurtel, D. Sturzenegger, M. Morari, Importance of occupancy information for building climate control. Appl. Energy **101**, 521–532 (2013)
14. X. Pan, C.S. Han, K. Dauber, K.H. Law, A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. AI & Soc. **22**(2), 113–132 (2007)
15. Y. Yamaguchi, Y. Shimoda, M. Mizuno, Transition to a sustainable urban energy system from a long-term perspective: case study in a Japanese business district. Energy Build. **39**(1), 1–12 (2007)
16. A. Roetzel, Occupant behaviour simulation for cellular offices in early design stages—architectural and modelling considerations. Build. Simul. **8**(2), 211–224 (2015)
17. D. Trivedi, V. Badarla, Occupancy detection systems for indoor environments: a survey of approaches and methods. Indoor Built Environ. **29**(8), 1053–1069 (2020). https://doi.org/10.1177/1420326X19875621
18. H. Saha, A.R. Florita, G.P. Henze, S. Sarkar, Occupancy sensing in buildings: a review of data analytics approaches. Energy Build. **188–189**, 278–285 (2019). http://www.sciencedirect.com/science/article/pii/S0378778818333176

19. J. Ahmad, H. Larijani, R. Emmanuel, M. Mannion, A. Javed, Occupancy detection in non-residential buildings—a survey and novel privacy preserved occupancy monitoring solution. Appl. Comput. Inform. **17**(2), 279–295 (2021)
20. X. Dai, J. Liu, X. Zhang, A review of studies applying machine learning models to predict occupancy and window-opening behaviours in smart buildings. Energy Build. **223**, 110159 (2020). http://www.sciencedirect.com/science/article/pii/S0378778820303017
21. S. Salimi, A. Hammad, Critical review and research roadmap of office building energy management based on occupancy monitoring. Energy Build. **182**, 214–241 (2019). http://www.sciencedirect.com/science/article/pii/S037877881830848X
22. K. Sun, Q. Zhao, J. Zou, A review of building occupancy measurement systems. Energy Build. **216**, 109965 (2020). http://www.sciencedirect.com/science/article/pii/S0378778819332918
23. V.L. Erickson, M.A. Carreira-Perpiñán, A.E. Cerpa, Occupancy modeling and prediction for building energy management. ACM Trans. Sen. Netw. **10**(3) (2014). https://doi.org/10.1145/2594771
24. V.L. Erickson, M.Á. Carreira-Perpiñán, 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* (IEEE, Piscataway, 2011), pp. 258–269
25. B. Dong, B. Andrews, Sensor-based occupancy behavioral pattern recognition for energy and comfort management in intelligent buildings, in *Proceedings of Building Simulation* (2009), pp. 1444–1451
26. L. Rabiner, B. Juang, An introduction to hidden Markov models. IEEE ASSP Mag. **3**(1), 4–16 (1986)
27. D.J. Patterson, D. Fox, H. Kautz, 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
28. C.M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer, Berlin, 2006)
29. R.E. Kalman, Mathematical description of linear dynamical systems. J. Soc. Ind. Appl. Math. Ser. A Control **1**(2), 152–192 (1963)
30. M. Rodriguez, C. Orrite, C. Medrano, D. Makris, One-shot learning of human activity with an map adapted GMM and simplex-HMM. IEEE Trans. Cybern. **47**(7), 1769–1780 (2017)
31. M. Wang, S. Abdelfattah, N. Moustafa, J. Hu, Deep Gaussian mixture-hidden Markov model for classification of EEG signals. IEEE Trans. Emerg. Top. Comput. Intell. **2**(4), 278–287 (2018)
32. J. Bilmes, A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Int. Comput. Sci. Inst. **4**, 126 (1998)
33. S. Amudala, S. Ali, 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, Piscataway, 2020), pp. 120–127
34. Z. Song, S. Ali, N. Bouguila, W. Fan, Nonparametric hierarchical mixture models based on asymmetric Gaussian distribution. Digit. Signal Process. **106**, 102829 (2020)
35. K. Maanicshah, S. Ali, W. Fan, 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, Berlin, 2019), pp. 94–105
36. Z. Song, S. Ali, N. Bouguila, Bayesian learning of infinite asymmetric Gaussian mixture models for background subtraction, in *International Conference on Image Analysis and Recognition* (Springer, Berlin, 2019), pp. 264–274
37. E. Epaillard, N. Bouguila, Variational Bayesian learning of generalized Dirichlet-based hidden Markov models applied to unusual events detection. IEEE Trans. Neural Netw. Learn. Syst. **30**(4), 1034–1047 (2019)

38. S. Ali, N. Bouguila, Variational learning of beta-Liouville hidden Markov models for infrared action recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2019)
39. R. Nasfi, M. Amayri, N. Bouguila, A novel approach for modeling positive vectors with inverted Dirichlet-based hidden Markov models. Knowl. Based Syst. **192**, 105335 (2020). http://www.sciencedirect.com/science/article/pii/S0950705119306057
40. W. Fan, L. Yang, N. Bouguila, Y. Chen, Sequentially spherical data modeling with hidden Markov models and its application to FMRI data analysis. Knowl. Based Syst. **206**, 106341 (2020). http://www.sciencedirect.com/science/article/pii/S0950705120305001
41. E. Epaillard, N. Bouguila, Hybrid hidden Markov model for mixed continuous/continuous and discrete/continuous data modeling, in *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP)* (2015), pp. 1–6
42. S.K. Ng, T. Krishnan, G.J. McLachlan, *The EM Algorithm* (Springer, Berlin, 2012), pp. 139–172
43. R.A. Redner, H.F. Walker, Mixture densities, maximum likelihood and the EM algorithm. SIAM Rev. **26**(2), 195–239 (1984)
44. A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Series B Methodol. **39**(1), 1–22 (1977). https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x
45. R.D. Bock, M. Aitkin, Marginal maximum likelihood estimation of item parameters: application of an EM algorithm. Psychometrika **46**(4), 443–459 (1981). https://doi.org/10.1007/BF02293801
46. B.S. Everitt, Maximum likelihood estimation of the parameters in a mixture of two univariate normal distributions; a comparison of different algorithms. J. R. Stat. Soc. Series D Stat. **33**(2), 205–215 (1984). https://rss.onlinelibrary.wiley.com/doi/abs/10.2307/2987851
47. S.E. Levinson, L.R. Rabiner, M.M. Sondhi, An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. Bell Syst. Tech. J. **62**(4), 1035–1074 (1983). https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1983.tb03114.x
48. S. Levinson, L. Rabiner, M. Sondhi, Speaker independent isolated digit recognition using hidden Markov models, in *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 8 (1983), pp. 1049–1052
49. J. Li, J.Y. Lee, 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
50. J.A. Hartigan, *Clustering Algorithms*, 99th edn. (Wiley, New York, 1975)
51. B.-J. Yoon, Hidden Markov models and their applications in biological sequence analysis. Curr. Genomics **10**(6), 402–415 (2009)
52. D. Khiatani, 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
53. E. Epaillard, 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 Recognit. **55**, 125–136 (2016). http://www.sciencedirect.com/science/article/pii/S0031320316000601
54. E. Epaillard, N. Bouguila, Hidden Markov models based on generalized Dirichlet mixtures for proportional data modeling, in *Artificial Neural Networks in Pattern Recognition*, ed. by N. El Gayar, F. Schwenker, C. Suen (Springer International Publishing, Cham, 2014), pp. 71–82
55. X. Wang, H. Wu, 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

56. M. Haid, B. Budaker, M. Geiger, D. Husfeldt, M. Hartmann, 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
57. S. Wolf, J.K. Mazller, M.A. Bitsch, J. Krogstie, H. Madsen, A Markov-switching model for building occupant activity estimation. Energy Build. **183**, 672–683 (2019). http://www.sciencedirect.com/science/article/pii/S0378778818320887
58. P. Kumar, 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
59. M.A. Fouad, 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
60. S. Espana-Boquera, M.J. Castro-Bleda, J. Gorbe-Moya, F. Zamora-Martinez, Improving offline handwritten text recognition with hybrid HMM/ANN models. IEEE Trans. Pattern Anal. Mach. Intell. **33**(4), 767–779 (2010)
61. A. Schlapbach, H. Bunke, A writer identification and verification system using HMM based recognizers. Pattern Anal. Appl. **10**(1), 33–43 (2007)
62. A. Schlapbach, H. Bunke, Using HMM based recognizers for writer identification and verification, in *Ninth International Workshop on Frontiers in Handwriting recognition* (IEEE, Piscataway, 2004), pp. 167–172
63. K. Tokuda, H. Zen, A.W. Black, An HMM-based speech synthesis system applied to English, in *IEEE Speech Synthesis Workshop* (2002), pp. 227–230
64. J. Jayakumari, 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
65. B. Mor, S. Garhwal, A. Kumar, A systematic review of hidden Markov models and their applications. Arch. Comput. Methods Eng. (2020). https://doi.org/10.1007/s11831-020-09422-4
66. S. Ali, N. Bouguila, Online learning for beta-Liouville hidden Markov models: incremental variational learning for video surveillance and action recognition, in *2020 IEEE International Conference on Image Processing (ICIP)* (IEEE, Piscataway, 2020), pp. 3249–3253
67. S. Ali, 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)* (IEEE, Piscataway, 2020), pp. 1–6
68. S. Ali, 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)* (IEEE, Piscataway, 2020), pp. 4081–4087
69. Y. Yuan, X. Li, Z. Liu, X. Guan, Occupancy estimation in buildings based on infrared array sensors detection. IEEE Sens. J. **20**(2), 1043–1053 (2020)
70. Z. Li, B. Dong, A new modeling approach for short-term prediction of occupancy in residential buildings. Build. Environ. **121**, 277–290 (2017)
71. Y. Yuan, X. Li, Z. Liu, X. Guan, Occupancy estimation in buildings based on infrared array sensors detection. IEEE Sens. J. **20**(2), 1043–1053 (2020)
72. S. Ali, N. Bouguila, Towards scalable deployment of hidden Markov models in occupancy estimation: a novel methodology applied to the study case of occupancy detection. Energy Build. **254**, 111594 (2022). https://www.sciencedirect.com/science/article/pii/S0378778821008781
73. M. Yoshida, S. Kleisarchaki, L. Gtirgen, H. Nishi, Indoor occupancy estimation via location-aware HMM: an IOT approach, in *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)* (2018), pp. 14–19
74. S.H. Ryu, H.J. Moon, Development of an occupancy prediction model using indoor environmental data based on machine learning techniques. Build. Environ. **107**, 1–9 (2016). http://www.sciencedirect.com/science/article/pii/S0360132316302463

75. B. Dong, B. Andrews, K.P. Lam, M. Hauynck, R. Zhang, Y.-S. Chiou, D. Benitez, An information technology enabled sustainability test-bed (ITEST) for occupancy detection through an environmental sensing network. Energy Build. **42**(7), 1038–1046 (2010). http://www.sciencedirect.com/science/article/pii/S037877881000023X
76. K.P. Lam, M. Hauynck, B. Dong, B. Andrews, Y. Shang Chiou, D. Benitez, J. Choi, Occupancy detection through an extensive environmental sensor network in an open-plan office building, in *Proc. of Building Simulation 09, an IBPSA Conference* (2009)
77. B. Dong, K.P. Lam, A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting. Build. Simul. **7**(1), 89–106 (2014). https://doi.org/10.1007/s12273-013-0142-7
78. B. Dong, K.P. Lam, C. Neuman, Integrated building control based on occupant behavior pattern detection and local weather forecasting, in *Twelfth International IBPSA Conference. Sydney: IBPSA Australia*. Citeseer (2011), pp. 14–17
79. J.L. Gomez Ortega, L. Han, N. Bowring, A novel dynamic hidden semi-Markov model (D-HSMM) for occupancy pattern detection from sensor data stream, in *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)* (2016), pp. 1–5
80. C. Papatsimpa, J.M.G. Linnartz, Improved presence detection for occupancy control in multi-sensory environments, in *2017 IEEE International Conference on Computer and Information Technology (CIT)* (2017), pp. 75–80
81. C. Papatsimpa, J.P.M.G. Linnartz, Using dynamic occupancy patterns for improved presence detection in intelligent buildings, in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)* (2018), pp. 1–5
82. J. Chaney, E. Hugh Owens, A.D. Peacock, An evidence based approach to determining residential occupancy and its role in demand response management. Energy Build. **125**, 254–266 (2016). http://www.sciencedirect.com/science/article/pii/S0378778816303255
83. L. Song, X. Niu, Q. Lyu, S. Lyu, T. Tian, A time-aware method for occupancy detection in a building, in *12th EAI International Conference on Mobile Multimedia Communications, Mobimedia 2019* (2019)
84. B. Dong, K.P. Lam, Building energy and comfort management through occupant behaviour pattern detection based on a large-scale environmental sensor network. J. Build. Perform. Simul. **4**(4), 359–369 (2011). https://doi.org/10.1080/19401493.2011.577810
85. B. Dong, B. Andrews, Sensor-based occupancy behavioral pattern recognition for energy and comfort management in intelligent buildings, in *Proceedings of Building Simulation* (2009), pp. 1444–1451
86. D. Chen, Y. Tang, H. Zhang, L. Wang, X. Li, Incremental factorization of big time series data with blind factor approximation. IEEE Trans. Knowl. Data Eng. **33**, 1–1 (2019)
87. J. Page, D. Robinson, N. Morel, J.-L. Scartezzini, A generalised stochastic model for the simulation of occupant presence. Energy Build. **40**(2), 83–98 (2008)
88. W. Shen, G. Newsham, B. Gunay, Leveraging existing occupancy-related data for optimal control of commercial office buildings: a review. Adv. Eng. Inform. **33**, 230–242 (2017). http://www.sciencedirect.com/science/article/pii/S1474034616301987
89. K. Akkaya, I. Guvenc, R. Aygun, N. Pala, A. Kadri, Iot-based occupancy monitoring techniques for energy-efficient smart buildings, in *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)* (2015), pp. 58–63
90. L.M. Candanedo, V. Feldheim, Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models. Energy Build. **112**, 28–39 (2016)
91. B. Dong, D. Yan, Z. Li, Y. Jin, X. Feng, H. Fontenot, Modeling occupancy and behavior for better building design and operation—a critical review. Build. Simul. **11**(5), 899–921 (2018). https://doi.org/10.1007/s12273-018-0452-x

Bounded Asymmetric Gaussian Mixture-Based Hidden Markov Models

Zixiang Xian, Muhammad Azam, Manar Amayri, Wentao Fan, and Nizar Bouguila

1 Introduction

Hidden Markov models were introduced by Baum and his colleagues by estimating its parameters using the maximum likelihood (ML) approach [1–4]. HMM has long been referred to as a dynamic probabilistic model with discrete transition probability that has been implemented in various applications such as speech processing [5–7], anomaly detection [8], signature verification [9–11], as well as pattern recognition applications like gesture and texture recognition [12–14]. Furthermore, it has been also used in the smart buildings domain for occupancy estimation [15–17]. The primary goal for using HMMs was to characterize real-world signals in terms of signal models, which can help us improve signals by reducing noise and transmission distortion, as well as to learn details about the signal source without having to have the source available via simulations [7].

HMMs have been proved to be very practical while dealing with non-observable data over a time interval to disclose the future values or reveal the latent variables. Although some research works tend to improve the HMM structure by tuning the initialization step in the context of parameter setting [18, 19], the training

Z. Xian · M. Azam · N. Bouguila (✉)
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: zi_xian@encs.concordia.ca; mu_azam@encs.concordia; nizar.bouguila@concordia.ca

M. Amayri
G-SCOP Laboratory, Grenoble Institute of Technology, Grenoble, France
e-mail: manar.amayri@grenoble-inp.fr

W. Fan
Department of Computer Science and Technology, Huaqiao University, Xiamen, China
e-mail: fwt@hqu.edu.cn

N. Bouguila et al. (eds.), *Hidden Markov Models and Applications*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-99142-5_2

process of HMM considers the identical regulated classic form via the Expectation–Maximization algorithm [20]. However, in most cases, the choice of emission probability distributions is less discussed and generally adopts Gaussian mixture models (GMMs) by default, often because of mathematical and practical convenience and strong assumption of a common pattern for the data [21]. However, this strong assumption is potentially insufficient to achieve the best modeling performance because real-world data cannot be symmetric in all cases, not to mention the unbounded support that prevents it from having a reliable modeling capability in the presence of outliers. Note that most real-world data are asymmetric, which is especially true in natural images, as shown in [22]. Therefore, some research works have put forward the generalized Gaussian mixture model (GGMM) [23–28], which can consider different shapes by changing its shape parameters that control the distribution's tail. Some research works have focused on improving the distribution support for real-world data, which is always defined with a bounded range, proposing the bounded Gaussian mixture model (BGMM) [29–31] and the bounded generalized Gaussian mixture model (BGGMM) [32–34]. A bounded asymmetric Gaussian mixture model (BAGMM) has been proposed in [35] to tackle the drawbacks of assuming symmetric unbounded data in real-life applications.

In the present chapter, we propose to explore and evaluate the performance of HMM by adopting BAGMM as emission probability distribution and comparing it with the Gaussian mixture model-based HMM (GMM-HMM) and other general Gaussian-based versions. Although we break the strong assumptions made by Gaussian mixture models from emission probability distributions, the HMM still has two main limitations [36]. A significant limitation is an assumption that successive observations are independent. Another limitation is the Markov assumption itself, i.e., that the probability of being in a given state at time t only depends on the previous state at time $t - 1$. Therefore, we aim to show that the combination of BAGMM and HMM can acquire better performance when handling real-world data compared with traditional HMM and other Gaussian mixture-derived HMMs. We will reveal the details about the parameters learning process of the proposed model, including the parameters setting, i.e., the number of hidden states and mixture components, and the performance. Indeed, the parameters setting has its share of effect on the modeling accuracy. To tackle the parameters estimation task, we introduce Expectation–Maximization (EM) framework [20] to maximize log-likelihood. To emphasize our significant contributions to this research work briefly, we first introduce a complete derivation of the equations for integrating the bounded asymmetric Gaussian mixture into the HMM framework and apply this novel HMM framework to real-world applications while comparing it with traditional HMM and Gaussian mixture-derived HMMs.

The remainder of this chapter is organized as follows: After the introduction, we recall the bounded asymmetric Gaussian mixture model (BAGMM) in detail, including its probability density function (PDF) in Sect. 2. Section 3 briefly recalls the structure and definition of traditional HMM. In Sect. 4, we specify the complete

procedure about how to frame the BAGMM into HMM, including the parameters learning algorithm. In Sect. 5, we present the applications and the results of our proposed model compared with other selected models. The conclusion and potential future works are presented in Sect. 6.

2 Bounded Asymmetric Gaussian Mixture Model

Given a D-dimensional random variable $\mathbf{X} = [X_1, \ldots, X_D]$ that follows a M-component mixture distribution, its probability density function (PDF) can be written as:

$$p(\mathbf{X}|\Lambda) = \sum_{m=1}^{M} p(\mathbf{x}|\xi_m)\, p_m \tag{1}$$

where p_m are the mixing coefficients that satisfy $p_j \geq 0$, $\sum_{m=1}^{M} p_m = 1$, ξ_m is the parameter of the distribution associated with mth cluster and $\Lambda = (\xi_1, \ldots, \xi_M, p_1, \ldots, p_M)$ is the complete set of parameters of the asymmetric Gaussian mixture model (AGMM).

The PDF associated with each component is the multidimensional asymmetric Gaussian distribution (AGD) [37–43]:

$$f(\mathbf{X}|\xi_m) = \prod_{d=1}^{D} \frac{2}{\sqrt{2\pi}(\sigma_{l_{md}} + \sigma_{rmd})} \times \begin{cases} \exp\left[-\frac{(\mathrm{X}_d - \mu_{md})^2}{2\sigma_{l_{md}}^2}\right] & \mathrm{X}_d < \mu_{md} \\ \exp\left[-\frac{(\mathrm{X}_d - \mu_{md})^2}{2\sigma_{r_{md}}^2}\right] & \mathrm{X}_d \geq \mu_{md} \end{cases} \tag{2}$$

where $\xi_m = (\boldsymbol{\mu}_m, \boldsymbol{\sigma}_{\mathbf{l_m}}, \boldsymbol{\sigma}_{\mathbf{r_m}})$ represents the parameters of AGD. Here, $\boldsymbol{\mu}_m = (\mu_{m1}, \ldots, \mu_{mD})$, $\boldsymbol{\sigma}_{\mathbf{l_m}} = (\sigma_{l_{m1}}, \ldots, \sigma_{l_{mD}})$, and $\boldsymbol{\sigma}_{\mathbf{r_m}} = (\sigma_{r_{m1}}, \ldots, \sigma_{r_{mD}})$ are the mean, left standard deviation, and right standard deviation of the D-dimensional AGD, respectively. The bounded asymmetric Gaussian distribution (BAGD) for the vector $\mathbf{X}$ can be written as:

$$p(\mathbf{X}|\xi_m) = \frac{f(\mathbf{X}|\xi_m)\mathrm{H}(\mathbf{X}|\Omega_m)}{\int_{\partial_m} f(\mathrm{u}|\xi_m)d\mathrm{u}}, \text{ where } \mathrm{H}(\mathbf{X}|\Omega_m) = \begin{cases} 1 & \text{if } \mathbf{X} \in \partial_m \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $f(\mathbf{X}|\xi_m)$ is the PDF of the AGD, the term $\int_{\partial_m} f(\mathrm{u}|\xi_m)d\mathrm{u}$ in Eq. (3) is the normalized constant that shows the share of $f(\mathbf{X}|\xi_m)$, which belongs to the support region ∂. Given a set of independent and identically distributed vectors represented

by $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$, arising from a mixture of BAGDs with M components, then its log-likelihood function can be defined as follows:

$$p(\mathbf{X}|\Lambda) = \prod_{n=1}^{N} \sum_{m=1}^{M} p(\mathbf{x}_n|\xi_m) p_m \tag{4}$$

We introduce stochastic indicator vectors $\mathbf{Z}_n = (Z_{n1}, \ldots, Z_{nM})$, which satisfy $Z_{nm} \in \{0, 1\}$, $\sum_{m=1}^{M} Z_{nm} = 1$. In other words, Z_{nm}, the hidden variable in each indicator vector equals 1 if $\mathbf{x}_n$ belongs to component j and 0, otherwise. The complete data likelihood is given by:

$$p(\mathbf{X}, \mathbf{Z}|\Lambda) = \prod_{n=1}^{N} \prod_{m=1}^{M} (p(\mathbf{x}_n|\xi_m) p_m)^{Z_{nm}} \tag{5}$$

where Z_{nm} is the posterior probability and can be written as:

$$Z_{nm} = p(m|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\xi_m) p_m}{\sum_{m=1}^{M} p(\mathbf{x}_n|\xi_m) p_m} \quad \text{and} \quad \mathbf{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_N\}. \tag{6}$$

3 Hidden Markov Model

For many real-world applications, such as occupancy estimation in buildings, we wish to predict the following number of people in a time series given sequences of the previous values. It is impractical to consider a general dependence of future observations on all previous values. Therefore, the HMM assumes that the future predictions are dependent on the most recent observations only. Moreover, the HMM is a specific instance of the state space model that the latent variables are discrete. The latent variable, which is the state of this hidden process, satisfies the Markov property; that is, given the value of s_{n-1}; the current state s_n is independent of all the states prior to the time $n - 1$. $X = [x_1, \ldots, x_N]$ represents the observed variables and $S = [s_1, \ldots, s_n]$ is the hidden state. A hidden Markov model is governed by a set of parameters, such as the set of state transitions and emission probability. There are three main tasks for HMM-based modeling; first is to optimize those parameters for the model given training data; second is scoring that calculates the joint probability of a sequence given the model; third is decoding that finds the optimal series of hidden states (Fig. 1).

According to [44], given time series observations $X = [x_1, \ldots, x_n, \ldots, x_N]$ generated by hidden states $S = [s_1, \ldots, s_n, \ldots, s_N]$; $s_k \in [1, K]$ where K is the number of the hidden states, we define the transition probability matrix as A: $A_{jk} = p(s_{nk} = 1 \mid s_{n-1,k} = 1)$. They should satisfy $0 \leq A_{jk} \leq 1$ with

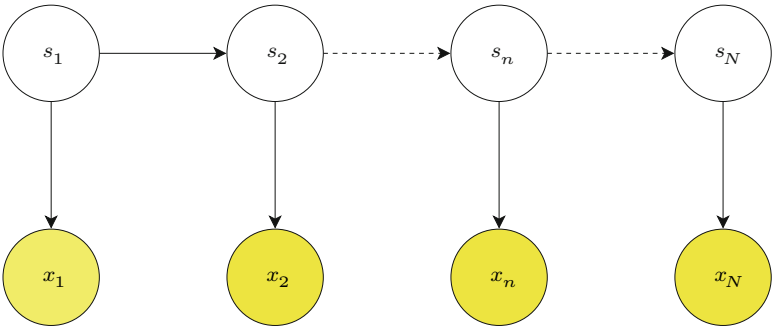


Fig. 1 Graphical representation for HMM

$\sum_k A_{jk} = 1$, because they are probabilities. $P(x_m \mid \Lambda)$ is known as emission probability, where Λ is a set of parameters governing the distribution if x is continuous. Note that $P(x_m \mid \Lambda)$ will be an emission probability matrix if x is discrete. The joint probability distribution over both hidden states and observed variables is then given by:

$$p(\mathbf{X}, \mathbf{S} \mid \boldsymbol{\Theta}) = p\left(\mathbf{s}_1 \mid \boldsymbol{\pi}\right) \left[\prod_{n=2}^{N} p\left(\mathbf{s}_n \mid \mathbf{s}_{n-1}, \mathbf{A}\right)\right] \prod_{m=1}^{N} p\left(\mathbf{x}_m \mid \Lambda\right) \tag{7}$$

where $\mathbf{X} = [x_1, \ldots, x_N]$, $\mathbf{S} = [s_1, \ldots, s_N]$, and $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \mathbf{A}, \Lambda\}$ defines the set of parameters of HMM. Indeed, there are a wide range of choices for emission distribution that include Gaussian distribution and mixture models such as Gaussian mixture model (GMM). It is worth mentioning that the emission distributions are often taken as Gaussian mixtures for most continuous observations cases [44–47].

The parameters learning task is crucial for HMM. In this chapter, we focus on the maximum log-likelihood approach via EM algorithm, which can also be considered as a selection process among all models in such a way to determine which model best matches the observations. It is intractable to directly maximize the log-likelihood function, leading to complex expressions with no closed-form solutions.

The EM framework starts with some initial parameters. Then, we need to accumulate sufficient statistics and find the posterior distribution of the state $p\left(\mathbf{S} \mid \mathbf{X}, \boldsymbol{\Theta}^{\text{old}}\right)$ by applying forward–backward algorithm in E step. We utilize this posterior distribution to update parameters Θ via maximizing the complete data likelihood with respect to each parameter in M step. The function $Q\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}}\right)$ can be defined as:

$$Q\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}}\right) = \sum_{\mathbf{S}} p\left(\mathbf{S} \mid \mathbf{X}, \boldsymbol{\Theta}^{\text{old}}\right) \ln p(\mathbf{X}, \mathbf{S} \mid \boldsymbol{\Theta}) \tag{8}$$

We introduce $\gamma\left(s_{nk}\right)$ to denote the marginal posterior distribution of the nth state s_{nk} and $\xi\left(s_{n-1,j}, s_{nk}\right)$ to define the joint posterior distribution of two successive states $s_{n-1,j}, s_{nk}$ that x_{n-1}, x_n are emitted from the jth and kth model state, respectively.

$$\begin{aligned} \gamma\left(s_{nk}\right) &= P(s_{nk} \mid \mathbf{X}, \boldsymbol{\Theta}) \\ \xi\left(s_{n-1,j}, s_{nk}\right) &= P(s_{n-1,j}, s_{nk} \mid \mathbf{X}, \boldsymbol{\Theta}) \end{aligned} \tag{9}$$

where $\gamma\left(s_{nk}\right)$ denotes the conditional probability $p(s_{nk} \mid \mathbf{X}, \boldsymbol{\theta})$, where $s_{nk} = 1$ if x_n is emitted from the kth model state, and $s_{nk} = 0$, otherwise.

We can make use of the definition of γ and ξ and substitute Eq. (8) with Eq. (9). We obtain $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right)$ as:

$$\begin{aligned} Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right) = &\sum_{k=1}^{K} \gamma\left(s_{1k}\right) \ln \pi_k + \sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi\left(s_{n-1,j}, s_{nk}\right) \ln A_{jk} \\ &+ \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma\left(s_{nk}\right) \ln p\left(\mathbf{x}_n \mid \Lambda_{nk}\right) \end{aligned} \tag{10}$$

4 BAGMM Integration into the HMM Framework

From the previous section, the emission distribution $p\left(\mathbf{x}_n \mid \Lambda_{nk}\right)$ is often taken as Gaussian mixture model (GMM) for most continuous observations cases. However, the Gaussian distribution assumes that the data is symmetric and has an infinite range, which prevents it from having a good modeling capability in the presence of outliers. So, we suggest integrating the bounded asymmetric Gaussian mixture model (BAGMM) into the HMM framework. The primary motivation behind this choice is the bounded range support from BAGMMM and its asymmetric nature for modeling non-symmetric real-world data. The BAGMM is flexible and has good capabilities to model both symmetric and asymmetric data.

By replacing the emission probability distribution as BAGMM, we can integrate BAGMM into the HMM framework, which is to substitute $p\left(\mathbf{x}_n \mid \Lambda_{nk}\right)$ with Eq. (3) in Eq. (10). In the E step, we obtain $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right)$ using Eq. (10). In the M step, we maximize $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right)$ with respect to the parameters $\Theta = \{\boldsymbol{\pi}, \mathbf{A}, \Lambda\}$ in which we treat γ, ξ as a constant. The details are discussed in the next subsection.

4.1 Estimation of π and A

Using Lagrange multipliers, the maximization concerning π_k and A_{jk} gives the following:

$$\pi_k = \frac{\gamma\left(s_{1k}\right)}{\sum_{j=1}^{K}\gamma\left(s_{1j}\right)} \tag{11}$$

$$A_{jk} = \frac{\sum_{n=2}^{N}\xi\left(s_{n-1,j}, s_{nk}\right)}{\sum_{l=1}^{K}\sum_{n=2}^{N}\xi\left(s_{n-1,j}, s_{nl}\right)} \tag{12}$$

Note that the initialization for π_k and A_{jk} should respect the summation constraints, $\sum_{k=1}^{K}\pi_k = 1$ and $\sum_{k=1}^{K}A_{jk} = 1$.

4.2 Estimation of Λ

To maximize $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right)$ with respect to Λ_k, we note that the final term in Eq. (10) depends on Λ_k. The Λ_k is a set of parameters of the kth state emission probability distribution, $\Lambda_k = [p_1, \ldots, p_m, \mu_1, \ldots, \mu_m, \sigma_{l1}, \ldots, \sigma_{lm}, \sigma_{r1}, \ldots, \sigma_{rm}]$. Here, we denote by $\varphi_n(k, m)$ the probability of being at state s_k at time n with respect to the mth bounded asymmetric Gaussian mixture. According to [36, 48], the $\varphi_n(k, m)$ can be computed as:

$$\varphi_n(k, m) = \frac{\alpha\left(s_{nk}\right)\beta\left(s_{nk}\right)}{\sum_{k=1}^{K}\alpha\left(s_{nk}\right)\beta\left(s_{nk}\right)} \cdot \frac{p(\mathbf{x}_n|\xi_{km})p_{km}}{\sum_{m=1}^{M}p(\mathbf{x}_n|\xi_{km})p_{km}} \tag{13}$$

where $\alpha\left(\mathbf{s}_n\right)$ denotes the joint probability of observing all of the given data up to time n and the hidden state s_n, whereas $\beta\left(\mathbf{s}_n\right)$ represents the conditional probability of all future data from time $n+1$ up to N given the hidden state of s_n:

$$\alpha\left(\mathbf{s}_n\right) \equiv p\left(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{s}_n\right) \tag{14}$$

$$\beta\left(\mathbf{s}_n\right) \equiv p\left(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_N \mid \mathbf{s}_n\right) \tag{15}$$

The mixing coefficient p_{km}^{new} of the mth bounded asymmetric Gaussian mixture in the state k is given by:

$$p_{km}^{\text{new}} = \frac{\sum_{n=1}^{N}\varphi_n(k, m)}{\sum_{n=1}^{N}\sum_{m=1}^{M}\varphi_n(k, m)} \tag{16}$$

The mean $\boldsymbol{\mu}_{kmd}^{\text{new}}$ can be defined using the same approach.

$$\boldsymbol{\mu}_{kmd}^{\text{new}} = \frac{\sum_{n=1}^{N} \varphi_n(k,m) \left\{ \mathbf{x}_{nd} - \frac{\int_{\partial_{km}} f(\mathbf{u}|\xi_{km})(\mathbf{u}-\mu_{kmd})d\mathrm{u}}{\int_{\partial_{km}} f(\mathbf{u}|\xi_{km})d\mathbf{u}} \right\}}{\sum_{n=1}^{N} \varphi_n(k,m)} \tag{17}$$

Note that in Eq. (17), the term $\int_{\partial_{km}} f(\mathrm{u}|\xi_{km})(\mathrm{u} - \mu_{kmd})du$ is the expectation of function $(\mathrm{u} - \mu_{kmd})$ under the probability distribution $f(\mathbf{x}_d|\xi_{km})$. Then, this expectation can be approximated as:

$$\int_{\partial_{km}} f(\mathrm{u}|\xi_{km})(\mathrm{u} - \mu_{kmd})d\mathrm{u} \approx \frac{1}{M} \sum_{m=1}^{M} (m_{kmd} - \mu_{kmd})\mathrm{H}(m_{kmd}|\Omega_{km}) \tag{18}$$

where $m_{kmd} \sim f(\mathrm{u}|\xi_{km})$ is a set of random variables drawn from the asymmetric Gaussian distribution for the particular component m of the mixture model at the state k. The term $\int_{\partial_{km}} f(\mathrm{u}|\xi_{km})d\mathrm{u}$ in Eq. (17) can be approximated as:

$$\int_{\partial_{km}} f(\mathrm{u}|\xi_{km})d\mathrm{u} \approx \frac{1}{M} \sum_{m=1}^{M} \mathrm{H}(m_{kmd}|\Omega_{km}) \tag{19}$$

and

$$\boldsymbol{\mu}_{kmd}^{\text{new}} = \frac{\sum_{n=1}^{N} \varphi_n(k,m) \left\{ \mathbf{x}_{nd} - \frac{\sum_{m=1}^{M} (m_{kmd}-\mu_{kmd})\mathrm{H}(m_{kmd}|\Omega_{km})}{\sum_{m=1}^{M} \mathrm{H}(m_{kmd}|\Omega_{km})} \right\}}{\sum_{n=1}^{N} \varphi_n(k,m)} \tag{20}$$

The left standard deviation can be estimated by maximizing the log-likelihood function with respect to $\sigma_{l_{kmd}}$, which can be performed using Newton–Raphson method:

$$\sigma_{l_{kmd}}^{\text{new}} = \sigma_{l_{kmd}}^{\text{old}} - \left[\left(\frac{\partial^2 Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right)}{\partial \sigma_{l_{kmd}}^2} \right)^{-1} \left(\frac{\partial Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right)}{\partial \sigma_{l_{kmd}}} \right) \right] \tag{21}$$

where the first derivative of the model's complete data log-likelihood with respect to left standard deviation $\sigma_{l_{kmd}}$ is given as follows:

$$\frac{\partial Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right)}{\partial \sigma_{l_{kmd}}} = 0 \tag{22}$$

$$\frac{\partial Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right)}{\partial \sigma_{l_{kmd}}} = \frac{\partial}{\partial \sigma_{l_{kmd}}} \sum_{n=1}^{N} \varphi_n(k,m) \times$$

$$\left\{ \log p_{km} + \log f(\mathbf{x}_n|\xi_{km}) + \log \mathrm{H}(\mathbf{x}_n|km) - \log \int_{\partial_{km}} f(\mathrm{u}|\xi_{km}) d\mathrm{u} \right\}$$

$$= \frac{\partial}{\partial \sigma_{l_{kmd}}} \sum_{n=1}^{N} \varphi_n(k,m) \left\{ \log f(\mathbf{x}_n|\xi_{km}) - \log \int_{\partial_{km}} f(\mathrm{u}|\xi_{km}) d\mathrm{u} \right\}$$

$$= \sum_{n=1, \mathbf{x}_{nd} < \mu_{kmd}}^{N} \varphi_n(k,m) \left(\frac{(\mathbf{x}_{nd} - \mu_{kmd})^2}{\sigma_{l_{kmd}}^3} \right)$$

$$- \sum_{i=1, \mathbf{x}_{nd} < \mu_{jd}}^{N} \frac{\varphi_n(k,m)}{\sigma_{l_{kmd}}^3} \left\{ \frac{\int_{\partial_{km}} \mathrm{g}_1(\mathrm{u} \mid \xi_{km}) (\mathrm{u} - \mu_{kmd})^2 d\mathrm{u}}{\int_{\partial_{km}} \mathrm{g}_1(\mathrm{u} \mid \xi_{km}) d\mathrm{u}} \right\} \tag{23}$$

The term $\int_{\partial_{km}} \mathrm{g}_1(\mathrm{u}|\xi_{km})(\mathrm{u} - \mu_{kmd})^2 d\mathrm{u}$ can be approximated as below:

$$\int_{\partial_{km}} \mathrm{g}_1(\mathrm{u}|\xi_{km})(\mathrm{u} - \mu_{kmd})^2 du \approx \frac{1}{M} \sum_{m=1}^{M} (\mathrm{l}_{kmd} - \mu_{kmd})^2 \mathrm{H}(\mathrm{l}_{kmd}|\Omega_{km}) \tag{24}$$

where $\mathrm{l}_{kmd} \sim \mathrm{g}_1(\mathbf{x}_n|\xi_{km})$ is a set of random variables drawn from the asymmetric Gaussian distribution with $\mathrm{u} < \mu_{kmd}$ for the particular component m of the mixture model at the state k. Similarly, the term $\int_{\partial_{km}} \mathrm{g}_1(\mathrm{u}|\xi_{km}) d\mathrm{u}$ in Eq. (23) can be approximated as:

$$\int_{\partial_{km}} \mathrm{g}_1(\mathrm{u}|\xi_{km}) d\mathrm{u} \approx \frac{1}{M} \sum_{m=1}^{M} \mathrm{H}(\mathrm{l}_{kmd}|\Omega_{km}) \tag{25}$$

The same approximation for the second-order derivative of the model's complete data log-likelihood with respect to left standard deviation is defined as follows:

$$\frac{\partial^2 Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right)}{\partial \sigma_{l_{kmd}}^2} = -3 \sum_{n=1, \mathbf{x}_{nd} < \mu_{kmd}}^{N} \varphi_n(k,m) \left(\frac{(\mathbf{x}_{nd} - \mu_{kmd})^2}{\sigma_{l_{kmd}}^4} \right)$$

$$- \sum_{n=1, \mathbf{x}_{nd} < \mu_{jd}}^{N} \varphi_n(k,m) \left(\frac{-2}{\sigma_{l_{kmd}}^3 \left(\sigma_{l_{kmd}} + \sigma_{r_{kmd}}\right)} \right) \times$$

$$\left\{ \frac{\frac{1}{M} \sum_{m=1}^{M} (\mathrm{l}_{kmd} - \mu_{kmd})^2 \mathrm{H}(\mathrm{l}_{kmd} \mid \Omega_{km})}{\frac{1}{M} \sum_{m=1}^{M} \mathrm{H}(\mathrm{l}_{kmd} \mid \Omega_{km})} \right\}$$

$$
\begin{aligned}
&- \sum_{n=1,\mathbf{x}_{nd}<\mu_{kmd}}^{N} \frac{\varphi_n(k,m)}{\sigma_{l_{kmd}}^6} \left\{ \frac{\frac{1}{M}\sum_{m=1}^{M} (\mathrm{l}_{kmd}-\mu_{kmd})^4 \,\mathrm{H}\,(\mathrm{l}_{kmd} \mid \Omega_{km})}{\frac{1}{M}\sum_{m=1}^{M} \mathrm{H}\,(\mathrm{l}_{kmd} \mid \Omega_{km})} \right\} \\
&- \sum_{n=1,\mathbf{x}_{nd}<\mu_{kmd}}^{N} \frac{-3\varphi_n(k,m)}{\sigma_{l_{kmd}}^4} \left\{ \frac{\frac{1}{M}\sum_{m=1}^{M} (\mathrm{l}_{kmd}-\mu_{kmd})^2 \,\mathrm{H}\,(\mathrm{l}_{kmd} \mid \Omega_{km})}{\frac{1}{M}\sum_{m=1}^{M} \mathrm{H}\,(\mathrm{l}_{kmd} \mid \Omega_{km})} \right\} \\
&- \sum_{n=1,\mathbf{x}_{nd}<\mu_{kmd}}^{N} \frac{\varphi_n(k,m)}{\sigma_{l_{kmd}}^6} \left\{ \frac{\left(\frac{1}{M}\sum_{m=1}^{M} (\mathrm{l}_{kmd}-\mu_{kmd})^2 \,\mathrm{H}\,(\mathrm{l}_{kmd} \mid \Omega_{km})\right)^2}{\left(\frac{1}{M}\sum_{m=1}^{M} \mathrm{H}\,(\mathrm{l}_{kmd} \mid \Omega_{km})\right)^2} \right\}
\end{aligned}
\tag{26}
$$

The right standard deviation can be estimated by maximizing the log-likelihood function with respect to $\sigma_{r_{kmd}}$, which can be performed using Newton–Raphson method:

$$
\sigma_{r_{kmd}}^{\text{new}} = \sigma_{r_{kmd}}^{\text{old}} - \left[\left(\frac{\partial^2 Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right)}{\partial \sigma_{r_{kmd}}^2} \right)^{-1} \left(\frac{\partial Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right)}{\partial \sigma_{r_{kmd}}} \right) \right] \tag{27}
$$

Similar approximations are used for $\frac{\partial Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right)}{\partial \sigma_{r_{kmd}}}$ as follows:

$$
\begin{aligned}
\frac{\partial Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right)}{\partial \sigma_{r_{kmd}}} &= \frac{\partial}{\partial \sigma_{r_{kmd}}} \sum_{n=1}^{N} \varphi_n(k,m) \times \\
&\quad \left\{ \log p_{km} + \log f(\mathbf{x}_n|\xi_{km}) + \log \mathrm{H}(\mathbf{x}_n|km) - \log \int_{\partial_{km}} f(\mathrm{u}|\xi_{km}) d\mathrm{u} \right\} \\
&= \frac{\partial}{\partial \sigma_{r_{kmd}}} \sum_{n=1}^{N} \varphi_n(k,m) \left\{ \log f(\mathbf{x}_n|\xi_{km}) - \log \int_{\partial_{km}} f(\mathrm{u}|\xi_{km}) d\mathrm{u} \right\} \\
&= \sum_{n=1,\mathbf{x}_{nd}\geq\mu_{kmd}}^{N} \varphi_n(k,m) \left(\frac{(\mathbf{x}_{nd}-\mu_{kmd})^2}{\sigma_{r_{kmd}}^3} \right) \\
&\quad - \sum_{i=1,\mathbf{x}_{nd}\geq\mu_{jd}}^{N} \frac{\varphi_n(k,m)}{\sigma_{r_{kmd}}^3} \left\{ \frac{\int_{\partial_{km}} \mathrm{g}_2\,(\mathrm{u} \mid \xi_{km})\,(\mathrm{u}-\mu_{kmd})^2\, d\mathrm{u}}{\int_{\partial_{km}} \mathrm{g}_2\,(\mathrm{u} \mid \xi_{km})\, d\mathrm{u}} \right\}
\end{aligned}
\tag{28}
$$

The term $\int_{\partial_{km}} \mathrm{g}_2(\mathrm{u}|\xi_{km})(\mathrm{u}-\mu_{kmd})^2 d\mathrm{u}$ can be approximated as below:

$$
\int_{\partial_{km}} \mathrm{g}_2(\mathrm{u}|\xi_{km})(\mathrm{u}-\mu_{kmd})^2 du \approx \frac{1}{M} \sum_{m=1}^{M} (\mathrm{r}_{kmd}-\mu_{kmd})^2 \mathrm{H}(\mathrm{r}_{kmd}|\Omega_{km}) \tag{29}
$$

where $\mathrm{r}_{kmd} \sim \mathrm{g}_2(\mathbf{x}_n|\xi_{km})$ is a set of random variables drawn from the asymmetric Gaussian distribution with $\mathrm{u} \geq \mu_{kmd}$ for the particular component m of the mixture model at the state k. Similarly, the term $\int_{\partial_{km}} \mathrm{g}_2(\mathrm{u}|\xi_{km})d\mathrm{u}$ in Eq. (28) can be approximated as:

$$\int_{\partial_{km}} \mathrm{g}_2(\mathrm{u}|\xi_{km})d\mathrm{u} \approx \frac{1}{M}\sum_{m=1}^{M} \mathrm{H}(\mathrm{r}_{kmd}|\Omega_{km}) \tag{30}$$

Similar approximations are used for $\frac{\partial^2 Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^{\mathrm{old}}\right)}{\partial\sigma_{r_{kmd}}^2}$ as follows:

$$\begin{aligned}
\frac{\partial^2 Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^{\mathrm{old}}\right)}{\partial\sigma_{r_{kmd}}^2} &= -3\sum_{n=1,\mathbf{x}_{nd}\geq\mu_{kmd}}^{N} \varphi_n(k,m)\left(\frac{(\mathbf{x}_{nd}-\mu_{kmd})^2}{\sigma_{r_{kmd}}^4}\right) \\
&- \sum_{n=1,\mathbf{x}_{nd}\geq\mu_{jd}}^{N} \varphi_n(k,m)\left(\frac{-2}{\sigma_{r_{kmd}}^3\left(\sigma_{l_{kmd}}+\sigma_{r_{kmd}}\right)}\right)\times \\
&\left\{\frac{\frac{1}{M}\sum_{m=1}^{M}(\mathrm{r}_{kmd}-\mu_{kmd})^2\,\mathrm{H}(\mathrm{r}_{kmd}\mid\Omega_{km})}{\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{kmd}\mid\Omega_{km})}\right\} \\
&- \sum_{n=1,\mathbf{x}_{nd}\geq\mu_{kmd}}^{N} \frac{\varphi_n(k,m)}{\sigma_{r_{kmd}}^6}\left\{\frac{\frac{1}{M}\sum_{m=1}^{M}(\mathrm{r}_{kmd}-\mu_{kmd})^4\,\mathrm{H}(\mathrm{r}_{kmd}\mid\Omega_{km})}{\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{kmd}\mid\Omega_{km})}\right\} \\
&- \sum_{n=1,\mathbf{x}_{nd}\geq\mu_{kmd}}^{N} \frac{-3\varphi_n(k,m)}{\sigma_{r_{kmd}}^4}\left\{\frac{\frac{1}{M}\sum_{m=1}^{M}(\mathrm{r}_{kmd}-\mu_{kmd})^2\,\mathrm{H}(\mathrm{r}_{kmd}\mid\Omega_{km})}{\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{kmd}\mid\Omega_{km})}\right\} \\
&- \sum_{n=1,\mathbf{x}_{nd}\geq\mu_{kmd}}^{N} \frac{\varphi_n(k,m)}{\sigma_{r_{kmd}}^6}\left\{\frac{\left(\frac{1}{M}\sum_{m=1}^{M}(\mathrm{r}_{kmd}-\mu_{kmd})^2\,\mathrm{H}(\mathrm{r}_{kmd}\mid\Omega_{km})\right)^2}{\left(\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{kmd}\mid\Omega_{km})\right)^2}\right\}
\end{aligned} \tag{31}$$

4.3 Complete Algorithm

The complete learning of BAGMM-HMM is given in Algorithm 1, where $epoch_{max}$ is the maximum number of iterations. The goal of this algorithm is to find the optimal parameters of $\Theta = \{\boldsymbol{\pi}, \mathbf{A}, \Lambda\}$.

The flowchart of this algorithm is shown in Fig. 2. First, we initialize π and transition probability $\mathbf{A}$ with the mean probability according to the number of hidden states and number of mixture components and employ K-Means to initialize

Algorithm 1 Parameters learning for BAGMM-HMM

Input:Dataset $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$, $epoch_{max}$.
Output: $\{\boldsymbol{\pi}, \mathbf{A}, \Lambda\}$.
{**Initialization for** $\Theta = [\pi, A, \Lambda]$ }:
{Expectation Maximization}:
while iterations $\leq epoch_{max}$ **or** relative changes of parameters **not** converged **do**
  {[**E Step**]}:
  for all $[\mathbf{X}_1, \ldots, \mathbf{X}_N]$ do
    Compute $\gamma\left(s_{nk}\right)$ and $\xi\left(s_{n-1,j}, s_{nk}\right)$ using forward–backward algorithm.
    Accumulate sufficient statistics according to Eq. (8)
  {[**M step**]}:
  for all $1 \leq j \leq K$ do
    Update π_k, A_{jk} using Eqs. (11) and (12)
    Update p_{km}^{new}, $\boldsymbol{\mu}_{kmd}^{\text{new}}$, $\sigma_{l_{kmd}}^{\text{new}}$, $\sigma_{r_{kmd}}^{\text{new}}$ & $\boldsymbol{\sigma}_{r_j}$ using Eqs. (16), (17), (21), and (27).
  end for
end while

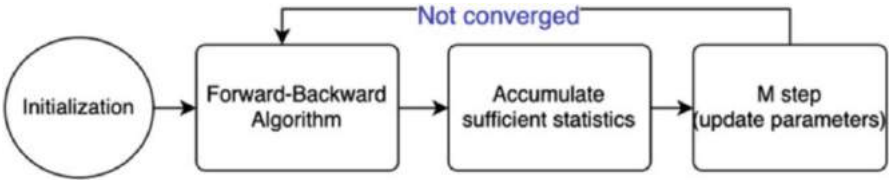


Fig. 2 Training process

parameters of BAGMM. Then, we iterate through the E step and M step until convergence where we accumulate sufficient statistics using the forward–backward algorithm in the E step and update the parameters in the M step.

5 Experimental Results

In this section, the effectiveness of our model is tested on some real-world applications, including occupancy estimation and human activity recognition (HAR). We compare our approach (BAGMM-HMM) with asymmetric Gaussian mixture model hidden Markov model (AGMM-HMM), bounded Gaussian mixture hidden Markov model (BGMM-HMM), and Gaussian mixture model hidden Markov model (GMM-HMM). For comparison, we use the following metrics: accuracy, which is computed as:

$$\left(\frac{TP + TN}{TP + TN + FP + FN}\right)$$

precision, which is computed as:

$$\left(\frac{TP}{TP+FP}\right)$$

recall, which is computed as:

$$\left(\frac{TP}{TP+FN}\right)$$

and specificity, which is computed as:

$$\left(\frac{TN}{TN+FP}\right)$$

In addition, particularly in case of imbalanced dataset, we must also examine the F1 Score, the harmonic mean of precision and recall, which is computed as:

$$2 \times (precision \times recall)/(precision + recall)$$

G-mean 1, the geometric mean of precision and recall, which is computed as:

$$\sqrt{precision \times recall}$$

G-mean 2, the geometric mean of specificity and recall, which is computed as:

$$\sqrt{specificity \times recall}$$

Mathew's correlation coefficient (MCC), which is computed as:

$$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Here, the term TP stands for true positives, TN for true negatives, FP for false positives, and FN stands for false negatives. Here, the term TP stands for true positives, TN for true negatives, FP for false positives, and FN stands for false negatives.

5.1 Occupancy Estimation

Indoor occupancy estimation is a critical analytical task for several applications, such as smart buildings or monitoring the energy consumption for power saving. Automating the devices in a building based on occupancy estimation has proved to

be very efficient since some research works have indicated that one-third of energy can be saved while using this technique [49, 50].

In terms of privacy, most occupancy detection systems and their modeling approaches avoid employing cameras or audio recorders in favor of non-intrusive sensors, which can be divided into two categories: pyroelectric infrared (PIR) sensors and ambient sensors. For the first category, some research works have been proposed to utilize PIR sensors, and ultrasonic sensors [51, 52]. For the second category, some research works [17, 53] have considered environmental features, such as CO_2 human emission, temperature, humidity, and sound level. Moreover, many machine learning approaches have been used to predict occupants, such as Support Vector Machines (SVMs) [52], Logistic Regression [54], and HMMs [15, 55, 56]. They have been utilized to model the extracted features from the environmental data and proved their effectiveness in the occupancy estimation task.

In this section, we employ BAGMM-HMM to estimate occupancy in an office room and hence be the first to tackle this problem with a bounded asymmetric Gaussian mixture-based HMM. Our occupancy estimation task is based on low-cost non-intrusive environmental sensors without bothering privacy policy.

5.1.1 Occupancy Detection Dataset

The dataset of the first experiment for occupancy detection is from UCI machine learning Repository [55]. The experimental data about temperature, humidity, light, the ratio of humidity, and CO_2 were obtained from time-stamped pictures taken every minute, which have two labels, occupied and not occupied, respectively. We select training data from two days with 1993 observations and validation data from four days with 4879 observations, for our experiments.

The results in Table 1 showed promising average accuracy for our BAGMM-HMM as compared to AGMM-HMM, BGMM-HMM, and GMM-HMM: 94.90%, 78.30%, 83.58%, and 76.84%, respectively. These results show the effectiveness of our proposed model for occupancy detection. BAGMM-HMM, AGMM-HMM, and BGMM-HMM converge faster than traditional GMM-HMM because of bounded range support.

In Fig. 3, we present the confusion matrix for this dataset using BAGMM-HMM. Since this is binary classification, our parameters setting is 2 for both the number of hidden states and mixture components. Figure 4 displays the ground truth and our estimated results. From the figures mentioned above, we can see again that our model has an excellent performance.

5.1.2 Occupancy Estimation Dataset

The dataset consists of environmental sensors data collected in an office of Grenoble Institute of Technology, which is housing four people. The dataset comprises luminance, CO_2 concentration, relative humidity (RH), temperature, motion, power

Table 1 Occupancy detection results using different HMM models

	HMM models			
Metrics	BAGMM-HMM	AGMM-HMM	BGMM-HMM	GMM-HMM
Epoch	4	3	3	15
Accuracy	94.90%	78.30%	83.58%	76.84%
Precision	95.83%	88.96%	90.51%	88.59%
Recall	94.90%	78.30%	83.58%	76.84%
Specificity	98.45%	93.70%	91.51%	93.28%
F1-score	95.06%	80.06%	84.80%	78.74%
G-mean 1	95.36%	83.45%	86.97%	82.50%
G-mean 2	96.66%	85.65%	89.22%	84.66%
MCC	87.24%	60.52%	67.49%	58.77%

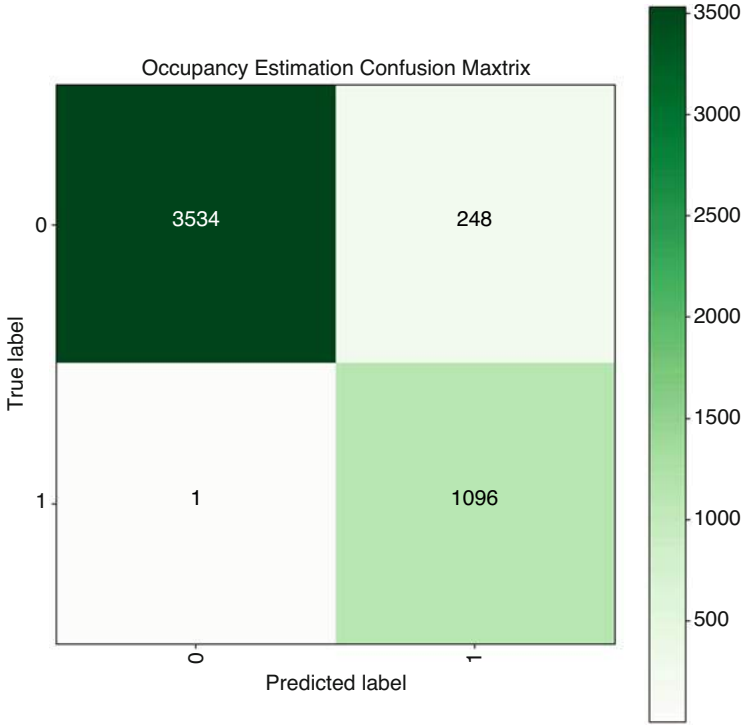


Fig. 3 Occupancy detection confusion matrix for BAGMM-HMM

consumption, window, door position, and acoustic pressure from a microphone. The data collection is performed continuously with an interval of half an hour. The number of occupants is obtained from recorded videos and used for validation only.

The dataset excludes the timestamp and label of occupants, which is observed information, where the number of occupants is the hidden states that we need to

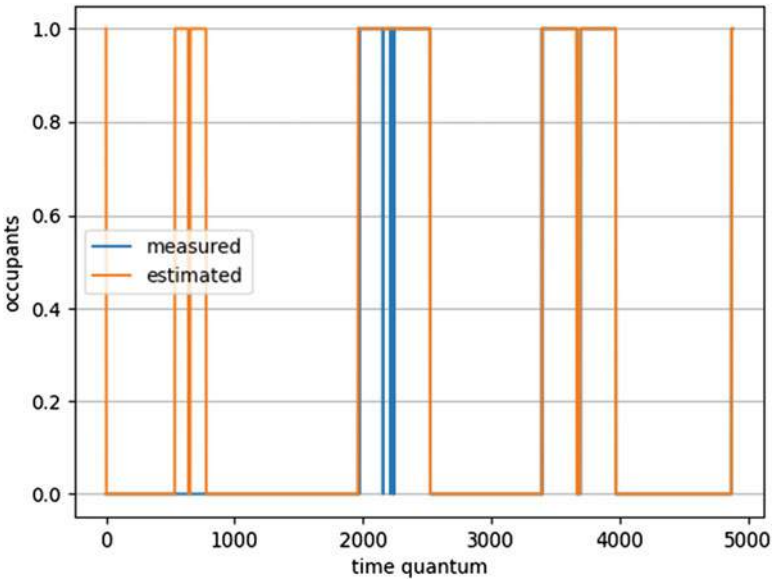


Fig. 4 Occupancy detection using BAGMM-HMM

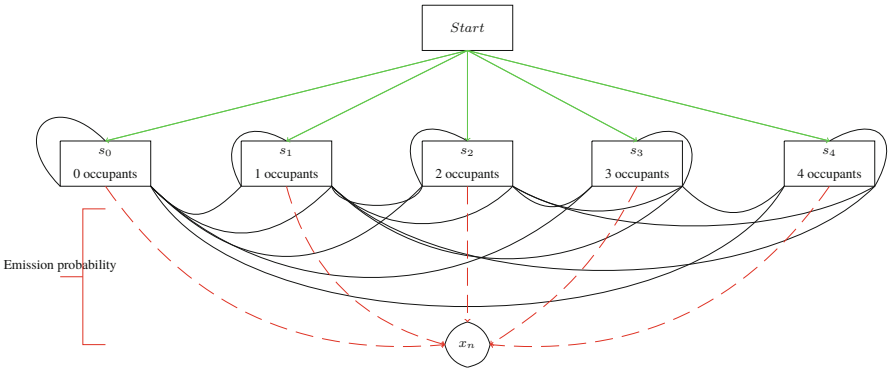


Fig. 5 HMM for occupancy estimation according to the case of study

determine. Eight dimensional sensors outputs over a time interval $t = 30$ minutes represent our data and there are five hidden states $S = \{s_0, s_1, s_2, s_3, s_4\}$ in this dataset as shown in Fig. 5. At time t_0, the number of occupants can be one of the hidden states as shown using green arrows in Fig. 5. Each hidden state may switch to another with the transition probability at any time, as shown using black arrows. The red dashed arrows are the emission probabilities indicating the connections between hidden states and observations at a specific time t_n.

With respect to the choice of features, the research paper [57] indicates that the level of CO_2 does not rise immediately as a person comes in, and the authors

only employed a subset of features for training: {acoustic pressure, occupancy from power, motion counting}. Another consideration is to re-evaluate the nature of the selected emission probability distribution, which is BAGMM in our work. In this experiment, we use all the features except the datetime and occupancy labels. The BAGMM-HMM is trained according to Algorithm 1 to estimate the model parameters that are employed to test the validation dataset.

5.1.3 Experimental Results

The observations in the dataset are collected in the time frame of 20 days every 30 minutes. We choose to train our model using the data collected on days from May 4th, 2015 to May 14th, 2015; test and adjust the model parameters using the data from May 15th, 2015 to May 20th, 2015; validate the model for the rest of data. The compared models are also trained with the same raw data. We just let the models exploit the features and tune the hyperparameters for the models.

After many experiments, the HMM models for our experiments use $K = 5$ for the number of hidden states and $M = 3$ for the number of mixtures to have the best performance. The occupancy estimation comparison results are presented in Table 2. The BAGMM-HMM achieves the best performance with an average accuracy of 86.39% and the highest F1-score with 85.52% compared to 78.45% and 79.28% for AGMM-HMM, 75.42%, and 64.86% for BGMM-HMM against 70.69%, and 75.42% for GMM-HMM, respectively. Our proposed model distinguishes itself as compared to the other models with respect to the considered performance metrics.

The normalized confusion matrix is given in Fig. 6. We notice the dataset is an imbalanced dataset from the confusion matrix. But overall, our model can outperform the other HMM models with the same training data.

Figure 7 presents the results obtained from the BAGMM-HMM with 86.39% accuracy, compared with the ground truth as shown with the blue line.

Table 2 Occupancy estimation comparison using different HMM models

	HMM models			
Metrics	BAGMM-HMM	AGMM-HMM	BGMM-HMM	GMM-HMM
Epoch	4	4	2	10
Accuracy	86.39%	78.45%	75.42%	70.69%
Precision	85.71%	82.91%	56.89%	83.97%
Recall	86.38%	78.45%	75.42%	70.69%
Specificity	75.04%	82.47%	24.57%	88.57%
F1-score	85.52%	79.28%	64.86%	75.42%
G-mean 1	86.05%	80.66%	65.51%	77.05%
G-mean 2	80.52%	80.43%	43.05%	79.13%
MCC	68.35%	57.37%	52.28%	54.39%

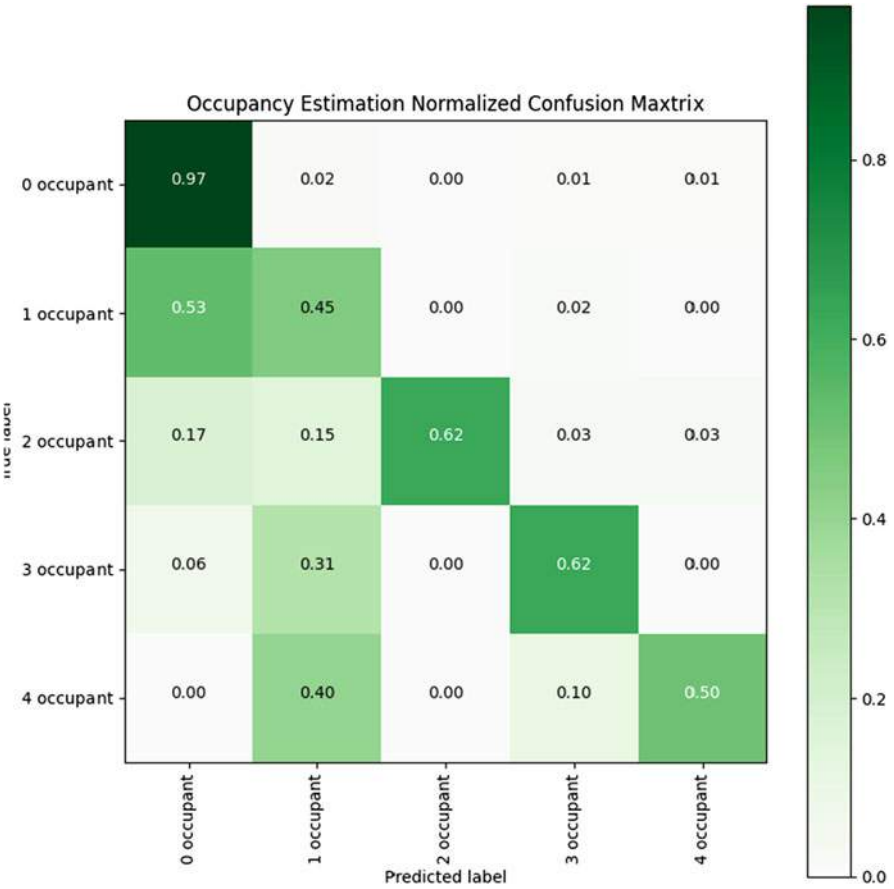


Fig. 6 Occupancy estimation normalized confusion matrix for BAGMM-HMM

5.2 Human Activity Recognition (HAR)

Human activity recognition (HAR) has emerged as an active area of research over the past few years [58, 59] due to many novel ubiquitous applications such as smart buildings, just-in-time surveillance, interactive game interfaces, and home healthcare. The goal of the activity recognition system is to recognize human activities given video clips or environmental sensors data (for privacy concerns) over a time series.

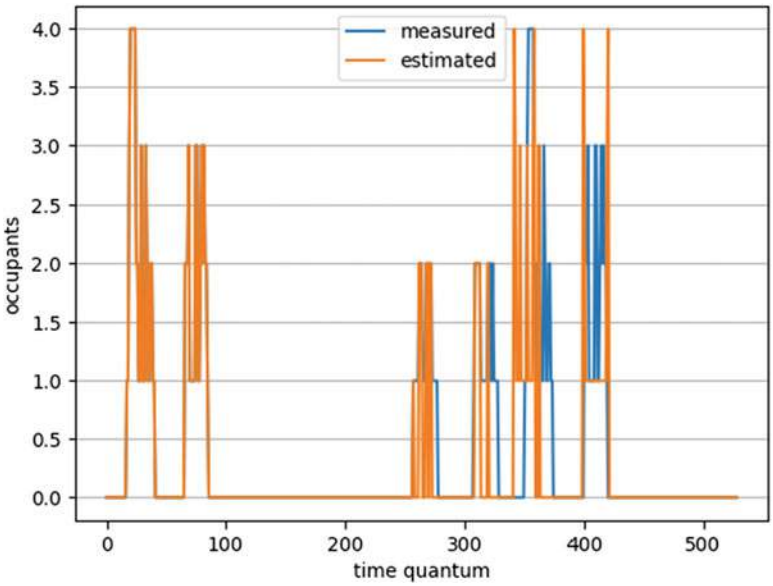


Fig. 7 Occupancy estimation using BAGMM-HMM

5.2.1 HAR Dataset

In this section, we present our experimental results of the proposed model on the challenging human activity recognition (HAR) dataset from UCI machine learning repository [60]. The experiments using this dataset have been carried out with a group of 30 volunteers who performed six activities (walking, walking upstairs, walking downstairs, sitting, standing, laying) wearing a smartphone on the waist. The data comprise 3-axial linear acceleration and 3-axial angular velocity collected by the smartphone's embedded accelerometer and gyroscope at a constant rate of 50 Hz. Besides, the experiments have been video-recorded to label the data manually. The dataset was randomly partitioned into two sets, where 70% of the volunteers were selected to generate the training data and 30% for the test data.

5.2.2 Preprocessing and Data Visualization

We concatenate all the signal data from the Inertial Signals folder, which has nine files, as our training features. However, the combined features are such a large matrix with a size of 7352×1152 to which we applied principal component analysis (PCA) to reduce the dimension from 1152 to 100. We utilize exploratory data

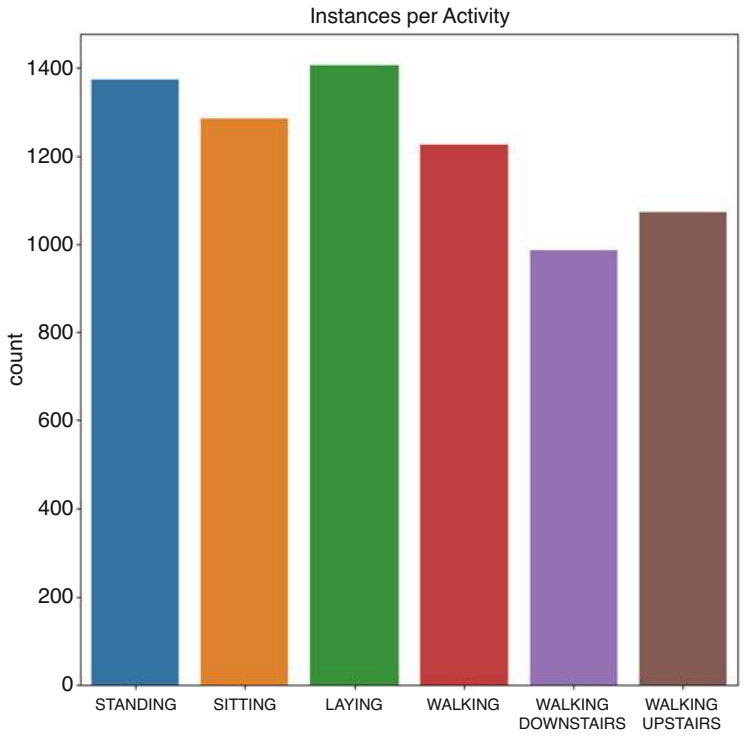


Fig. 8 HAR dataset: instances per activity

analysis (EDA) to analyze the dataset. We notice that the dataset is balanced, as indicated in Fig. 8 that shows the number of data instances per activity.

Furthermore, there are two categorical activities: static (sitting, standing, laying) and dynamic (walking, walking upstairs, walking downstairs) activities, respectively. The body acceleration features in the y-axis are significant in stationary activity while not substantial in moving action, as shown in Fig. 9.

5.2.3 Methodology and Results

An HMM is trained for classifying each human activity using corresponding training data. For the testing stage, the log-likelihood of given testing sensor data is calculated by the respective six trained HMMs, and the class label is assigned according to the maximum likelihood. Our training and predicting process can be observed in Fig. 10.

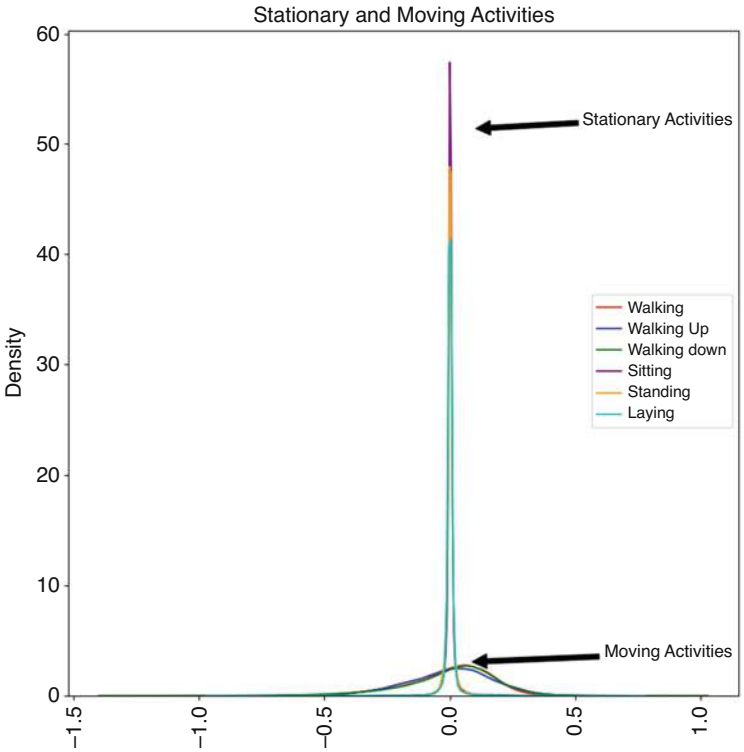


Fig. 9 HAR dataset: stationary and moving activities

Furthermore, our proposed model outperforms other HMMs, with the best configuration being $K = 2$ states and $M = 2$ mixture components associated with each state shown in Table 3. For the sake of time saving, we decrease the number of draws from the asymmetric Gaussian distribution during the M step from 4000 to 1000. The convergence of BAGMM-HMM is faster than the GMM-HMM model. The results obtained with the BAGMM-HMM are indubitably better than those with the HMMs, especially the highest accuracy of 84.64% for BAGMM-HMM.

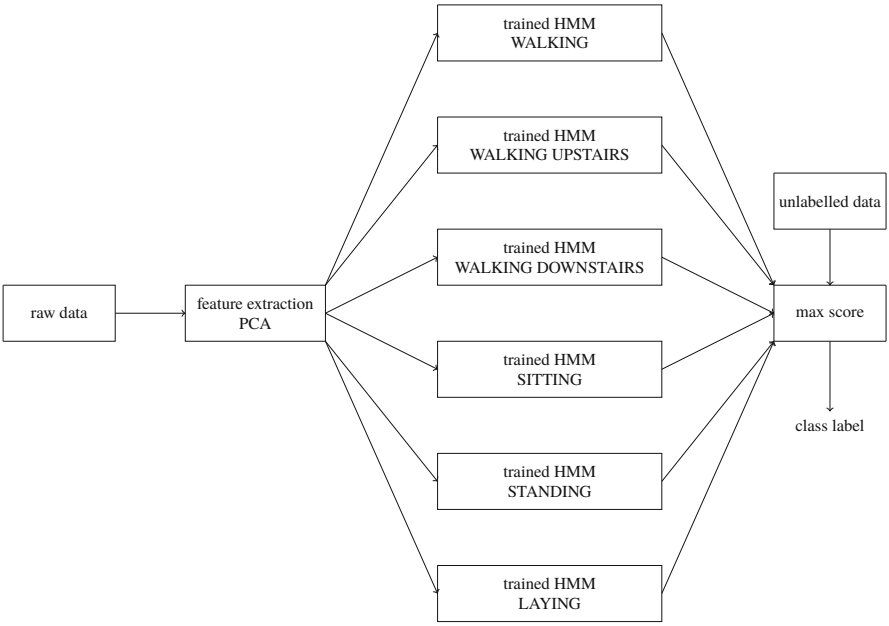


Fig. 10 HMM for activity recognition according to the case of study

Table 3 Activity recognition results using different HMM models

	HMM models			
Metrics	BAGMM-HMM	AGMM-HMM	BGMM-HMM	GMM-HMM
Accuracy	84.62%	77.27%	76.92%	75.00%
Precision	92.31%	69.32%	70.94%	69.44%
Recall	84.62%	77.27%	76.92%	75.00%
Specificity	97.20%	95.24%	24.57%	95.00%
F1-score	83.44%	71.21%	71.64%	68.88%
G-mean 1	88.38%	73.19%	73.87%	72.16%
G-mean 2	90.69%	85.79%	85.45%	84.40%
MCC	83.93%	69.98%	70.16%	68.02%

6 Conclusion

In this chapter, we presented a new extension for the traditional HMM by modifying its emission probability distribution as bounded asymmetric Gaussian mixture. The main goal was to enhance HMM's capability of modeling non-symmetric data with bounded support without performing major modifications on its underlying conventional structure. It is examined from all real-life applications that we have performed that the proposed model outperforms all the comparable Gaussian mixture-based HMMs, including the AGMM-HMM, BGMM-HMM, and the traditional Gaussian

mixture-based HMM. The particular motivation in adopting bounded asymmetric Gaussian mixtures as the emission probability distribution is encouraged by their sound mathematical foundation and excellent capabilities to approximate and model diverse shapes of real-world data. We have proved that our proposed approach is very efficient for occupancy estimation in the context of smart buildings and for activities recognition. Nonetheless, there are still various future works that have been raised to extend this research. Future research could be devoted, for instance, to adding feature selection to improve the modeling of high-dimensional time series datasets or to integrate other probability density functions.

Acknowledgments The completion of this research was made possible thanks to Natural Sciences and Engineering Research Council of Canada (NSERC), the National Natural Science Foundation of China (61876068), and to the framework of the EquipEx program AmiQual4Home ANR-11-EQPX-00 and the cross disciplinary program Eco-SESA.

References

1. L.E. Baum, T. Petrie, Statistical inference for probabilistic functions of finite state Markov chains. Ann. Math. Stat. **37**(6), 1554–1563 (1966)
2. L.E. Baum, J.A. Eagon, An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. Bull. Am. Math. Soc. **73**(3), 360–363 (1967)
3. L.E. Baum, T. Petrie, G. Soules, N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann. Math. Stat. **41**(1), 164–171 (1970)
4. L.E. Baum, An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. Inequalities **3**(1), 1–8 (1972)
5. L.R. Bahl, F. Jelinek, R.L. Mercer, A maximum likelihood approach to continuous speech recognition. IEEE Trans. Pattern Anal. Mach. Intell. **5**(2), 179–190 (1983)
6. J. Baker, The dragon system—an overview. IEEE Trans. Acoust. Speech Signal Process. **23**(1), 24–29 (1975)
7. L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
8. E. Epaillard, 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 Recognit. **55**, 125–136 (2016)
9. L. Batista, E. Granger, R. Sabourin, Dynamic selection of generative–discriminative ensembles for off-line signature verification. Pattern Recognit. **45**(4), 1326–1340 (2012)
10. L.S. Oliveira, E. Justino, C. Freitas, R. Sabourin, The graphology applied to signature verification, in *12th Conference of the International Graphonomics Society* (2005), pp. 286–290
11. E.J. Justino, A. El Yacoubi, F. Bortolozzi, R. Sabourin, An off-line signature verification system using HMM and graphometric features, in *Proc. of the 4th International Workshop on Document Analysis Systems* (2000), pp. 211–222
12. J.K. Aggarwal, Q. Cai, Human motion analysis: a review. Comput. Vis. Image Underst. **73**(3), 428–440 (1999)
13. E. Epaillard, N. Bouguila, D. Ziou, Classifying textures with only 10 visual-words using hidden Markov models with Dirichlet mixtures, in *International Conference on Adaptive and Intelligent Systems* (Springer, Berlin, 2014), pp. 20–28

14. Y. Qiao, L. Weng, Hidden Markov model based dynamic texture classification. IEEE Signal Process. Lett. **22**(4), 509–512 (2014)
15. M. Amayri, Q.-D. Ngo, S. Ploix et al., Bayesian network and hidden Markov model for estimating occupancy from measurements and knowledge, in *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 2 (IEEE, Piscataway, 2017), pp. 690–695
16. B. Ai, Z. Fan, R.X. Gao, Occupancy estimation for smart buildings by an auto-regressive hidden Markov model, in *2014 American Control Conference* (IEEE, Piscataway, 2014), pp. 2234–2239
17. R. Nasfi, M. Amayri, N. Bouguila, A novel approach for modeling positive vectors with inverted Dirichlet-based hidden Markov models. Knowl. Based Syst. **192**, 105335 (2020)
18. M. Bicego, U. Castellani, V. Murino, A hidden Markov model approach for appearance-based 3D object recognition. Pattern Recognit. Lett. **26**(16), 2588–2599 (2005)
19. H. Lee, D. Lee, H.-J. Lee, A predictive initialization of hidden state parameters in a hidden Markov model for hand gesture recognition, in *2018 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)* (IEEE, Piscataway, 2018), pp. 206–212
20. T.K. Moon, The expectation-maximization algorithm. IEEE Signal Process. Mag. **13**(6), 47–60 (1996)
21. S.A. Frank, The common patterns of nature. J. Evol. Biol. **22**(8), 1563–1585 (2009)
22. A. Hyvärinen, P. Hoyer, Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces. Neural Comput. **12**(7), 1705–1720 (2000)
23. M.S. Allili, N. Bouguila, D. Ziou, Finite general Gaussian mixture modeling and application to image and video foreground segmentation. J. Electron. Imaging **17**(1), 1–13 (2008)
24. T. Elguebaly, N. Bouguila, Bayesian learning of finite generalized Gaussian mixture models on images. Signal Process. **91**(4), 801–820 (2011)
25. T. Elguebaly, N. Bouguila, Bayesian learning of generalized Gaussian mixture models on biomedical images, in *Artificial Neural Networks in Pattern Recognition, 4th IAPR TC3 Workshop, ANNPR 2010, Cairo, Egypt, April 11–13, 2010. Proceedings*, ed. by F. Schwenker, N.E. Gayar. Lecture Notes in Computer Science, vol. 5998 (Springer, Berlin, 2010), pp. 207–218
26. T. Elguebaly, N. Bouguila, Infinite generalized Gaussian mixture modeling and applications, in *Image Analysis and Recognition - 8th International Conference, ICIAR 2011, Burnaby, BC, Canada, June 22–24, 2011. Proceedings, Part I*, ed. by M. Kamel, A.C. Campilho. Lecture Notes in Computer Science, vol. 6753 (Springer, Berlin, 2011), pp. 201–210
27. T. Elguebaly, N. Bouguila, A nonparametric Bayesian approach for enhanced pedestrian detection and foreground segmentation, in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2011, Colorado Springs, CO, USA, 20–25 June, 2011* (IEEE Computer Society, Washington, 2011), pp. 21–26
28. T. Elguebaly, N. Bouguila, Generalized Gaussian mixture models as a nonparametric Bayesian approach for clustering using class-specific visual features. J. Vis. Commun. Image Represent. **23**(8), 1199–1212 (2012)
29. J. Lindblom, J. Samuelsson, Bounded support gaussian mixture modeling of speech spectra. IEEE Trans. Speech Audio Process. **11**, 88–99 (2003)
30. M. Azam, N. Bouguila, Multivariate-bounded Gaussian mixture model with minimum message length criterion for model selection. Expert Syst. **38**(2), e12688 (2021)
31. M. Azam, N. Bouguila, Speaker verification using adapted bounded Gaussian mixture model, in *2018 IEEE International Conference on Information Reuse and Integration (IRI)* (IEEE, Piscataway, 2018), pp. 300–307
32. T.M. Nguyen, Q.J. Wu, H. Zhang, Bounded generalized Gaussian mixture model. Pattern Recognit. **47**(9), 3132–3142 (2014)
33. M. Azam, N. Bouguila, Bounded generalized Gaussian mixture model with ICA. Neural Process. Lett. **49**, 1299–1320 (2019)

34. M. Azam, N. Bouguila, Blind source separation as pre-processing to unsupervised keyword spotting via an ICA mixture model, in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)* (IEEE, Piscataway, 2018), pp. 833–836
35. M. Azam, B. Alghabashi, N. Bouguila, *Multivariate Bounded Asymmetric Gaussian Mixture Model* (Springer International Publishing, Cham, 2020), pp. 61–80
36. L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
37. T. Elguebaly, N. Bouguila, Background subtraction using finite mixtures of asymmetric Gaussian distributions and shadow detection. Mach. Vis. Appl. **25**(5), 1145–1162 (2014)
38. T. Elguebaly, N. Bouguila, Simultaneous high-dimensional clustering and feature selection using asymmetric Gaussian mixture models. Image Vis. Comput. **34**, 27–41 (2015)
39. S. Fu, N. Bouguila, Bayesian learning of finite asymmetric Gaussian mixtures, in *Recent Trends and Future Technology in Applied Intelligence - 31st International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2018, Montreal, QC, Canada, June 25–28, 2018, Proceedings*, ed. by M. Mouhoub, S. Sadaoui, O.A. Mohamed, M. Ali. Lecture Notes in Computer Science, vol. 10868 (Springer, Berlin, 2018), pp. 355–365
40. S. Fu, N. Bouguila, Asymmetric Gaussian-based statistical models using Markov chain Monte Carlo techniques for image categorization, in *17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018, Orlando, FL, USA, December 17–20, 2018*, ed. by M.A. Wani, M.M. Kantardzic, M.S. Mouchaweh, J. Gama, E. Lughofer (IEEE, Piscataway, 2018), pp. 1205–1208
41. S. Fu, N. Bouguila, A Bayesian intrusion detection framework, in *2018 International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2018, Glasgow, June 11–12, 2018* (IEEE, Piscataway, 2018), pp. 1–8
42. S. Fu, N. Bouguila, Asymmetric Gaussian mixtures with reversible jump MCMC, in *2018 IEEE Canadian Conference on Electrical & Computer Engineering, CCECE 2018, Quebec, QC, May 13–16, 2018* (IEEE, Piscataway, 2018), pp. 1–4
43. S. Fu, N. Bouguila, A soft computing model based on asymmetric Gaussian mixtures and Bayesian inference. Soft Comput. **24**(7), 4841–4853 (2020)
44. L. Rabiner, B. Juang, An introduction to hidden Markov models. IEEE ASSP Mag. **3**(1), 4–16 (1986)
45. M. Bicego, U. Castellani, V. Murino, A hidden Markov model approach for appearance-based 3d object recognition. Pattern Recognit. Lett. **26**(16), 2588–2599 (2005)
46. S.E. Levinson, L.R. Rabiner, M.M. Sondhi, An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. Bell Syst. Tech. J. **62**(4), 1035–1074 (1983)
47. E. Andrade, S. Blunsden, R. Fisher, Hidden Markov models for optical flow analysis in crowds, in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1 (2006), pp. 460–463
48. C.M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer, New York, 2006)
49. J. Brooks, S. Kumar, S. Goyal, R. Subramany, P. Barooah, Energy-efficient control of underactuated HVAC zones in commercial buildings. Energy Build. **93**, 160–168 (2015)
50. V.L. Erickson, M.Á. Carreira-Perpiñán, 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
51. P. Liu, S.-K. Nguang, A. Partridge, Occupancy inference using pyroelectric infrared sensors through hidden Markov models. IEEE Sens. J. **16**(4), 1062–1068 (2016)
52. J. Petersen, N. Larimer, J.A. Kaye, M. Pavel, T.L. Hayes, SVM to detect the presence of visitors in a smart home environment, in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (2012), pp. 5850–5853
53. H. Rahman, H. Han, Bayesian estimation of occupancy distribution in a multi-room office building based on co 2 concentrations. Build. Simul. **11**(3), 575–583 (2018)

54. M. Snyder, M. Freeman, S. Purucker, C. Pringle, Using occupancy modeling and logistic regression to assess the distribution of shrimp species in lowland streams, Costa Rica: Does regional groundwater create favorable habitat? Freshw. Sci. **35**, 80–90 (2015)
55. L.M. Candanedo, V. Feldheim, Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. Energy Build. **112**, 28–39 (2016)
56. B. Dong, B. Andrews, K.P. Lam, M. Höynck, R. Zhang, Y.-S. Chiou, D. Benitez, An information technology enabled sustainability test-bed (ITEST) for occupancy detection through an environmental sensing network. Energy Build. **42**(7), 1038–1046 (2010)
57. M. Amayri, Q.-D. Ngo, S. Ploix, Estimating occupancy from measurement and knowledge with Bayesian networks, in *2016 International Conference on Computational Science and Computational Intelligence (CSCI)* (IEEE, Piscataway, 2016), pp. 508–513
58. Z. Chen, L. Zhang, Z. Cao, J. Guo, Distilling the knowledge from handcrafted features for human activity recognition. IEEE Trans. Ind. Inf. **14**(10), 4334–4342 (2018)
59. A. Ignatov, Real-time human activity recognition from accelerometer data using convolutional neural networks. Appl. Soft Comput. **62**, 915–922 (2018)
60. D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz et al., A public domain dataset for human activity recognition using smartphones, in *Proceedings of ESANN*, vol. 3 (2013), p. 3

Using HMM to Model Neural Dynamics and Decode Useful Signals for Neuroprosthetic Control

Stefano Diomedi, Francesco Edoardo Vaccari, Kostas Hadjidimitrakis, and Patrizia Fattori

1 Introduction

Neurophysiological recordings consist in driving micro-electrodes in the brain to directly record action potentials (also known as 'spikes') from the neurons. This technique is known to have both an extremely good spatial (10–2 mm) and temporal (10–3 s) resolution at the cost of high invasiveness [1]. Nevertheless, it has contributed enormously to acquire new knowledge on brain function. Historically, starting from Cajal's work at the end of the nineteenth century, the predominant 'neuron doctrine' posited that single neurons were the functional units of the nervous system [2–4], so studying their activity was crucial to test the neuroscientific hypotheses. ¬This approach allowed to gain important insights about the function of sensory areas, especially in the vision domain, since neurons in these areas act as filters applied to the incoming stimuli. However, it fails to account for the more complex associative and motor cortices for which, in the years, discrepancies started to emerge [5–7]. In recent years, the focus of neuroscientific research shifted from single cells to the ensembles of neurons [8] that are currently seen as the basic

Diomedi Stefano and Vaccari Francesco Edoardo contributed equally with all other contributors.

S. Diomedi · F. E. Vaccari · K. Hadjidimitrakis (✉)
Department of Biomedical and Neuromotor Sciences, University of Bologna, Bologna, Italy
e-mail: stefano.diomedi2@unibo.it; francesco.vaccari6@unibo.it; kon.chatzidimitrakis@unibo.it

P. Fattori
Department of Biomedical and Neuromotor Sciences, University of Bologna, Bologna, Italy

Alma Mater Research Institute for Human-Centered Artificial Intelligence, University of Bologna, Bologna, Italy
e-mail: patrizia.fattori@unibo.it

N. Bouguila et al. (eds.), *Hidden Markov Models and Applications*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-99142-5_3

units of brain computations [9]. Thus, it became evident that studying what the single neurons encode might not be the best approach to fully understand the brain functions.

This new population approach was made possible, thanks to technological advances allowing to record simultaneously the activity of large numbers of neurons (in the order of the hundreds or even thousands) while the experimental animals perform complex tasks. The availability of large datasets has prompted the application of novel analytical methods that consider the neural population as a whole (e.g. dimensionality reduction techniques: [10, 11]). These new methods have become especially common in the study of associative and motor areas [12–14].

In the population approach, it is assumed that a neural population recorded within an area, due to the local connectivity, undergoes through a series of 'neural states' that are not observable directly from single cell discharges, but that can be inferred indirectly from the population activity. Under this assumption, the spiking activity of a neural ensemble can be modelled as a stochastic process defined by precise dynamics, termed Hidden Markov Model (HMM). Since the pioneering work by Abeles et al. [15] in motor cortex, HMMs have been widely used to unravel the population dynamics that go beyond the single cell spikes. In brief, the results about the detected neural states can be interpreted following two main strands: for functional research purposes and for decoding relevant information for neuroprosthetic applications. In the first strand, Mazurek et al. [16, 17] investigated the mirror and non-mirror neurons of motor and premotor cortex and reported that the population activity of the former led that of the latter during movement generation. Interestingly, another application of HMM revealed that the activity of the primary motor cortex can be segmented into two distinct neural states that correspond in time with the acceleration/deceleration phases of the arm movements [18].

Other authors exploited the HMM to decode information from neural population activity in order to recognize patterns of activation characteristic of relevant behavioural events. Indeed, it has been possible to efficiently detect transition from a baseline activity to the planning epoch following a target presentation with a few hundred milliseconds delay [19]. The specific target can be decoded directly from the HMM obtaining discrete spatial positions [19] or running a continuous decoder to move a cursor and, in parallel, an HMM to 'click' on a specific letter allowing a faster communication [20].

More details about several past HMM applications in neurophysiology are provided in Sect. 5.2.

2 General Principles OF HMM

Generally speaking, an HMM is a machine learning approach used for modelling time series data [21]. It can be used to detect various patterns present within a time series. These patterns are called 'hidden states' because they are not directly

observable but can be indirectly detected from the time series. At the core of this model there is the interaction between a Markov chain that determines the sequence of the hidden states and a single (or multiple) observable time series called emission sequences. In a Markov process the future state of the system stochastically depends only on the present state and not on the past. In the case of the discrete HMMs, the emission sequences are composed by 'symbols', each of which is emitted by a hidden state with a specific emission probability. These probability distributions can be non-parametric [18, 22] or governed by any parametric distribution. In this regard, typically the Poisson distribution is used in neuroscience [19, 23], but also Gaussian [20]. In the case of continuous HMMs, the observations are continuous (typically drawn by a Gaussian distribution). Theoretical work proved that when an appropriate discretization method is applied, discrete and continuous HMMs reach similar performances [24]. However, since spikes are discrete by definition, discrete HMMs are commonly used in neuroscience and here we will focus on this type of HMM application. Moreover, for simplicity, we will treat the case of a single emission variable HMM and show how it can be adapted for multiple emission variables. The key concepts are still valid for multiple variables HMMs.

An HMM is generally defined by the following elements:

1. The number M of observable symbols in the time series (or emission sequences). These sequences are used to train and validate the model.
2. The number N of the hidden states in the model.
3. The transition matrix ($N \times N$) of the Markov chain that represents the probability to switch from one hidden state to another and so it shapes the topology of the model.
4. The emission matrix ($N \times M$) that contains the probability for each of the N states to emit each of the M symbols.
5. The initial probability vector that indicates the probability to trigger the Markov chain starting from each hidden state.

In summary, every HMM is defined by the transition matrix, emission matrix and the initial probability vector. To estimate these model parameters, the expectation–maximization (EM) algorithm is commonly used. This is an iterative method in which each iteration consists of two sequential steps: the 'expectation' (E) step, where the expected log-likelihood function is computed and the 'maximization' (M) step in which parameters are adjusted to maximize the log-likelihood. A special case of the EM called Baum–Welch algorithm is usually applied to estimate the HMM parameters that use a forward–backward process for the expectation step (see below). However, since Baum–Welch algorithm is conditioned by the random initialization of the model parameters, convergence to the global log-likelihood maximum is not guaranteed. Thus, it is usual to train several models starting from different initial parameters and to select the one with the highest log-likelihood (but other strategies, such as pruning have been proposed; see below). During the testing phase, given an experimental observation, the 'Forward–Backward' algorithm allows to estimate at each time point t the hidden states probability combining information going forward from time 0 to time t (forward density) and

also going from the end of the emission sequence back to time t (backward density). The product between forward and backward densities at time t represents the hidden state probability.

3 Neural Implementation of HMMs

In the previous section, we introduced the general principles behind the Hidden Markov Model. In this section, we will provide concepts and instructions to apply this model to neural activity data recorded from single electrodes or arrays.

Briefly, an electrophysiological dataset consists of the firing activity measured as spike times (time stamps) of each neuron in the population during the task, usually recorded for several trials. It also contains the timing of the task events such as sensory stimuli and/or behavioural responses (e.g. visual/acoustic stimuli, eye/arm movements) that allows to correlate neural activity to the observed behaviour.

In order to model the spikes with HMM, raw data must be converted into emission sequences. If neurons are not simultaneously recorded, their time stamps are aligned at one or multiple events by selecting a fixed temporal window. Then the vector of spike time stamps is binned and converted to the vector of spike counts (i.e. the number of spikes observed in each bin). The choice of the bin width is crucial in order to achieve a good balance between noise/computational load on one hand (narrower bins cause more noisy emission sequences and more data) and temporal resolution needed to appropriately investigate the neural dynamics on the other. For the application proposed here, the binned spike counts (a vector for each unit) were summed up into a unique emission sequence that represented the population activity. For this purpose, we assigned to each emission sequence bin a symbol indicating the neuron that discharged during this time interval (for example, '1' if the first neuron discharged, '2' if the second discharged and so on), whereas to the bins during which no spike was observed '0' was assigned. When more than one cell discharged in the same bin, one symbol was randomly selected [25]. For offline applications, especially when the amount of data is limited (in terms of trial repetitions), the training data can be increased by permuting the trials (for example, [cell1trial1 cell2trial1], but also [cell1trial1 cell2trial2] . . .) and repeating several times the random selection of one neuron from the pool that discharged in the same bin [22]. After these processes, a set of emission sequences that represent the population activity is created.

Before estimating the HMM parameters, the topology of the Markov process must be explicitly defined during the initialization. In particular, two elements have to be considered: the total number of hidden states in the model and which transitions will be allowed and which ones will not.

The number of the states (the so-called order estimation problem) can be chosen arbitrarily or it can be the result of a data driven approach. In this regard, many methods have been proposed. We will briefly present a few of them and then we will provide details about how we tackled this problem in our HMM neural application.

The apparently simplest way to select a model would be to choose a topology type (see below) and build several HMM with an increasing number of states, then to select the model with the highest likelihood (or log-likelihood). Unfortunately, this is not possible, since a more complex model is likely to fit better the data (overfitting the data) rather than a simpler one, thus penalty terms to the likelihood can be introduced to account for the increasing complexity of the model. This is the case of Bayesian Information Criterion (BIC; [26]) and Akaike Information Criterion (AIC; [27]), often used to model selection in the context of HMMs. However, the estimate of an HMM complexity itself (required to compute the penalty term) is not straightforward as for other models (such as linear models, where the number of beta coefficients directly indicate model complexity) and it can be a problem per sè because it dependents on the mere number of states and the number of symbols (or the number of the probability distribution parameters in the case of continuous HMMs), but it also strongly influenced by the topology of the Markov process. Indeed, many estimates of HMM complexity have been adopted, some considering the number of emission matrix elements [28], others the non-zero elements of the transition matrix [29] or both the emission and the transition matrices [24, 30].

In order to obviate the issue of defining the complexity of an HMM, other more complex metrics for model selection, such as Shannon's entropy [31], mixture minimum description length [30], inverse condition number (ICN) of the transition matrix and analysis of HMM residuals [25] have been proposed. Alternatively, different approaches to the problem are possible. We will present in detail the 'consistency analysis' on shuffled data that we performed for our application as well as a couple of 'pruning' techniques.

The 'consistency analysis' involves training several HMMs with an increasing number of states (and a fixed topology) and validating the models on a set of 'test' emission sequences in order to get the corresponding state probability sequences. Then, it is possible to define as 'consistent' the sequences in which the probability of each state exceeds an established threshold (e.g. 0.6, [17]). Accordingly, in a sequence consistent with a 2-state-HMM both two states should have the maximum probability above the threshold and so on for an increasing number of states. In general, when the number of sequences that are consistent with a model increases, the model accounts better for the data. Note that test emission sequences can be shuffled (see below) to avoid overfitting [22]. Similarly, the number of the states in the model can be selected using the log-likelihood on the shuffled emission sequences. Also in this case, a set of HMMs with an increasing state number is trained and then the log-likelihood (i.e. the probability to observe an emission sequence given an HMM) is averaged across the test sequences. The number of states of the HMM with the highest log-likelihood is selected for the subsequent steps. Note that, in this case, the likelihood computed on shuffled data can be seen as a 'cross-validated' likelihood, which is known to be not prone to overfitting when model complexity increases and thus to not require penalty terms (see, for example, [32]).

The topology of an HMM can assume a wide variety of designs, giving the possibility to model many different processes [33]. A 'stationary' process can be

modelled with an ergodic HMM, where all the states are fully connected (i.e. every state transition is possible). When the processes to be modelled have a clear linear or chronological structure, other HMM topologies, such as the simple linear model where each state is only connected with itself (self-transition) and the next state, can be used. To improve the flexibility of a linear HMM, for example, new state transitions can be allowed to skip specific states in the sequence.

The topology can be defined when initializing the transition matrix by manually setting to zero the probabilities that correspond to the forbidden state transitions. Thus, during the training phase the model will never detect these transitions (initialized as 'impossible') and will estimate only the non-zero transition probabilities (initialized as 'possible'). In this way, it is possible, for example, to allow a state sequence that flows from state 1 to state 4, but not vice versa, or allow a transition from state 1 to both 2 and 3. In HMM applications to neural data, often the topology depends on the behavioural task and on the area of the brain used to collect the neural data. Note that the number of states and the topology of the Markov process are defined mainly by the initialization of the transition matrix (sizes, zero and non-zero elements, etc.), thus they are not independent from one another.

As an alternative to manually setting a topology and then estimating the optimal number of states in the model, 'pruning' procedures can be used for choosing the best structure for an HMM application (simultaneously determining the number of states and the topology) with a *semi*-automated approach. In a few words, a complex HMM is initially trained with the maximum number of states to be tested and the topology as general as possible. Then, the least probable state [30] is removed (either from the transition matrix, the emission matrix and the initial state probability vector). The remaining parameters of the pruned model are used to initialize again the training. The procedure is iteratively repeated until some stopping criterion is met (based, for example, on BIC or AIC, see above). Instead of removing the least probable state, during each pruning step it is possible to set to 0 the transition matrix elements that do not cross a threshold [34]. During the subsequent step, the threshold is increased to remove another state and the iterative procedure stops when some criteria is met. The procedures of pruning are not common but allow an automatic choice of the best model structure less biased by human intervention and less sensitive to the random initialization of the parameters [34].

Once the number of states and the topology of the HMM have been chosen, a key step is the initialization of the emission and transition matrices with non-zero elements, since during the training phase the algorithm can converge to local minima. These initial values can be assigned arbitrarily or with a pseudo-random procedure (see next section).

To proceed with model fitting, it is common in the machine learning field to split the dataset (the emission sequences) into two subsets, one for training and one for validation (k-fold or leave-k-out cross-validation). The initialized HMM parameters are fit during the training phase. After that, the validation dataset is used to assess the goodness-of-fit and to obtain the state probability sequences. Results can be then averaged across cross-validations.

The detected states can be further analysed to extract information about the brain area of interest such as the specific sequence of active states as well as the timing of state onsets and offsets. In addition, these neural states (depending on the experimental task) can carry important temporal/spatial information that can thus be indirectly decoded from the neural data. In the following section, we will present a real HMM application to provide an example of both possibilities (functional research and decoding of neural data).

4 HMMs of Parietal Cortex Activity During an Arm Movement Task

In our HMM application the dataset consisted of neural discharges recorded from three different posterior parietal cortical areas, namely V6A, PEc and PE [35–37]. A detailed description about the general and experimental methods can be found in previous reports [38–40]. Data were collected from a monkey trained to perform an instructed delay arm reaching task towards visual targets located in nine spatial positions (three directions × three depths), placed at eye level (Fig. 1a, b). The animal sat on a primate chair in complete darkness and pressed a home button (HB) to begin the trial. Randomly, one of the 9 LEDs lit up green and monkey had to fix its gaze on it. After a variable waiting time, when the LED changed colour to red ('Go' signal), the animal was required to perform an arm reaching movement and hold the position until the LED turned off. Then, the animal returned the hand and pressed the home button (Fig. 1c).

Cells that were recorded for ten correct trials for each target position were taken into account for the analyses, without any further preselection. Our dataset included three different populations of neurons recorded from three distinct parietal cortex areas in the same monkey (V6A:105; PEc: 83; PE: 88 neurons).

For our neural application of HMM, we converted each trial spike train into an emission sequence as described above and repeated this procedure 100 times for

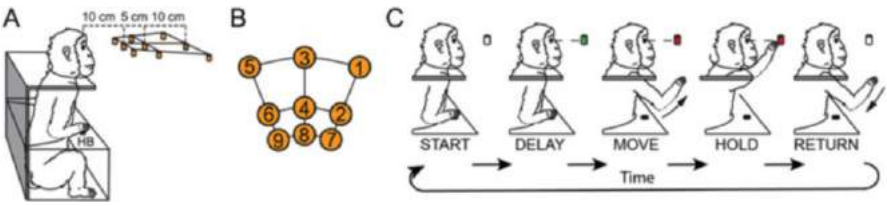


Fig. 1 Experimental Design. (**a**) Experimental setup. Reaching movements were performed toward one of nine LEDs (orange). HB: Home Button. (**b**) Top view of the reaching targets. (**c**) Task sequence, from left to right: trial start (HB press, START), fixation onset and delay phase (DELAY), movement (MOVE), touch and holding of the target for a variable time (HOLD), led switch-off and return of arm to the HB to restart the trial (RETURN)

each trial, thus obtaining 100 emission seq./trial. Since the task had a temporal evolution with well-defined, sequential phases (initial phase, target onset, fixation phase, go signal, movement and hold phase), we assumed that the dynamics of the neural activity related to the task would follow a linear progress as well. We thus built a linear HMM where only transitions from one state to the next or to the state itself were allowed. Accordingly, the diagonal elements of transition matrix (i.e. the probability to remain in a state) were initialized with pseudo-random values in the range [0.99,0.999] and elements above the diagonal (i.e. the probability to proceed to the next state) were set equal to: $(1 - ai,i)/(1 - N)$ where ai,i are the diagonal elements of the transition matrix and N is the total number of the states. The elements under the diagonal were set to zero to prevent the Markov process from moving backwards i.e. (from state 2 back to state 1). We then normalized the matrix rows to obtain a total probability equal to 1. The elements of the emission matrix were initialized as equal to $1/M$ (where M is the number of 'symbols' in the alphabet, i.e. the number of neurons). Because the Baum–Welch maximization is sensitive to the initial values used, during the training phase, we ran the algorithm ten times for every target position starting each time with different initial parameters as detailed above. More details can be found in [22].

To avoid overfitting, the models were cross-validated in two different ways. For the preliminary consistency analysis (i.e. to choose the optimal number of states), we trained the models on emission sequences generated from all the available trials and we decoded sequences generated in the same way, but with an additional step of bin shuffling (i.e. the tth-bin of the jth-sequence was randomly substituted by the tth-bin of the ith-sequence). This atypical cross-validation allowed to test the models on data not completely new, but not identical to the training dataset, with a great computational advance. For all the subsequent analyses, we used a leave-one-out cross-validation (models trained on nine trials and validated on the one left out) and all the subsequent results here reported are referred to the validation dataset, never seen by the models.

We trained the HMMs on data spanning from 1000 ms before and 1000 ms after movement onset (one for each target position, nine models in total). We validated them on the held-out data to study the population neural activity of areas V6A, PEc and PE during the arm movements. The state probability sequences obtained by these models were further analysed to get insights on the function and relationship of the three parietal areas (see below).

As already mentioned, the HMM can be used to decode task related information (i.e. task epoch, reaching target position) from the neural activity with a few additional steps. In our case, in order to decode both the target position and the task phase, we combined the pretrained HMMs in a boosted HMM with a larger number of possible states (also known as 'compound' HMM; [33]). Specifically, we merged the nine (one for each target position) 3-state HMMs already trained. Then, we added the state that corresponded to the task epoch before target presentation called 'FREE' by averaging its parameters across nine different 2-state HMMs (one for each target position) trained on data that spanned from 500 ms before till 500 ms after target presentation. The topology of the resulting 28-state ($9 \times 3 + 1$) boosted

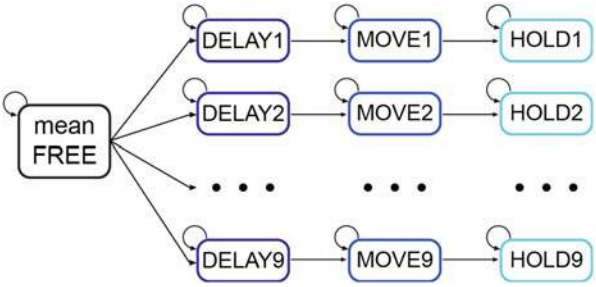


Fig. 2 Schematic representation of the 28-state boosted HMM topology used for the decoding. Arrows indicate which transitions are allowed (probability >0). The first state is averaged across the nine targets ('mean FREE'), while the others are specific for one of the nine targets. For spatial reasons, black dots represent the missing states

HMM is shown in Fig. 2: the first possible state is the 'mean FREE' from which the Markov process can go towards one of the nine possible DELAY states, depending on the target position. Transitions from a state related with a target to a state related with another target (for example, from DELAY1 to MOVE2) were not allowed (Fig. 2).

We based our boosted model on the prior knowledge about the high spatial tuning during reaching movements of the neural activity in the parietal areas we were analysing [38, 39, 41–43]. If it had not been the case, we would have probably tested that the states to put in 'parallel' in the model were different from one another (enabling the target decoding). A viable option would have been to use a 2-way ANOVA on the estimated emission matrices and evaluate the significance of the interaction states*neurons to check whether different states corresponded to different neural modulations [44].

For the decoding, we merged emission sequences generated in the last 500 ms before the target presentation with emission sequences generated in an interval that spanned from 1000 ms before till 1000 ms after the movement onset signal. We then fed the boosted HMM with fragments of the resulting emission sequences obtained with a 200-ms sliding window (10-ms step). We took the neural state with the highest probability averaged across each segment as the output of HMM classifier: the selected states carried information both about the target (for example, DELAY1 vs DELAY2; nine possible targets and FREE, i.e. no target) and the epoch (for example, DELAY1 vs MOVE1; four possible epoch: FREE, DELAY, MOVE and HOLD).

As a measure of the classification performance, we computed the accuracy (also known as 'recognition rate', i.e. the number of correct classifications over the total of classifications). The chance level was calculated shuffling 1000 times the vectors of true class labels for epoch and target separately.

4.1 Functional Characterization of the Parietal Cortex Areas

The application of HMM to the neural data succeeded in identifying hidden neural states in all the three parietal areas of interest (V6A, PEc and PE), while the monkey performed arm reaching movements [22]. A growing number of states (from 2 up to 7) were tested for the consistency analysis and 3 resulted in optimal number to efficiently model the neural population discharges (three states were detected in 100% of the emission sequences for both V6A and PEc, and in the 85.7% of the sequences for PE). When we trained the model to identify more than three states, the number of consistent sequences was greatly reduced. Furthermore, in agreement with the consistency analysis, the models with two and three states had the highest log-likelihoods in all areas and the goodness-of-fit dramatically collapsed as the number of states was increased. Thus, in the next paragraphs we will present the results obtained training 3-state HMMs.

Figure 3a shows the average state probability sequence inferred by the 3-state models previously trained and validated separately for each spatial position reached by the monkey. For the three areas (left, middle and right panels) the probabilities and timing of the individual neural states are quite similar.

The first state (dark blue lines) was always present with high probability at the beginning of the trial and fell shortly before the onset of movement. A second neural state appeared (blue lines) at movement onset and lasted until movement end. The third and last state (light blue lines) increased in probability around the end of the movement and remained stable until the end of the time window analysed.

Just from this first inspection, the neural states seemed to be coincident with the main phases of the task, that is waiting for the 'go' signal, moving the arm towards the target and holding the reached target. Thus, these neural states were likely to represent the neural correlates of the animal behaviour and, for simplicity, we will refer to them as DELAY, MOVE and HOLD.

R2 was computed to measure the similarity between state sequence across the three areas. We calculated the mean state probability sequence across the nine targets for each area, then we reshaped the resulting [3 $\times$ 1000] ([n° of states $\times$ n° of bins]) matrices with state probability to get probability vectors. Pairwise comparisons between these vectors showed high $R2$ values, especially when comparing V6A and PEc ($R2 = 0.97666$, p-value <0.05). PE state sequence resulted slightly different from PEc and V6A at a first glance, but the $R2$ values were still high (V6A vs PE $R2 = 0.95864$, p-value <0.05; PEc vs PE $R2 = 0.95762$ p-value <0.05).

To check the consistency of these results and to extract functional information about the area of interest, we further studied state onset and offset timing.

Given the decoded state probability of an emission sequence, we checked when a state was active or not placing a threshold of 0.7: when the state probability rose above the threshold, we considered the state active (onset) and vice versa when the probability fell under the threshold, the neural state was no more observable (offset). Figure 3b shows the onset and offset timing distributions separately for

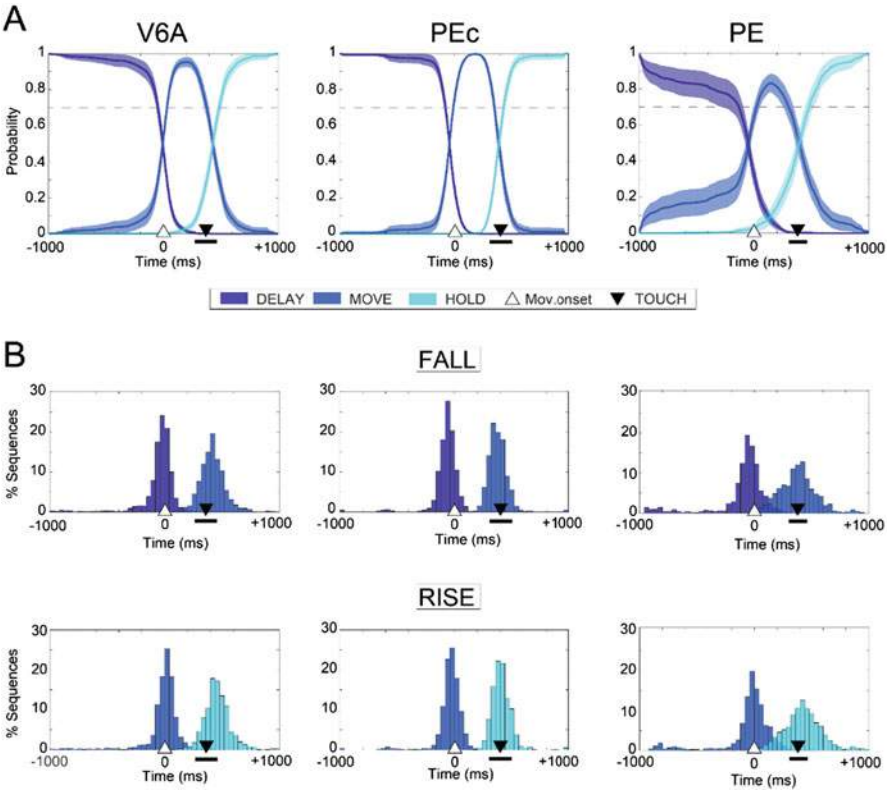


Fig. 3 Neural states in parietal areas and their transitions. (**a**) Average hidden state probability for 3-state HMMs. Coloured lines represent the time course of the probability of each state. Symbols on x-axis represent the main behavioural events averaged across trials and positions; black horizontal bars represent temporal variability. Horizontal dashed line: threshold set to identify the transitions between states (0.7). (**b**) Timing of state rises (i.e. a hidden state probability exceeded the threshold) and fall (i.e. the state probability dropped below the threshold). Y-axis: frequency expressed as percentage of emission sequences. X-axis: time was binned in 40 ms. Other conventions as in (**a**)

each area. Obviously, neural states involved in the same transitions had overlapping onset/offset timing distribution (for example, DELAY offset and MOVE onset greatly overlapped because they were involved in the first neural state transition).

The DELAY state ended at -54 ± 160 (mean $\pm$ SD), -75 ± 119 and -106 ± 232 ms, in V6A, PEc and PE, respectively, with respect to movement onset, whereas the MOVE state rose in V6A, PEc and PE at -14 ± 153, -40 ± 118 and -51 ± 223 ms and fell 415 ± 113, 379 ± 84 and 371 ± 167 ms with respect to the same event. Subsequently, the state HOLD rose at 457 ± 113, 409 ± 85 and 419 ± 166 ms, respectively, for V6A, PEc and PE. The distributions of the transition timing were compared performing a series of Wilcoxon test. From pairwise comparisons, all the distributions resulted significantly different ($p < 0.05$),

even if timing differences were as low as 11 ms, likely due to high number of samples in the distributions (only MOVE state fell timing resulted equal in PEc and PE, $p > 0.05$; timing difference = 8 ms).

The Gini index was computed on the emission matrices to understand how single units participated in the generation of the neural hidden states (similar to [23]; see Sect. 5.2). Gini index ranges between 0 and 1. Values close to 1 indicate that neural states activate only a few units, whereas values close to 0 indicate that the entire population is active to generate the neural states. As reference values, we computed Gini on three sets of synthetic emission matrices simulating three different types of neural populations (10,000 simulations for each set). The first set of emission matrices simulated a population with 1/3 (33%) of the cells active during each state (median Gini index = 0.33), the second set a population with 2/3 of the cells (66%; median Gini index = 0.55) and the last set a population entirely involved in each state (100%; median Gini index 0.77). From our experimental data, we obtained a median Gini index equal to 0.53 (V6A 0.49; PEc 0.54; PE 0.55), similar to the values obtained from the second set of synthetic emission matrices (66% of the population involved in each state). Thus, we could conclude that the majority of the parietal neurons we recorded were active during multiple hidden states.

4.2 Decoding of Task Epoch and Target Position

We built a boosted-HMM algorithm merging HMMs trained separately (see above and Fig. 2 for a schematic representation of the model). The state sequence decoded by this boosted model allowed us to reliably predict the target position and the behavioural epoch using neural data binned in short time windows. Indeed, for epoch decoding, the accuracy was 83%, 88% and 68% (V6A, PEc and PE, respectively; chance level: 28%); for target decoding, the accuracy was 88%, 74% and 40% (V6A, PEc and PE, respectively, chance level: 11%). Figure 4 shows the confusion matrices, a standard visualization method for classification results, in which each element $c(i,j)$ represents the probability to have an observation known to be in class i and predicted to be in class j. Thus, the off-diagonal elements are misclassifications. As it could be expected, errors occurred most frequently between targets close one to another and misclassifications decreased with the increase of the distance ($R = -0.64$; $p = 5 \times 10–10$). Regarding task phase decoding, misclassifications involved most frequently assigning one of the previous states with respect to the correct one rather than assigning one of the following states (i.e., for example, MOVE state, in case of error, was more likely classified as DELAY instead of HOLD).

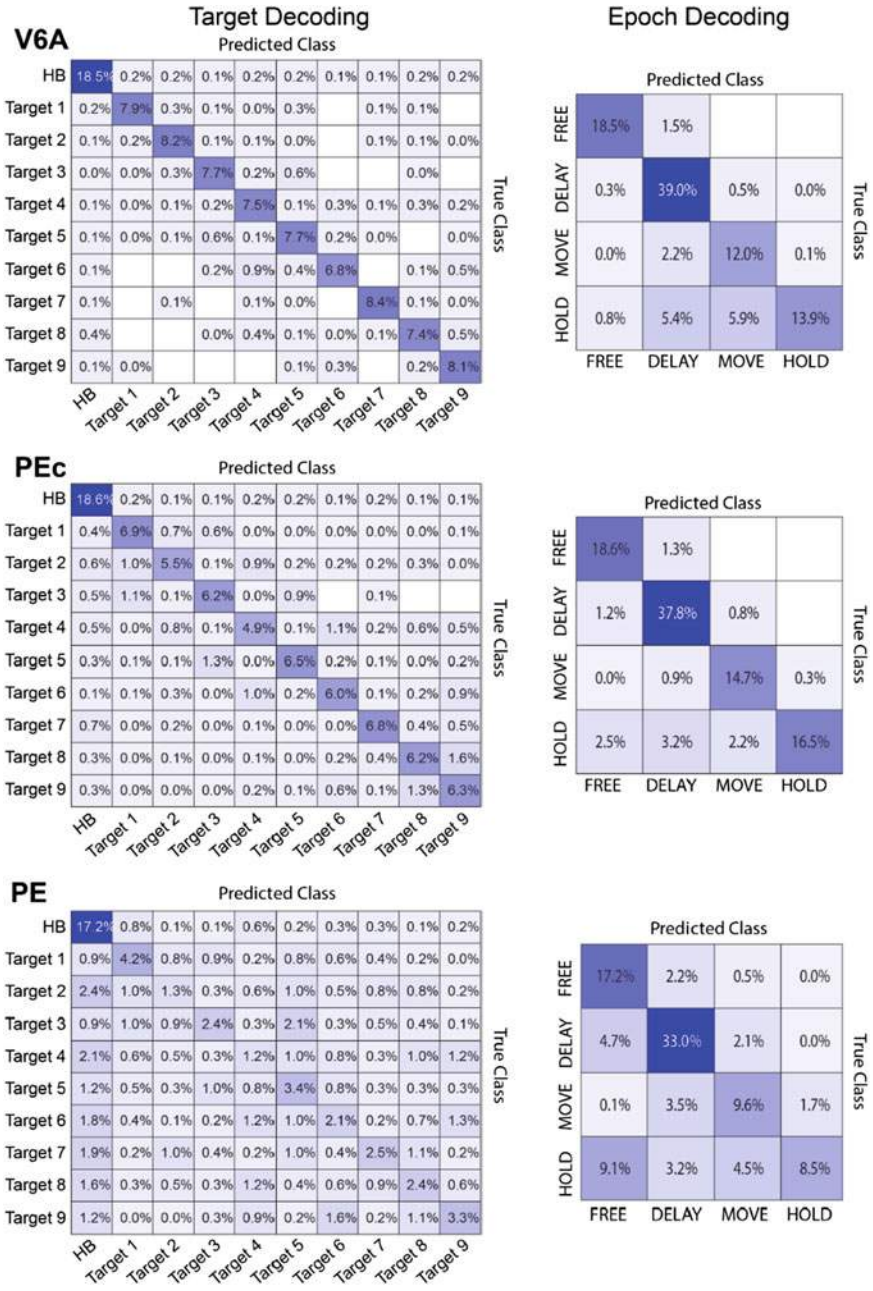


Fig. 4 Confusion matrices describing the errors made by boosted Hidden Markov Model in the recognition of target position (left) and task epochs (right). Results are expressed as percentage of the total count of observations

4.3 HMM: Parietal Cortex Functions and Information Decoding

We showed how HMM can be applied to infer functional properties of individual cortical areas and as a classifier to decode relevant information from neural activity. HMM allowed to automatically segment the population discharges in distinct patterns with a data driven approach. In this regard, the consistency analysis proposed here allows to find the optimal state number, contributing to a further reduction of the bias due to subjective data interpretation with respect to when the number of states is arbitrarily chosen. For example, in a recent HMM application [22], prompted by consistency analysis, we found a set of additional hidden states (reaction time, initial and final movement phases and early part of the hold phase) with respect to the three main states shown in Fig. 3.

Regarding the functional properties of the three parietal areas, the state transition timing showed a temporal gradient in the detection of the onset of the movement state. This transition occurred first in the most anterior area, i.e. PE, whereas it occurred later moving towards the most posterior, PEc and then V6A. This phenomenon is in line with the view of the parietal cortex as a body state estimator [45], receiving an efferent copy of the motor plans produced in the motor cortex of the frontal lobe and comparing the predicted posture with information coming from the visual system (located more posteriorly than the areas we analysed here, in the occipital lobe) and the proprioceptive one (located in the brain regions more anterior than those studied here). The connectivity of the parietal region [35] can explain the temporal gradient that characterized the differential timing of the MOVE state activation.

We also showed the versatility of Hidden Markov Model in decoding relevant information from neural activity, in our case target location and task phase. With a unique algorithm, we were able to decode both these task features.

simultaneously and with high accuracy, especially inV6A and PEc. This is particularly interesting in the field of BMIs, where such a neural decoder can be easily implemented to reliably trigger the movement of a prosthetic artificial arm and to eventually reinforce the target estimation.

In conclusion, the flexibility of this model makes it adaptable to address many neuroscientific issues such as decision making [25], action execution and observation [16, 17], space navigation [23]. Moreover, we showed that this machine learning tool is well suited to functionally characterize a neuron ensemble as well as to decode any information of interest contained in the population activity. For more information on other HMM applications in neurophysiology, see the next section.

5 Related Works

5.1 HMM Applications to Model Neurophysiological Data

In this section, we will present a few works that used HMM to model neurophysiological data, highlighting the key points of these applications, the most relevant differences with the application we proposed and their main findings. First, we will propose a selection of studies that have exploited HMMs to investigate mainly functional aspects of neural activity. Then, we will report works that used HMMs to decode the activity of neurons from a BMI perspective.

Abeles et al. [15] were among the first to apply HMMs in neuroscience to examine whether the cortical activity went through a sequence of distinct neural states in order to produce a particular behaviour, which in their case was moving the arm to reach or withholding the movement. Two types of models, one with a finer and another with a coarser time resolution were used and both detected the same sequence of hidden states, showing a 'global' nature of the detected cognitive states. To evaluate the possibility to predict the behaviour of the animals. Given four different HMMs (each one trained on a different condition of the task) and an unknown spike train, the highest likelihood was used to assess which model could generate the new spike train with a highest probability and thus which behaviour was 'encoded' in the neural activity, reaching an overall accuracy of 90%. In addition, to further investigate the neural hidden states, the authors computed the cross-correlation between pairs of neurons in different states and found that the correlation between single neurons varied considerably across the different states.

Similarly, Bollimunta et al. [25] applied HMM on the activity of lateral intraparietal (LIP) neurons during a random-dot motion direction discrimination task and found that every choice that the animals made about the direction of motion was associated with a specific sequence of hidden states. Furthermore, in some cases, a state sequence could contain states associated with two different choice alternatives indicating changes of mind between two possible future choices.

In another paper, Mazurek et al. [17] used HMM to study populations of mirror and non-mirror neurons. Note that mirror neurons are known to respond during the execution of a motor act, but also during the observation of the same act [46]. They found that, during the observation of an action, mirror neurons encode the same sequence of hidden states (initial, reaction time, movement, final) that they showed during the execution of the same action. Coherently, a generalization analysis proved that HMMs trained on execution trials were able to decode the correct state sequence in observation trials for mirror, but not for non-mirror neurons. In a further study, Mazurek and Schieber [16] compared the transition timing between mirror neurons and non-mirror neurons and they unexpectedly found that mirror neurons anticipated state activation and transition with respect to non-mirror cells.

HMMs have been well suited also to model the activity of hippocampal neurons, known to be sensitive to the spatial position during environment navigation [23]. The authors demonstrated that each hidden state was tightly coupled with a

specific position in space. To investigate the topology of the Markov process, the authors computed the Gini coefficient, a sparsity measure, on the transition matrix concluding that each hidden states were connected only to a few other states. The same measure, computed on the emission matrix, revealed that each detected neural state involved the activation of most parietal neurons in a cooperative way [22].

Kadmon Harpaz et al. [18] published a paper applying HMM to primary motor cortex recordings. They found that the neural activity could be segmented during arm movements in distinct patterns related to acceleration and deceleration phases. As training algorithm, they adapted the standard forward–backward algorithm to handle multiple emission variables and they incorporated the algorithm in a simulated annealing regime, a probabilistic technique which aim is to approximate the global maximum of a target function, in order to reduce the sensitivity to model initialization.

Among the studies that tried to infer the animal behaviour decoding the neural activity, surely one of the most relevant was published by Kemere et al. [19]. The authors modelled the neural activity recorded in the premotor cortex with a multiple emission variable HMM (continuous HMM). They demonstrated the possibility to decode from neural states the different phases of a centre-out reaching task (baseline, preparation, execution) and the direction of the arm movement before movement onset (98% performance). An interesting aspect of the HMM approach used by Kemere and colleagues was the 'supervised' initialization of the model. The initial emission matrix values were assigned equal to the mean firing rate of each neuron in the corresponding task phases. Moreover, the authors explored the trade-off between the latency and the jitter (i.e. the trial-by-trial variability of the latency) of the prediction. Ideally, the latency should be as lower as possible to reduce the delay of the BMI activation and the jitter that can be seen as an indicator of decoding reliability (the lowest, the best), also should be low. The authors showed that both the latency and the jitter varied as a function of the probability threshold applied to decode the hidden states with higher thresholds leading to higher latencies, but lower jitter.

Another important work that leverages HMM for decoding neural activity is that by Kao et al. [20]. The main idea behind this study was to test if the discrete states inferred by HMM could be coupled with the continuous prediction of a different algorithm (an optimized Kalman filter) to enhance the performance of a BMI. This was the case, achieving a performance increase $\approx$5–10% with respect to the state-of-the-art methods in closed-loop experiments. Regarding this HMM implementation, we will report three peculiar features. First, two Markov topologies (with a different number of states) have been used, mainly to account for slight behavioural differences among animals. Second, the complexity of the recorded neural data was first reduced applying a dimensionality reduction technique (PCA) and only the first principal components were retained and used to feed the HMM algorithm. Finally, HMM training was conducted in a totally supervised manner. The emission matrix was built assigning the mean activity values obtained from a training dataset, whereas the transition matrix was learnt by calculating the proportion of transitions between potential states in the training dataset.

In this section we presented a few examples of different applications of HMM to neurophysiological data, trying to highlight the more relevant technical key points that may interest the reader. In the following section, on the other hand, we will discuss advantages and disadvantages of some mathematical tools or machine learning methods that could be used to treat time series instead of HMM.

5.2 Other Approaches to Model Neurophysiological Data

Many other mathematical tools have been exploited to unravel interesting aspects of the neural activity. It is getting more and more common to investigate the activity of a neurons population in the form of 'neural trajectories'. Basically, the idea is to project the recorded activity of single cells onto a set of a few axes (or different variables), reducing the dimensionality and so the complexity of the raw data. For this purpose, many techniques have been used including Principal Component Analysis (PCA, [47–49]) and a multitude of its derivatives, such as jPCA [10] and dPCA [50]; Locally Linear Embedding [51–54]; Gaussian-Process Factor Analysis (GPFA, [55]) and Linear Dynamical Systems (LDSs, [56, 57]).

In general, the greatest difference between these models and HMM is that they all provide a low-dimensional representation of the neural activity with continuous latent variables (or 'states'), whereas Markov segments the spiking data into a few discrete neural states. Thus, 'whereas a HMM would indicate when the switches occur [...], a dynamical model with continuous-valued states would allow one to study the details of how the switching is carried out—in particular, along what path and how quickly' [55]. The choice is between a model (HMM) that assumes stationarity in the neural activity within discrete states and the other models/techniques that assume stationarity of the dynamics along the task (i.e. once the axes/latent variables have been chosen, they are fix for the entire recording). To leverage the advantages of both model types (discrete HMM and continuous LDS), Switching Linear Dynamical Models (SLDSs) have been proposed, based on continuous dynamics hierarchically dependent from a discrete Markov process [58–61].

The decoding of behavioural states directly from cortical activity is a key point in the development of efficient BMIs and other algorithms, besides HMM, have been successfully applied. The Naïve Bayesian Classifier (NB) assured remarkably high performances, despite its simplicity [62–64]. Since this classifier does not incorporate any temporal dependency between the samples, this algorithm is not commonly used to model time series. Recently, interest in neural networks has exploded. To handle time series, Long-Short Term Memory networks (LSTM) are a class of recurrent neural networks (RNNs) extremely interesting. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional recurrent and a feature that is particularly relevant is their capability to deal with lags of unknown duration between important events in a time series. The capacity to learn variable temporal dependencies in the data is a

great advantage over HMM in which, by definition, each time bin depends only on the previous one. A way to obviate to this problem is applying Hidden Semi-Markov Models (HSMMs; [65]) that take into account the duration of the current state making more likely the transition towards the next state after a variable amount of time. HSMMs have rarely been applied in neuroscience with a few examples to analyse fMRI [66, 67] and EEG [68], but never for electrophysiological data, to the best of our knowledge. The great flexibility of LSTMs and the fact that they do not require special assumptions goes at the expense of their complexity and the number of parameters that need to be trained. Accordingly, it has been demonstrated that HMMs outperform LSTMs when the amount of data available is scarce, whereas LSTMs have higher performance when the amount of data increases [24]. Due to the paucity of the neural data in usual electrophysiological studies, until now and the lower computation load required during the training, HMMs have been applied more frequently. Finally, strength of HMMs with respect to other methods such as NB and LSTMs is that they are an unsupervised machine learning technique, whereas either NB (but also other classifiers, such as Support Vector Machines) or neural networks need labelled data to train.

6 Conclusions

To conclude, HMMs represent a broad class of models that can be used in different contexts and processes. In this chapter, we showed how to model neural activity and obtain relevant information about the dynamics of neuronal populations, but also how the very same model can be easily adapted for decoding purposes. This multi-purpose property is hard to find in other types of models, because those that are used to unravel the dynamics are not suited for directly decoding neural activity and vice versa.

In sum, we argued here that HMMs are a powerful tool to analyse time-series data, but like any other statistical method their strengths and weaknesses must be carefully considered to choose the more appropriate model for each individual application.

References

1. P.S. Churchland, T.J. Sejnowski, Perspectives on cognitive neuroscience. Science **242**(4879), 741–745 (1988)
2. S.R. Cajal, Estructura del cerebelo. Gac. Med. Catalana (11), 449–457 (1888)
3. S.R. Cajal, Estructura de los centros nerviosos de las aves. Rev. Trim. Histol. Norm. Patol. **1**, 1–10 (1888)
4. S.R. Cajal, *Textura del sistema nervioso del hombre y de los vertebrados RAMÓN Y CAJAL*, Santiago Editore. (Nicolas Moya, Madrid, 1904)

5. M.S.A. Graziano, T.N. Aflalo, Mapping behavioral repertoire onto the cortex. Neuron **56**(2), 239–251 (2007)
6. M. Omrani, M.T. Kaufman, N.G. Hatsopoulos, P.D. Cheney, Perspectives on classical controversies about the motor cortex. J. Neurophysiol. **118**(3), 1828–1848 (2017)
7. H. Tanaka, Modeling the motor cortex: Optimality, recurrent neural networks, and spatial dynamics. Neurosci. Res. **104**, 64–71 (2016)
8. E. Fetz, Are movement parameters recognizably coded in the activity of single neurons? Behav. Brain Sci. **15**(4), 679–690 (1992)
9. R. Yuste, From the neuron doctrine to neural networks. Nat. Rev. Neurosci. **16**(8), 487–497 (2015)
10. M.M. Churchland, J.P. Cunningham, M.T. Kaufman, J.D. Foster, P. Nuyujukian, S.I. Ryu, K.V. Shenoy, Neural population dynamics during reaching. Nature **487**(7405), 51–56 (2012)
11. J.P. Cunningham, B.M. Yu, Dimensionality reduction for large-scale neural recordings. Nat. Neurosci. **17**(11), 1500–1509 (2014)
12. M.T. Kaufman, M.M. Churchland, S.I. Ryu, K.V. Shenoy, Cortical activity in the null space: Permitting preparation without movement. Nat. Neurosci. **17**(3), 440–448 (2014)
13. V. Mante, D. Sussillo, K.V. Shenoy, W.T. Newsome, Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature **503**(7474), 78–84 (2013)
14. M.G. Stokes, M. Kusunoki, N. Sigala, H. Nili, D. Gaffan, J. Duncan, Dynamic coding for cognitive control in prefrontal cortex. Neuron **78**(2), 364–375 (2013)
15. M. Abeles, H. Bergman, I. Gat, I. Meilijson, E. Seidemann, N. Tishby, E. Vaadia, Cortical activity flips among quasi-stationary states. Proc. Natl. Acad. Sci. U. S. A. **92**(19), 8616–8620 (1995)
16. K.A. Mazurek, M.H. Schieber, Mirror neurons precede non-mirror neurons during action execution. J. Neurophysiol. **122**, 2630–2635 (2019)
17. K.A. Mazurek, A.G. Rouse, M.H. Schieber, Mirror neuron populations represent sequences of behavioral epochs during both execution and observation. J. Neurosci. **38**, 4441–4455 (2018)
18. N. Kadmon Harpaz, D. Ungarish, N.G. Hatsopoulos, T. Flash, Movement decomposition in the primary motor cortex. Cereb. Cortex **29**, 1619–1633 (2019)
19. C. Kemere, G. Santhanam, B.M. Yu, A. Afshar, S.I. Ryu, T.H. Meng, K.V. Shenoy, Detecting neural-state transitions using hidden Markov models for motor cortical prostheses. J. Neurophysiol. **100**(4), 2441–2452 (2008)
20. J.C. Kao, P. Nuyujukian, S.I. Ryu, K.V. Shenoy, A high-performance neural prosthesis incorporating discrete state selection with hidden Markov models. I.E.E.E. Trans. Biomed. Eng. **64**(4), 935–945 (2017)
21. L.R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
22. S. Diomedi, F.E. Vaccari, C. Galletti, K. Hadjidimitrakis, P. Fattori, Motor-like neural dynamics in two parietal areas during arm reaching. Prog. Neurobiol. **1**, 102116 (2021)
23. K. Maboudi, E. Ackermann, L.W. de Jong, B.E. Pfeiffer, D. Foster, K. Diba, C. Kemere, Uncovering temporal structure in hippocampal output patterns. elife **7**, e34467 (2018)
24. M. Tadayon, G. Pottie, Comparative analysis of the hidden markov model and LSTM: a simulative approach. arXiv: Learning (2020)
25. A. Bollimunta, D. Totten, J. Ditterich, Neural dynamics of choice: Single-trial analysis of decision-related activity in parietal cortex. J. Neurosci. **32**(37), 12684–12701 (2012)
26. G. Schwarz, Estimating the dimension of a model. Ann. Stat. **6**(2), 461–464 (1978)
27. H. Akaike, Information theory and the maximum likelihood principle, in *2nd International Symposium on Information Theory*, ed. by B. N. Petrov, F. Csäki, (Akademiai Ki à do, Budapest, 1973)
28. N. Dridi, M. Hadzagic, Akaike and Bayesian information criteria for hidden Markov models. IEEE Sig. Process. Lett. **26**, 302–306 (2019)
29. W. Zucchini, I.L. MacDonald, *Hidden Markov Models for Time Series: An Introduction Using R*, 1st edn. (Chapman and Hall/CRC, 2009)

30. M. Bicego, V. Murino, M.A.T. Figueiredo, A sequential pruning strategy for the selection of the number of states in hidden Markov models. Pattern Recogn. Lett. **24**(9–10), 1395–1407 (2003)
31. B. Roblès, M. Avila, F. Duculty, P. Vrignat, S. Begot, F. Kratz, Methods to choose the best Hidden Markov Model topology for improving maintenance policy. *MOSIM'12 9th International Conference of Modeling, Optimization and Simulation* (Bordeaux, 2012), p. 1. ffhal-00706781f
32. P. Smyth, Model selection for probabilistic clustering using cross-validated likelihood. Stat. Comput. **10**, 63–72 (2000)
33. J.I. Figueroa-Angulo, J. Savage, E. Bribiesca, B. Escalante, L. Sucar, Compound hidden markov model for activity labelling. Int. J. Intell. Syst. **05**, 177–195 (2015)
34. S. Gagnon, J. Rouat, Moving toward high precision dynamical modelling in hidden Markov models. arXiv preprint arXiv:1607.00359 (2016)
35. M. Gamberini, L. Passarelli, P. Fattori, C. Galletti, Structural connectivity and functional properties of the macaque superior parietal lobule. Brain Struct. Funct. **225**(4), 1349–1367 (2020)
36. K. Hadjidimitrakis, S. Bakola, Y.T. Wong, M.A. Hagan, Mixed spatial and movement representations in the primate posterior parietal cortex. Front. Neural Circ. (2019)
37. L. Passarelli, M. Gamberini, P. Fattori, The superior parietal lobule of primates: A sensory-motor hub for interaction with the environment. J. Integr. Neurosci. **20**(1), 157–171 (2021)
38. M. De Vitis, R. Breveglieri, K. Hadjidimitrakis, W. Vanduffel, C. Galletti, P. Fattori, The neglected medial part of macaque area PE: Segregated processing of reach depth and direction. Brain Struct. Funct. **224**(7), 2537–2557 (2019)
39. K. Hadjidimitrakis, G. Dal Bo', R. Breveglieri, C. Galletti, P. Fattori, Overlapping representations for reach depth and direction in caudal superior parietal lobule of macaques. J. Neurophysiol. **114**(4), 2340–2352 (2015)
40. K. Hadjidimitrakis, F. Bertozzi, R. Breveglieri, C. Galletti, P. Fattori, Temporal stability of reference frames in monkey area V6A during a reaching task in 3D space. Front. Neural Circ. **222**(4), 1959–1970 (2017)
41. R. Breveglieri, C. Galletti, G. Dal Bò, K. Hadjidimitrakis, P. Fattori, Multiple aspects of neural activity during reaching preparation in the medial posterior parietal area V6A. J. Cogn. Neurosci. **26**(4), 878–895 (2014)
42. P. Fattori, D.F. Kutz, R. Breveglieri, N. Marzocchi, C. Galletti, Spatial tuning of reaching activity in the medial parieto-occipital cortex (area V6A) of macaque monkey. Eur. J. Neurosci. **22**(4), 956–972 (2005)
43. S. Ferraina, A. Battaglia-Mayer, A. Genovesio, B. Marconi, P. Onorati, R. Caminiti, Early coding of visuomanual coordination during reaching in parietal area PEc. J. Neurophysiol. **85**(1), 462–467 (2001)
44. L.M. Jones, A. Fontanini, B.F. Sadacca, P. Miller, D.B. Katz, Natural stimuli evoke dynamic sequences of states in sensory cortical ensembles. Proc. Natl. Acad. Sci. U. S. A. **104**(47), 18772–18777 (2007)
45. W.P. Medendorp, T. Heed, State estimation in posterior parietal cortex: Distinct poles of environmental and bodily states. Prog. Neurobiol. **183**, 101691 (2019)
46. G. di Pellegrino, L. Fadiga, L. Fogassi, V. Gallese, G. Rizzolatti, Understanding motor events: A neurophysiological study. Exp. Brain Res. **91**(1), 176–180 (1992)
47. R. Levi, R. Varona, Y.I. Arshavsky, M.I. Rabinovich, A.I. Selverston, The role of sensory network dynamics in generating a motor program. J. Neurosci. **25**, 9807–9815 (2005)
48. O. Mazor, G. Laurent, Transient dynamics versus fixed points in odor representations by locust antennal lobe projection neurons. Neuron **48**, 661–673 (2005)
49. M.A.L. Nicolelis, L.A. Baccala, R.C.S. Lin, J.K. Chapin, Sensorimotor encoding by synchronous neural ensemble activity at multiple levels of the somatosensory system. Science **268**(5215), 1353–1358 (1995)

50. D. Kobak, W. Brendel, C. Constantinidis, C.E. Feierstein, A. Kepecs, Z.F. Mainen, X.L. Qi, R. Romo, N. Uchida, C.K. Machens, Demixed principal component analysis of neural population data. elife **5**, e10989 (2016)
51. B.M. Broome, V. Jayaraman, G. Laurent, Encoding and decoding of overlapping odor sequences. Neuron **51**, 467–482 (2006)
52. S.L. Brown, J. Joseph, M. Stopfer, Encoding a temporally structured stimulus with a temporally structured neural representation. Nat. Neurosci. **8**, 1568–1576 (2005)
53. S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding. Science **290**, 2323–2326 (2000)
54. M. Stopfer, V. Jayaraman, G. Laurent, Intensity versus identity coding in an olfactory system. Neuron **39**, 991–1004 (2003)
55. B.M. Yu, J.P. Cunningham, G. Santhanam, S.I. Ryu, K.V. Shenoy, M. Sahani, Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. J. Neurophysiol. **102**(1), 614–635 (2009)
56. J.H. Macke, L. Buesing, J.P. Cunningham, B.M. Yu, K.V. Shenoy, M. Sahani, Empirical models of spiking in neural populations. Adv. Neural Inf. Process. Syst. **24**, 1350–1358 (2011)
57. D. Pfau, E.A. Pnevmatikakis, L. Paninski, Robust learning of low-dimensional dynamics from large neural ensembles. Adv. Neural Inf. Process. Syst. **26**, 2391–2399 (2013)
58. J.I. Glaser, M.R. Whiteway, J. Cunningham, L. Paninski, S.W. Linderman, Recurrent switching dynamical systems models for multiple interacting neural populations. bioRxiv (2020)
59. B. Petreska, B.M. Yu, J.P. Cunningham, G. Santhanam, S.I. Ryu, K.V. Shenoy, M. Sahani, Dynamical segmentation of single trials from population neural data, in *Advances in Neural Information Processing Systems*, (2011), pp. 756–764
60. J. Taghia, W. Cai, S. Ryali, J. Kochalka, J. Nicholas, T. Chen, V. Menon, Uncovering hidden brain state dynamics that regulate performance and decision-making during cognition. Nat. Commun. **9**(1), 2505 (2018)
61. Z. Wei, H. Inagaki, N. Li, K. Svoboda, S. Druckmann, An orderly single-trial organization of population dynamics in premotor cortex predicts behavioral variability. Nat. Commun. **10**(1), 216 (2019)
62. M. Filippini, A.P. Morris, R. Breveglieri, K. Hadjidimitrakis, P. Fattori, Decoding of standard and non-standard visuomotor associations from parietal cortex. J. Neural Eng. **17**(4), 046027 (2020)
63. H. Scherberger, M.R. Jarvis, R.A. Andersen, Cortical local field potential encodes movement intentions in the posterior parietal cortex. Neuron **46**(2), 347–354 (2005)
64. K.V. Shenoy, D. Meeker, S. Cao, S.A. Kureshi, B. Pesaran, C.A. Buneo, A.P. Batista, P.P. Mitra, J.W. Burdick, R.A. Andersen, Neural prosthetic control signals from plan activity. Neuroreport **14**(4), 591–596 (2003)
65. S.Z. Yu, Hidden semi-Markov models. Artif. Intell. **174**(2), 215–243 (2010)
66. S. Faisan, L. Thoraval, J.P. Armspach, M.N. Metz-Lutz, F. Heitz, Unsupervised learning and mapping of active brain functional MRI signals based on hidden semi-Markov event sequence models. IEEE Trans. Med. Imaging **24**(2), 263–276 (2005)
67. H. Shappell, B.S. Caffo, J.J. Pekar, M.A. Lindquist, Improved state change estimation in dynamic functional connectivity using hidden semi-Markov models. NeuroImage **191**, 243–257 (2019)
68. S. Chakravarty, T.E. Baum, J. An, P. Kahali, E.N. Brown, A hidden semi-Markov model for estimating burst suppression EEG, in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, vol. 2019, (2019), pp. 7076–7079

Fire Detection in Images with Discrete Hidden Markov Models

Samr Ali, Md. Hafizur Rahman, and Nizar Bouguila

1 Introduction

For any machine learning model, its capacity depends on how well the features are represented. In discriminative approaches, heavy feature engineering is required to get a proper representation of the target object. Such models perform well when test data follow the same distribution as training data with very low variance. This implies that discriminative models suffer from sample selection bias, i.e., small training data do not represent the whole population, whereas generative models learn probability distribution of the training dataset and can tackle out of distribution datapoints in testing environment.

Hidden Markov models (HMMs) which are models that belong to the generative paradigm learn a sequence of hidden states given any sequential observations. Joint probability distributions of observations and hidden states describe that each observation depends only on the hidden state at a certain timestamp which can be computed by the likelihood. The likelihood of observations can be discrete or continuous depending on the nature of the target object. In image classification, images are represented by a two-dimensional matrix of discrete pixel values and multinomial distribution can be used to compute the likelihood function.

S. Ali
Concordia University, Montreal, QC, Canada

Global Artificial Intelligence Accelerator (GAIA), Ericsson Canada, Montreal, QC, Canada
e-mail: al_samr@encs.concordia.ca; samr.ali@ericsson.com

Md. Hafizur Rahman · N. Bouguila (✉)
Concordia University, Montreal, QC, Canada
e-mail: r_mdhafi@encs.concordia.ca; nizar.bouguila@concordia.ca

N. Bouguila et al. (eds.), *Hidden Markov Models and Applications*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-99142-5_4

This is an interesting venue of research. For instance, the work [1] utilizes multinomial hidden Markov model to detect hostile behaviors by extracting dynamic features from the observations. 1D HMM proposed by [2] uses rescaled features after applying Haar wavelet transformation to surpass the performance of 2D face image recognition system. A second-order HMM proposed by [3] uses 3D state transition matrix for image segmentation. Extracting keypoints such as SIFT from images is a common feature engineering technique in image classification. The work [4] models images by extracting binary symbols corresponding to a 3×3 neighborhood of keypoints which is fed to HMM to learn the optimal state sequences. On the other hand, in Natural Language Processing (NLP), Qiao et al. [5] propose a diversified HMM where transitions follow multinomial distribution and Dirichlet process prior is placed on it to capture diversity of state sequences.

In comparison to the discussed, there is a research gap in modelling dynamic texture with HMM. Dynamic texture is produced from a moving object which is a sequence of images such as fire, sea-waves, smoke, etc. [6–11]. Each image or frame is considered as a timestamp and has certain stationary properties. Because of the nature of such problem, it can be modelled by HMM which has the capability to capture appearance information by encoding observed variables and dynamic properties over time by learning hidden states [12]. The authors [13] proposed an *n*th order HMM to describe the dynamic features by applying the Baum–Welch algorithm. Extracted features are used to classify target object by applying traditional maximum likelihood (ML) criterion.

Forest fire detection is a challenging task given the dynamic texture, shape, and color of fires depending on geographical and environmental factors. With the advancement of image processing techniques, many researchers have proposed different machine learning and deep learning-based solutions to detect fires efficiently and effectively. The authors [14] proposed an ensemble learning technique to improve the detection rate by applying three deep learning models and a decision strategy. Based on spatial features of fire and non-fire images, the authors [15] proposed a method using faster R-CNN model to detect suspected regions of fire. Gathered features are sequentially modelled by an LSTM architecture to decide whether there is fire or no fire in a certain timestamp. The authors [16] proposed a CNN architecture with an efficient technique to compute convolutional filters in Fourier domain for faster detection of wildfire on edge devices. To reduce large variations of dynamic features in fire images, the authors [17] utilize fully convolutional networks (FCNs) [18] to develop an encoder–decoder architecture for efficient segmentation of fire images.

In contrast to deep learning models, traditional fire detection models rely on hand-crafted features such as color, texture, and motion. The authors [19] proposed an early fire detection system by adopting an RGB (red, green, blue) model which makes decision based on intensity and saturation of R component. Brightness information and color information are decoupled by transforming RGB color into a mathematical space. Three decision rules are then used to extract fire pixel from an image and fed to a surveillance system. Due to high volume of data from live feed of surveillance cameras, it is necessary to filter irrelevant information without losing relevant ones. To reduce the number of false positives, the authors

[20] proposed BowFire, a classification model to detect fire in still images. Simple Linear Iterative Clustering (SLIC) is applied to generate super pixel of respective images. The model then uses color and texture features from super pixel regions for detection of fire. The authors [21] proposed a real-time fire detection approach based on HMMs. Moving pixels are first extracted by subtracting the intensity values of subsequent frames and passed through rule-based decision function to check whether they satisfy fire color conditions. Since fire pixels are dynamic in nature as the illumination continuously changes from frame to frame, a clustering technique is applied to determine the fire pixels, which is then used by HMM. A hybrid fire detection technique proposed by [22] uses similar approach to get the set of candidate fire regions. A luminance map is created from sequential observations which is used in combination with HMM as final decision function. Although deep learning-based models have superior performance to traditional approaches, more sophisticated techniques such as knowledge distillation, pruning, quantization, etc. are required to reduce model size for efficient deployment in a surveillance system.

Traditional hidden Markov models consider either high-level features or multi-dimensional transition matrix or complex priors to achieve better performance. For fire detection, HMM is applied to videos which requires moving object detection and dimensionality reduction techniques. There is a research gap in application of HMM to detect fire in still images. In this chapter, we present a simplified training and testing framework to develop an HMM-based classification model. We consider simple 1D representation of raw images by flattening their 2D representation and show that it achieves better performance in detecting fires from images in terms of four different evaluation metrics with faster inference time.

In summary, our contributions are the following:

- We present a technique for the deployment of HMMs in images. We utilize multinomial HMMs to match the underlying properties of the data. These are incarnated in the discrete values of the pixel data.
- We propose an end-to-end computer vision-enabled framework for the detection of fire in small scale image datasets. The system has a superior performance to neural network methods. We do not compare to deep learning methods due to its requirement of large scale of data as well as undue complexity.
- We demonstrate real-time capabilities of the framework.

The remainder of this chapter is organized as follows: Sect. 2 presents the proposed framework, Sect. 3 discusses the results and the experimental setup, and finally, Sect. 4 concludes the chapter.

2 Proposed System

We utilize a hybrid maximum a posteriori (MAP) setup for the HMM parameters and a pure Baum–Welch approach for the multinomial emission distribution. In other words, we set priors on the transition matrix $B_{i,j}, i \in [1, \ldots, K], j \in$

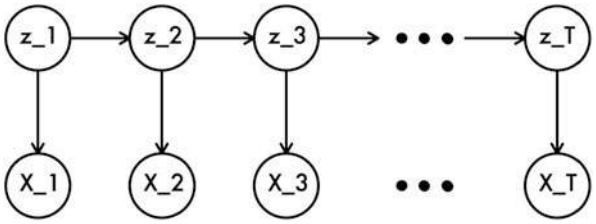


Fig. 1 A typical hidden Markov chain structure representation of a time series where z_1 denotes the first hidden state z_1 and X_1 denotes the corresponding observed state X_1. This is shown accordingly for a time series of length T

$[1, \ldots, K]$, and initial probabilities $\pi_i, i \in [1, \ldots, K]$. This is for K states s at any given time $t \in [1, \ldots, T]$. Both are drawn from a Dirichlet distribution $\mathcal{D}$ such that

$$P(\pi) = \mathcal{D}(\pi|\phi^{\pi}) = \mathcal{D}(\pi_1, \ldots, \pi_K|\phi_1^{\pi}, \ldots, \phi_K^{\pi}) \tag{1}$$

$$P(B) = \prod_{i=1}^{K} \mathcal{D}(b_{i1}, \ldots, b_{iK}|\phi_{i1}^{B}, \ldots, \phi_{iK}^{B}) \tag{2}$$

to satisfy their requirements of probability values that add up to a unit sum (Fig. 1).

B is then defined by $P(s_{t+1} = j|s_t = i)$. The discrete observation set $\Xi_{it}(m) = P(X_t = \xi_m|s_t = i)$ for an observation X_t. This is defined for $(i, t, m) \in [1, K]x[1, T]x[1, M]$ and $\xi = [xi_1, xi_2, \ldots, \xi_M]$, where M is the number of components. Hence, an HMM $\lambda = \{B, \Xi, \pi\}$.

In [23], Rabiner first introduces the three classical problems of HMMs: evaluation or likelihood, estimation or decoding, and training or learning. These are described as follows:

1. The evaluation problem is mainly concerned with computing the probability that a particular sequential or time series datum was generated by the HMM model, given both the observation sequence and the model. Mathematically, the primary objective is computing the probability $P(X|\lambda)$ of the observation sequence $X = X_1, X_2, \ldots, X_T$ with length T given an HMM model λ.
2. The decoding problem finds the optimum state sequence path $I = i_1, i_2, \ldots, i_T$ for an observation sequence X. This is mathematically $\mathbf{s}^* = \text{argmax}_{\mathbf{s}} P(\mathbf{s}|X, \lambda)$.
3. The learning problem refers to building an HMM model through finding or "learning" the right parameters to describe a particular set of observations. Formally, this is performed with maximizing the probability $P(X|\lambda)$ of the set of observation sequences X given the set of parameters determined λ. Mathematically, this is $\lambda^* = \text{argmax}_{\lambda} P(X|\lambda)$.

In the following discussion, we present the respective solutions for the HMM problems that we are concerned with in this chapter, assuming discrete emission

observations. These are the evaluation and the learning problems. We also briefly recall the two conditional independence assumptions that allow for the tractability of the HMM algorithms [24]:

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

$$P(s_t|s_{t-1}, X_{t-1}, \ldots, s_1, X_1) = P(s_t|s_{t-1}) \tag{3}$$

2. Given the tth hidden variable, the tth observation is independent of other variables such that

$$P(X_t|s_T, X_T, s_{T-1}, X_{T-1}, \ldots, s_{t+1}, X_{t+1}, s_t, s_{t-1}, X_{t-1}, \ldots, s_1, X_1) = P(X_t|s_t) \tag{4}$$

2.1 Forward Algorithm

The forward algorithm (Fig. 2) 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 $\rho_t(i) = P(X_1, X_2, \ldots, X_t, i_t = s_i|\lambda)$ 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: $\rho_1(i) = \pi_i \Xi_i(X_1)$, $1 \leqslant i \leqslant K$.

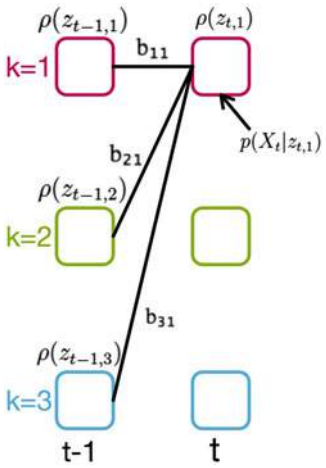


Fig. 2 Graphical representation of the evaluation of the ρ variable of the forward algorithm in an HMM lattice fragment

2. Calculate how state $q_{i'}$ is reached at time $t+1$ from the K possible states s_i, $i = 1, 2, \ldots, K$ at time t and sum the product over all the K possible states: $\rho_{t+1}(j) = \left[\sum_{i=1}^{K} \rho_t(i) b_{ij}\right] \Xi_j(X_{t+1})$ for $t = 1, 2, \ldots, T-1$, $1 \leqslant j \leqslant K$.
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.

2.2 Baum–Welch Algorithm

Figure 3 depicts the computation process of the backward algorithm in an HMM lattice structure. Together with the forward algorithm, this forms the forward–backward algorithm through consequent iteration. In the context of HMMs, the forward–backward algorithm is of extreme importance and is also known as the Baum–Welch algorithm [23]. The Baum–Welch algorithm is traditionally used to solve the estimation problem of HMMs. 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 [25].

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, is the backward algorithm. This has the variable $\theta_t(i) = P(X_{t+1}, X_{t+2}, \ldots, X_T, i_t = s_i|\lambda)$ and is solved as follows:

1. Compute an arbitrary initialization $\theta_T(i) = 1$, $1 \leqslant i \leqslant K$.
2. $\theta_t(i) = \sum_{i'=1}^{K} b_{ii'} \Xi_{i'}(X_{t+1})$ for $t = T-1, T-2, \ldots, 1$, $1 \leqslant i \leqslant K$.

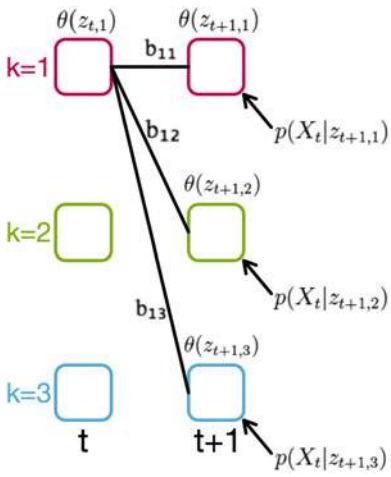


Fig. 3 Graphical representation of the evaluation of the θ variable of the backward algorithm in an HMM lattice fragment

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

$$\varphi_t(i,i') = P(i_t = s_i, i_{t+1} = s_i'|X,\lambda) = \frac{\rho_t(i)b_{ii'}\Xi_{i'}(X_{t+1})\theta_{t+1}(i')}{p(X|\lambda)} \tag{5}$$

where $\varphi_t(i,i')$ 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. $\rho_t(i)$ then considers the first observations ending at state s_i at time t, $\theta_{t+1}(i')$ 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

$$\gamma_t(i) = \sum_{i'=1}^{K} \varphi_t(i,i') \tag{6}$$

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

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

1. $\bar{\pi}_i = \gamma_1(i), 1 \leqslant i \leqslant K$
2. $\bar{b}_{ii'} = \sum_{t=1}^{T-1}\varphi_t(i,i') / \sum_{t=1}^{T-1}\gamma_t(i)$
3. $\bar{\Xi}_{i'}(k) = \sum_{\substack{t=1 \\ X_t=k}}^{T}\gamma_t(i') / \sum_{t=1}^{T}\gamma_t(i')$

2.3 HMM Framework

The proposed setup then constitutes of training a one-class multinomial HMM classifier for the detection, i.e., two classifiers in total. In the remainder of the chapter, we refer to these as *fire detected* and *fire not detected*. A depiction of the proposed framework may be observed in Fig. 4.

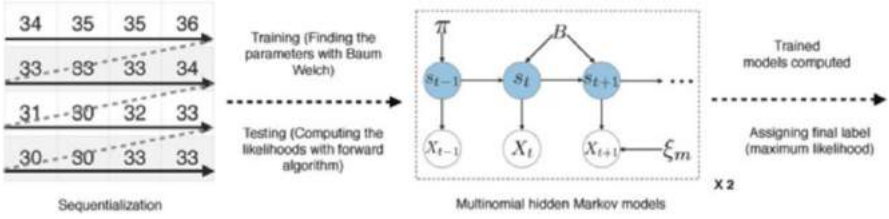


Fig. 4 The proposed framework. The dotted arrows refer to the actions that are carried out. Text above the dotted arrows refers to the training processes, while text below it refers to ones in the testing stage. Sequentialization represents the procedure of converting an input image into a sequential signal for the multinomial HMM to be trained on

We proceed by first extracting the red channel from the image. This follows the logic that fire would have a higher red saturation than other images or the second class in this case. This computer vision technique while simple is quite effective as we observe from the results of proposed model in the next section. The process may also be generalized to a variety of dynamic textures that exist in nature corresponding to their intrinsic properties. For instance, green saturation would be expected to be higher in leaves, trees, grass, or vegetation in nature. Another example is the higher saturation of the blue color for the detection of sea, rivers, or water in general.

Nonetheless, this procedure is not sufficient to handle the detection problem by itself. This is due to the complexity of the problem at hand, i.e., identification of dynamic textures from images. As discussed, dynamic textures are defined as textures that vary across time. Such sequential data has characteristics that present it as an attractive problem to solve with HMMs [26, 27].

Moreover, other classes may still have high red color saturation. We can observe in Sect. 3.1 that the negative class contains high red saturated images which do not belong to the fire class. One such obvious example is a tree in the fall with reddish leaves. As such, a purely color-based approach is not sufficient without another supplement for a successful algorithm. In this chapter, we choose a machine learning approach, i.e., the multinomial HMMs.

We refer to the serialization process to produce a sequence as sequentialization. This transfers an image into a sequence for the training as well as the testing of the proposed model. The process constitutes of collating the pixels from left to right and from top to bottom. This also emulates the way that humans perceive the pixel values if presented to them in a tabular form as in the figure.

In order to deploy our model, we also investigate the optimum number of states to employ for the classifiers. The approach that we carry out for this model selection problem is by exploring the search space of the states to reach the optimum performance in terms of the evaluation metrics. The latter are discussed in the following section.

Once the classifiers are trained, we compute the forward probabilities for each new testing image. This determines the likelihood that this image was generated by an HMM. The final label is then assigned to the maximum resultant likelihood.

As this chapter addresses an application-based novelty, we delve into further details in the following section (Sect. 3). Nonetheless, it is noteworthy to mention that whereas multinomial HMMs and their learning algorithms are established, the deployment of HMMs in the image domain is scarce at best. As such, this chapter addresses a dire need to fill this gap and to further research this topic.

3 Experimental Results

In this section, we present our results. In particular, Sect. 3.1 discusses the dataset that we evaluate our proposed system on. Section 3.2 briefly introduces the evaluation metrics employed. Finally, Sect. 3.3 presents the results.

3.1 Dataset

We evaluate our proposed framework using the forest fire detection dataset.[1] The dataset consists of 1900 3-channel (250–250) training and testing images split equally across the classes, 950 images each. That is we have a balanced dataset and the split for the training and testing is 80% to 20%, respectively. Samples of the dataset may be observed in Figs. 5 and 6 for the fire and the no fire classes, respectively.

However, the images are quite challenging due to its wide variability across the different properties such as the lighting, the background, the visual elements, etc. It is also imperative to mention that utilizing HMMs is motivated for this application by the availability of only relatively small datasets. This is also highlighted by our chosen dataset.

3.2 Evaluation Metrics

We evaluate the performance of the proposed model with the accuracy, precision, recall, and F1-score measures. We also utilize the prior for comparison to neural networks. The accuracy may be defined as the ability of the model to correctly distinguish true positives (TP) and true negatives (TN) out of the complete data; that is

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

where FP and FN represent the false positives and negatives, respectively.

Positive predictive value (PPV) or the precision denotes the percentage of correctly identified positives in the predicted positives by

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

[1] https://data.mendeley.com/datasets/gjmr63rz2r/1.

Fig. 5 Dataset samples of the fire class

Fig. 6 Dataset samples of the no fire class

True positive rate (TPR) or the recall measures the percentage of correctly identified positives in all positives. It is then mathematically denoted by

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

Finally, the harmonic average of the precision and the recall or the F1-score is defined by

$$\text{F1-score} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{10}$$

To demonstrate the real-time capabilities of the system, we record the time that is required in testing the performance of the framework across each frame. This is reported for each model deployed with a different number of states.

3.3 Result Discussions

3.3.1 Performance Evaluation

We have evaluated our HMM with different structure configurations. In particular, we report the performance for $K = 4$, $K = 5$, and $K = 6$. We carried out three runs for each of the models whose confusion matrices we recorded. These may be observed in Figs. 7, 8, and 9 for the 4-state HMM, Figs. 10, 11, and 12 for the 5-state HMM, and finally, Figs. 13, 14, and 15 for the 6-state HMM. The average time to carry out the testing of a sequence or an image is recorded in Table 1. The proposed model displays real-time capabilities that are desirable in sensitive applications such as the one at hand.

It may also be observed that the higher the complexity of the model, the larger the amount of time required for its execution. A good compromise then would be the five-state-based multinomial HMM. Nonetheless, further analysis is required to address the other performance metrics.

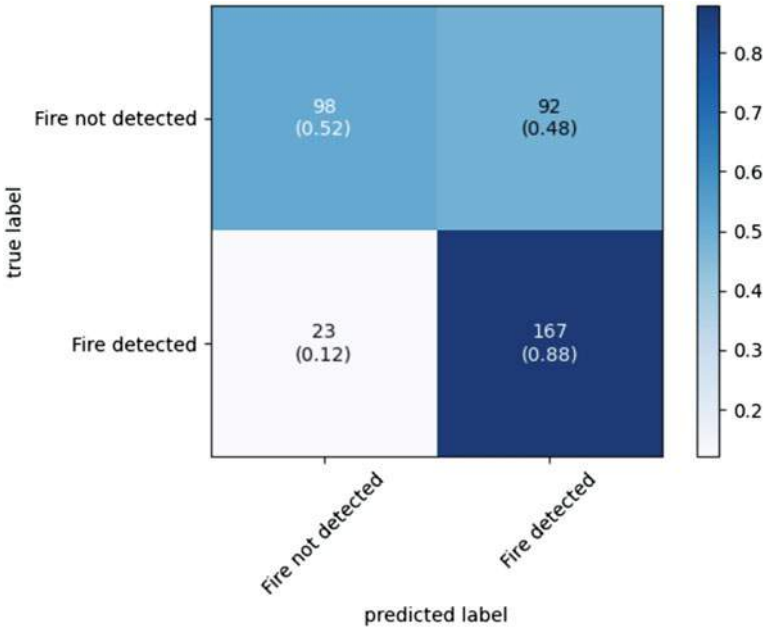


Fig. 7 Confusion matrix of the results for 4-state multinomial HMM—Run 1

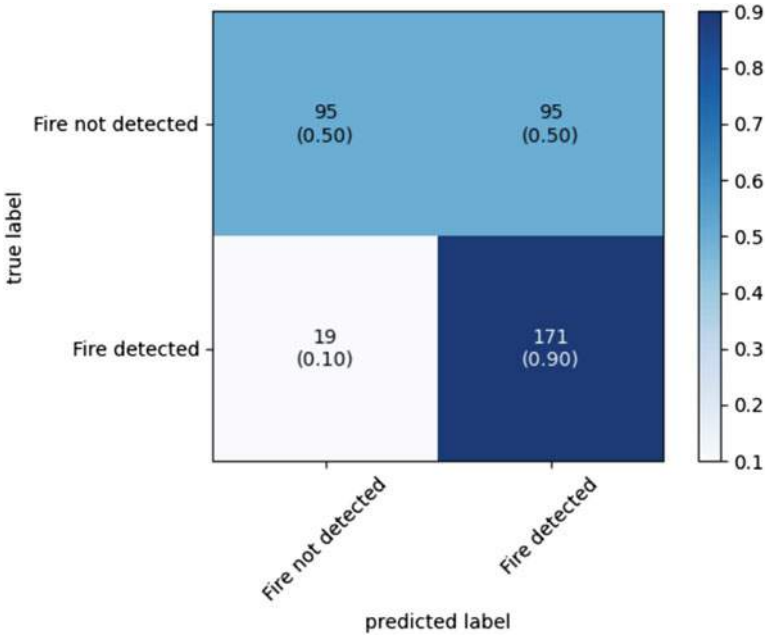


Fig. 8 Confusion matrix of the results for 4-state multinomial HMM—Run 2

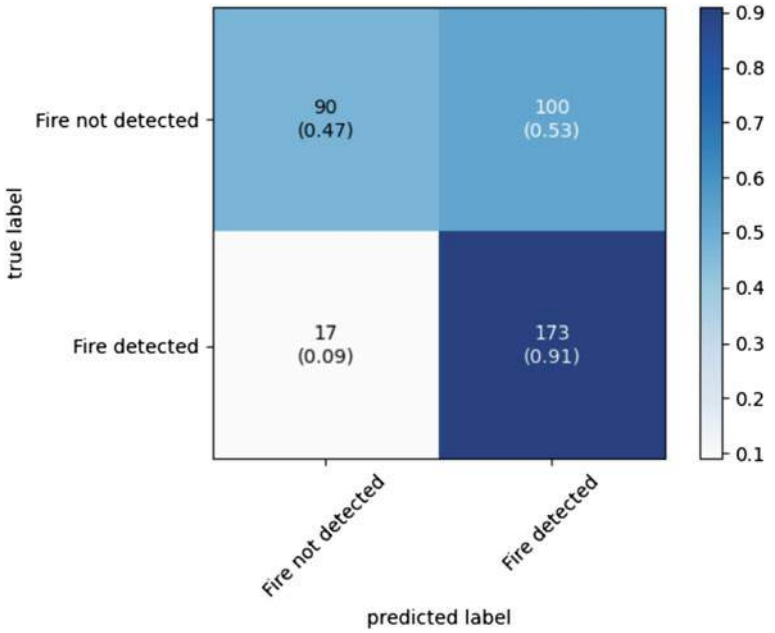


Fig. 9 Confusion matrix of the results for 4-state multinomial HMM—Run 3

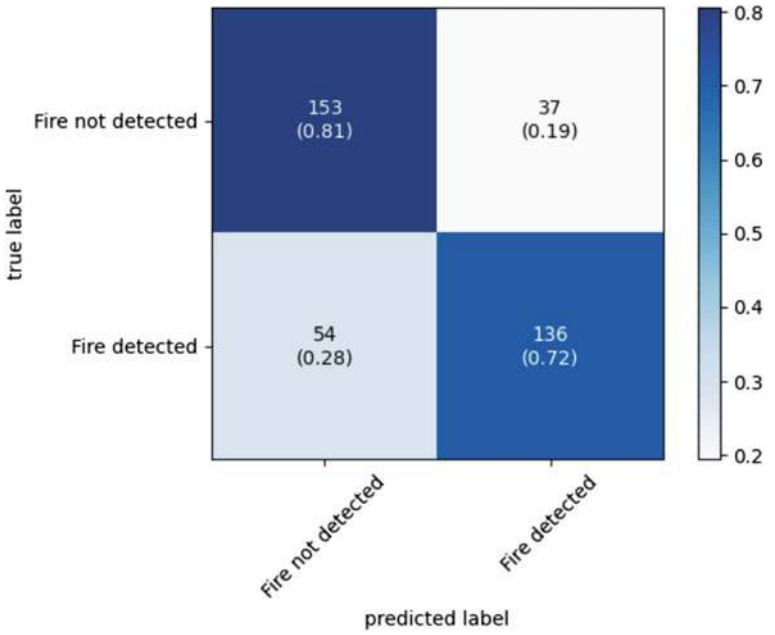


Fig. 10 Confusion matrix of the results for 5-state multinomial HMM—Run 1

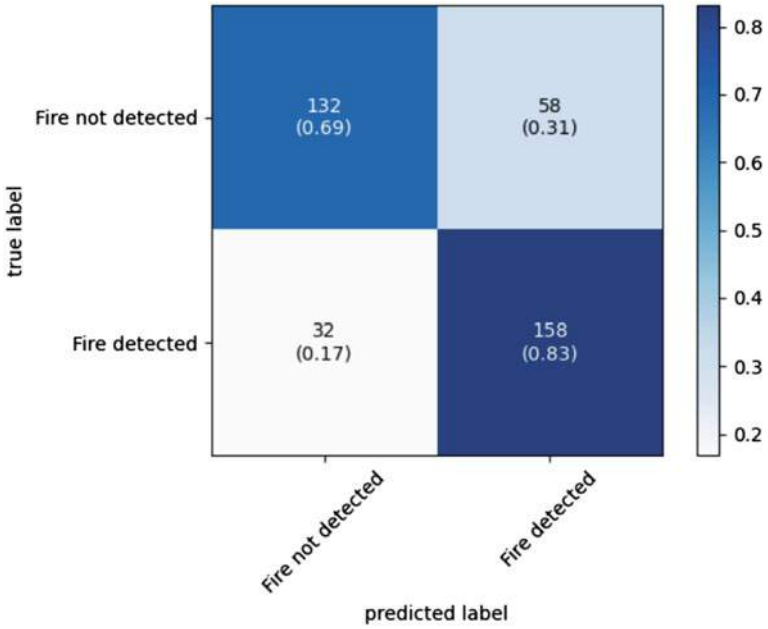


Fig. 11 Confusion matrix of the results for 5-state multinomial HMM—Run 2 (optimum)

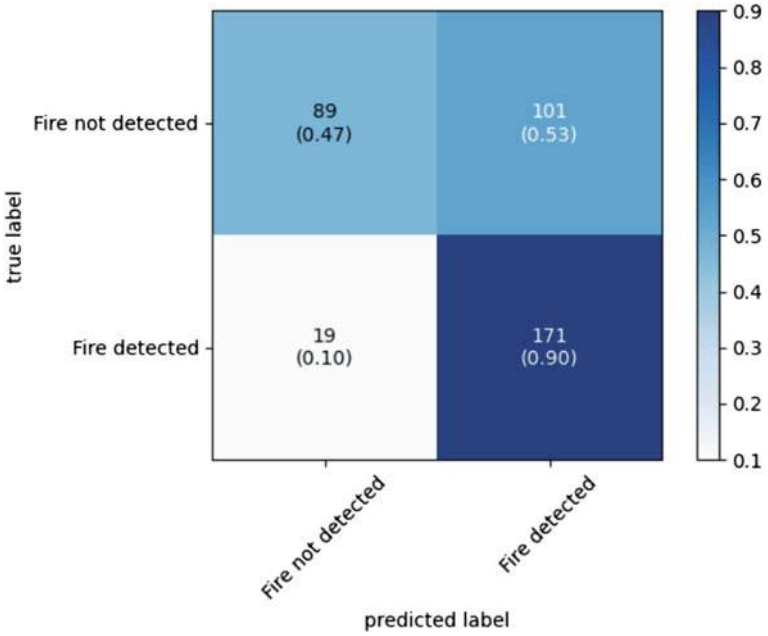


Fig. 12 Confusion matrix of the results for 5-state multinomial HMM—Run 3

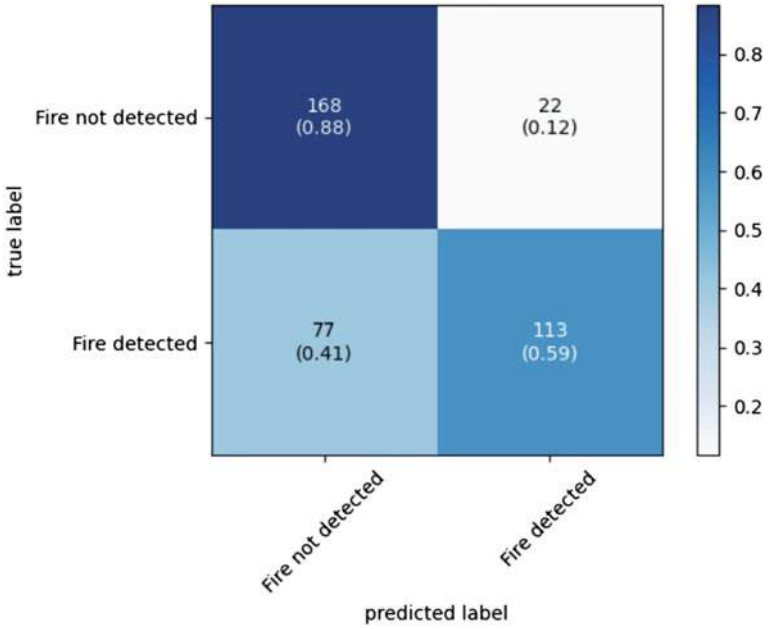


Fig. 13 Confusion matrix of the results for 6-state multinomial HMM—Run 1

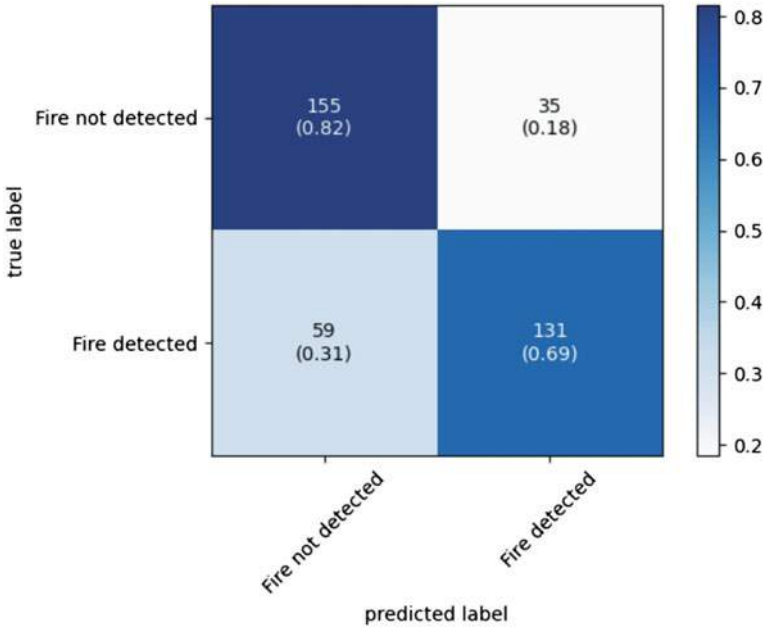


Fig. 14 Confusion matrix of the results for 6-state multinomial HMM—Run 2

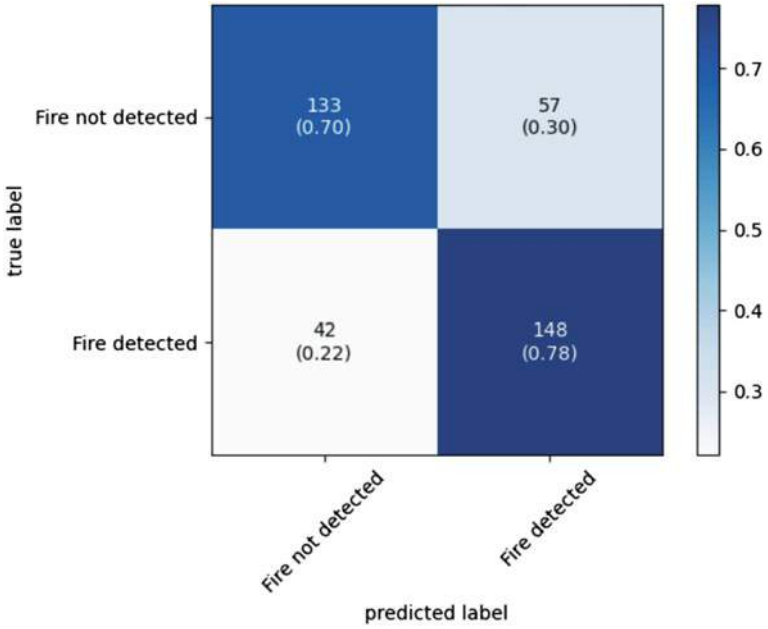


Fig. 15 Confusion matrix of the results for 6-state multinomial HMM—Run 3

Table 1 Time performance of the proposed framework across 3 runs. The maximum value is highlighted in bold

	Time (s)		
Number of States (K)	4	5	6
	0.02	0.02	0.03

Table 2 Accuracy fluctuation of the methods across 3 runs. The maximum value is highlighted in bold

	Accuracy (%)		
Number of States (K)	4	5	6
Run 1	69.74	76.05	73.95
Run 2	70.00	**76.32**	75.26
Run 3	69.21	68.42	73.95
Average	69.65	73.60	74.39

The fluctuation in accuracy, precision, recall, and F1-score may be observed in Tables 2, 3, 4, and 5, respectively.

The accuracy of the models is at a maximum in the second run of the five-state-based multinomial HMM. On the other hand, the highest average belongs to the six-state-based one. Nonetheless, we need to consider other metrics, especially the F1-score measure as it is a mean of the remaining two other measures: precision and recall.

In terms of precision, the average shows that the higher the number of states of a model, the better it performs. Indeed, the maximum precision which is highlighted in bold belongs to the 6-state-based multinomial HMM in its first run. This agrees

Table 3 Precision fluctuation of the methods across 3 runs. The maximum value is highlighted in bold

	Precision (%)		
Number of States (K)	4	5	6
Run 1	64.48	78.61	**83.70**
Run 2	64.29	73.15	78.92
Run 3	63.37	62.87	72.20
Average	64.05	71.54	78.27

Table 4 Recall fluctuation of the methods across 3 runs. The maximum value is highlighted in bold

	Recall (%)		
Number of States (K)	4	5	6
Run 1	87.89	71.58	59.47
Run 2	90.00	83.16	68.95
Run 3	**91.05**	90.00	77.89
Average	89.65	81.58	68.77

Table 5 F1-score fluctuation of the methods across 3 runs. The maximum value is highlighted in bold

	F1-score (%)		
Number of States (K)	4	5	6
Run 1	74.39	74.93	69.54
Run 2	75.00	**77.83**	73.60
Run 3	74.73	74.03	74.94
Average	74.71	**75.60**	72.69

with the intuition that the more complex a model, the better the performance. However, this is not the general case as in conclusion after discussing all the four metrics used to measure.

The recall fluctuates with the lower number of states reporting better values. Indeed, the highest value across the runs is depicted by the third run for the 4-state-based HMM. The average values across the runs also depict the overall pattern of performance.

In terms of F1-score, the highest across all runs is 5-state-based HMM with an overall pattern on improvement in performance from 4 to 5 states. This is generally followed by a plateau or drop by the 6 states. The average shows this pattern more clearly. Finally, we choose the HMM structure with the optimum performance, i.e., the five-state-based one of the second run.

3.3.2 Comparative Analysis

We also compare our proposed multinomial HMM with a neural network. The architecture is made up of 2 hidden layers each with a Rectified Linear Unit (ReLU) activation layer. We did not choose a deeper model to avoid overfitting. Moreover, we choose not to deploy a deep learning model on the evaluation dataset given its relatively small size.

Table 6 Performance of a neural network across 3 runs

Run	1	2	3	Average
Accuracy	54.47%	55.00%	53.68%	54.38%
Precision	87.00%	88.00%	85.00%	86.67%
Recall	10.53%	11.58%	8.94%	10.35%
F1-score	18.78%	20.47%	16.19%	18.48%

The learning rate we utilized was 0.001 with a stochastic gradient descent optimizer. This was applied for 10 training steps. Moreover, we employed a batch size of 256. The performance results of the comparative models may be observed in Table 6.

4 Conclusion

In conclusion, this chapter investigated a setup that allows us to successfully deploy HMMs on images. In particular, we develop a system for fire detection, a dynamic texture recognition problem. We applied model selection as well as ran multiple runs for testing. We have superior results of 76% accuracy and 78% F1-score ($K = 5$). This is in comparison to the neural networks which we trained on the same data. It has a testing best accuracy of 50% and an F1-score of 20.47% (run three times). We did not choose a deep learning model given the limited available data which is actually one of the challenges of fire detection in images (visible spectrum) and the availability of a benchmark dataset for it. Our proposed model also may be considered a real-time one, given that the testing time for each image is around 0.02 s. Future works may include the fusion of multispectral image data as well as relevant feature engineering.

References

1. L. Carlson, D. Navalta, M. Nicolescu, M. Nicolescu, G. Woodward, Multinomial hmms for intent recognition in maritime domains, in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (2019), pp. 1856–1858
2. H.-S. Le, H. Li, Simple 1d discrete hidden Markov models for face recognition, in *International Workshop on Visual Content Processing and Representation* (Springer, 2003), pp. 41–49
3. D. DeMenthon, M. Vuilleumier, D. Doermann, Hidden Markov models for images, in *Int. Conf. on Pattern Recognition, Barcelona, Spain* (Citeseer, 2000)
4. M. Mouret, C. Solnon, C. Wolf, Classification of images based on hidden Markov models, in *2009 Seventh International Workshop on Content-Based Multimedia Indexing* (IEEE, 2009), pp. 169–174
5. M. Qiao, W. Bian, R.Y. Da Xu, D. Tao, Diversified hidden Markov models for sequential labeling. IEEE Trans. Knowl. Data Eng. **27**(11), 2947–2960 (2015)
6. G. Doretto, A. Chiuso, Y.N. Wu, S. Soatto, Dynamic textures. Int. J. Comput. Vis. **51**(2), 91–109 (2003)

7. D. Chetverikov, R. Péteri, A brief survey of dynamic texture description and recognition. in *Computer Recognition Systems* (Springer, 2005), pp. 17–26
8. W. Fan, N. Bouguila, Online video textures generation, in *Advances in Visual Computing, 5th International Symposium, ISVC 2009, Las Vegas, NV, USA, November 30–December 2, 2009, Proceedings, Part II*, ser. Lecture Notes in Computer Science, vol. 5876, ed. by G. Bebis, R.D. Boyle, B. Parvin, D. Koracin, Y. Kuno, J. Wang, R. Pajarola, P. Lindstrom, A. Hinkenjann, L.M. Encarnação, C.T. Silva, D.S. Coming (Springer, 2009), pp. 450–459. [Online]. Available: https://doi.org/10.1007/978-3-642-10520-3_42
9. W. Fan, N. Bouguila, Dynamic textures clustering using a hierarchical pitman-yor process mixture of dirichlet distributions, in *2015 IEEE International Conference on Image Processing, ICIP 2015, Quebec City, QC, Canada, September 27–30, 2015* (IEEE, 2015), pp. 296–300. [Online]. Available: https://doi.org/10.1109/ICIP.2015.7350807
10. W. Fan, N. Bouguila, Generating video textures by PPCA and gaussian process dynamical model, in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, 14th Iberoamerican Conference on Pattern Recognition, CIARP 2009, Guadalajara, Jalisco, Mexico, November 15–18, 2009. Proceedings*, ser. Lecture Notes in Computer Science, vol. 5856, ed. by E. Bayro-Corrochano, J. Eklundh (Springer, 2009), pp. 801–808. [Online]. Available: https://doi.org/10.1007/978-3-642-10268-4_94
11. W. Fan, N. Bouguila, Novel approaches for synthesizing video textures. Expert Syst. Appl. **39**(1), 828–839 (2012). [Online]. Available: https://doi.org/10.1016/j.eswa.2011.07.081
12. S. Ali, 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 2019, Ottawa, ON, Canada, November 11–14, 2019* (IEEE, 2019), pp. 1–5. [Online]. Available: https://doi.org/10.1109/GlobalSIP45357.2019.8969450
13. Y. Qiao, L. Weng, Hidden Markov model based dynamic texture classification. IEEE Signal Process. Lett. **22**(4), 509–512 (2014)
14. R. Xu, H. Lin, K. Lu, L. Cao, Y. Liu, A forest fire detection system based on ensemble learning. Forests **12**(2), (2021). [Online]. Available: https://www.mdpi.com/1999-4907/12/2/217
15. B. Kim, J. Lee, A video-based fire detection using deep learning models. Applied Sciences **9**(14), 2862 (2019)
16. H. Pan, D. Badawi, A.E. Cetin, Computationally efficient wildfire detection method using a deep convolutional network pruned via fourier analysis. Sensors **20**(10), 2891 (2020)
17. F. Yuan, L. Zhang, X. Xia, B. Wan, Q. Huang, X. Li, Deep smoke segmentation (2018)
18. J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015)
19. T.-H. Chen, P.-H. Wu, Y.-C. Chiou, An early fire-detection method based on image processing, in *2004 International Conference on Image Processing, 2004. ICIP'04*, vol. 3 (IEEE, 2004), pp. 1707–1710
20. D.Y. Chino, L.P. Avalhais, J.F. Rodrigues, A.J. Traina, Bowfire: detection of fire in still images by integrating pixel color and texture analysis, in *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images* (IEEE, 2015), pp. 95–102
21. Z. Teng, J.-H. Kim, D.-J. Kang, Fire detection based on hidden Markov models. Int. J. Control Automat. Syst. **8**(4), 822–830 (2010)
22. L. Wang, M. Ye, J. Ding, Y. Zhu, Hybrid fire detection using hidden Markov model and luminance map. Comput. Electr. Eng. **37**(6), 905–915 (2011)
23. L. Rabiner, B. Juang, An introduction to hidden Markov models. IEEE ASSP Mag. **3**(1), 4–16 (1986)
24. J. Bilmes, A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Int. Comput. Sci. Inst. **4**, 126 (1998)
25. J.A. Hartigan, *Clustering Algorithms*, 99th edn. (Wiley, New York, NY, USA, 1975)

26. S. Ali, N. Bouguila, Hybrid generative-discriminative generalized dirichlet-based hidden Markov models with support vector machines, in *2019 IEEE International Symposium on Multimedia (ISM)* (2019), pp. 231–2311
27. S. Ali, N. Bouguila, Maximum a posteriori approximation of hidden Markov models for proportional sequential data modeling with simultaneous feature selection. IEEE Trans. Neural Netw. Learn. Syst., 1–12 (2021)

Hidden Markov Models: Discrete Feature Selection in Activity Recognition

Samr Ali and Nizar Bouguila

1 Introduction

Smart cities are now closer to reality than ever [1]. Several efforts are carried out across the areas of urban planning [2], energy [3], and buildings [4] among others to achieve this futuristic vision. It is then prevalent that we investigate the various applications of this paradigm. In particular, we focus on indoor activity recognition using Internet of Things (IoT) sensors.

While IoT technology is becoming more prevalent, they may be utilized for various applications [5–7]. Activity recognition is one such interesting application that addresses the classification of the current activity that is carried out by occupants of a particular environment space. Formally, this is a complex task given the various challenges that are inherent to its definition.

In particular, activities may be interleaved (beginning another activity as you complete another), concurrent (carrying out more than one activity at the same time), and open to various interpretations (subjective in nature and dependent on context) [8]. Furthermore, multiple residents further complicate the problem [9]. Another significant aspect to be considered in successfully studying this application is the nature of the sensors to be considered as the data to be modelled. For instance, these may be ambient, wearable, object, or mobile sensors [10].

S. Ali
Concordia University, Montreal, QC, Canada

Global Artificial Intelligence Accelerator (GAIA), Ericsson, Montreal, QC, Canada
e-mail: al_samr@encs.concordia.ca; samr.ali@ericsson.com

N. Bouguila (✉)
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca

N. Bouguila et al. (eds.), *Hidden Markov Models and Applications*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-99142-5_5

Each of these come with their own set of challenges and their appropriateness depends upon the application as well as the contextual environment. Intuitively, they may be used to complement each other. For thoroughness, we also mention that activity recognition may be carried out with other means of data such as the well-established vision-based methodologies [11, 12]. However, this falls outside the scope of this chapter and also suffers from a privacy constraint.

While such a topic is increasingly investigated, a benchmark for its evaluation is necessary and has been identified as the OPPORTUNITY dataset such as in [13]. Indeed, the focus of the aforementioned paper is to present the performance of various machine learning methods on this dataset. Specifically, the k-nearest neighbour (KNN), nearest centroid classifier, linear discriminant analysis, and quadratic discriminant analysis are compared whereby the KNN outperforms the other models.

Nonetheless, in this chapter, we investigate the performance of the hidden Markov models (HMMs) in solving this problem. HMMs are state-space generative models that are capable of capturing intricate underlying patterns in sequential data [14–23]. Sequential data refers to collected data where order is important. When this order is time-based, this data is now referred to as time series data. This is the case for sensor-collected data across time.

Furthermore, given the generative nature of the model, it is less prone to overfitting as it learns the underlying pattern of the representative data rather than the separating decision lines between different classes. The latter is the behaviour of discriminative models. It is noteworthy to mention that by learning this conditional probability, the performance of the latter is usually higher than the generative models that learn the joint probability. Nonetheless, in a study of activity recognition with mobile data, the performance of HMMs was found to be comparable with support vector machines and multilayer perceptron [24]. In addition, it is imperative that the model does not overfit in activity recognition given the aforementioned challenges.

Utilization of HMMs for activity recognition is prevalent in the literature [25]. For instance, [8] shows how HMMs may be utilized for this problem as well as several other machine learning approaches. On the other hand, [26] utilized a two-layered HMM to capture the variability of the activities by using the first layer to model groups of activities of the dataset followed by the individual activity at the lower level. It was found that such a configuration is superior in its performance in comparison to other models (traditional HMM, naive Bayes, and conditional random field (CRF)). HMMs also outperform CRFs in classification of activities in [27].

We also investigate both filter- and wrapper-based feature selection techniques for discrete sequential data for HMM-based modelling. Filter-based feature selection methods refer to techniques that are carried out as a step in the preprocessing of the data [28]. On the other hand, wrapper-based feature selection algorithms rely on the exploration of the entire feature combination subspace [29–35]. The aim is to reach the subset with the largest amount of information discarding any redundant or unnecessary features.

The rest of this chapter is organized as follows: Sect. 2 explores the HMM and the feature selection approaches employed in our investigation. Next, Sect. 3 presents the experimental setup and analyses the results. Finally, we draw the concluding remarks in Sect. 4.

2 Hidden Markov Models and Feature Selection

In this section, we describe the model as well as feature selection methods that we utilize for our investigations. Section 2.1 presents the HMM and its parameters with a focus on discrete data. Next, Sect. 2.2 identifies the filter- and wrapper-based techniques that we employ for feature selection.

2.1 Hidden Markov Models

A HMM is formed by double stochastic processes such as one is hidden and the other is observed. In particular, K represents the hidden states that characterize the model with a probability π_i to start in a state i. These states form a Markov chain and traversing between a state and another or even itself is defined according to $B = [b_{ii'} = P(s_{t+1} = i'|s_t = i)]$, a transition matrix defining the probabilities between the current state s_t and the next one s_{t+1}.

As we focus on discrete HMMs for our investigations, we define a correspondingly discrete observation set $\Xi_{it}(m) = P(X_t = \xi_m|s_t = i)$ for an observation X_t. This is defined for $(i, t, m) \in [1, K]x[1, T]x[1, M]$ and $\xi = [xi_1, xi_2, \ldots, \xi_M]$. This may be observed in Fig. 1. However, it is noteworthy to mention that a HMM can just as easily be defined in the continuous case by varying the definition of the observable process to match the distribution at hand, conventionally defined by a Gaussian distribution or mixture model.

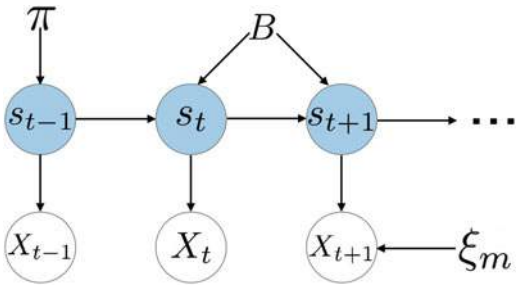


Fig. 1 Graphical representation of the multinomial hidden Markov model

2.2 Feature Selection Methods

The feature selection process that we incorporate for our investigations in this chapter is a preprocessing stage. That is, it is a separate entity from the model and carried out on the features before the training step. In this chapter, we address two types of such methods: filter-based in Sect. 2.2.1 and wrapper-based in Sect. 2.2.2.

2.2.1 Filter-Based Techniques

Filter-based techniques are founded on information criteria and other statistical measures for independence and correlation. They aim to 'filter' out redundant or unnecessary data/features. Famous ones include the Pearson's correlation [36] for numerical labels (regression) and linear discriminant analysis [37] for categorical labels (classification). Both of these are for numerical or continuous features. While there are several for continuous data, we focus on ones suitable for discrete data given our problem scope. In particular, we investigate the following methods:

1. **Chi^2**: It is a measure of independence that makes two inherent assumptions: (1) features are independent and (2) all expected frequencies are higher than 1 with no more than 20% of all cells less than 5 [38]. These conditions are satisfied by our data as the sensor data are collected independently of each other (though inherent relationships may exist but this is as close to satisfying this assumption as we can get). Moreover, the expected frequencies conditions are met. This measure is defined by the Chi^2:

$$\chi^2 = \sum \frac{o_{jl}^2 - e_{jl}^2}{e_{jl}} \tag{1}$$

   where o_{jl} is the observation value at j and l with j as the index for the rows/data instances in a total of J and l as the index for the columns/features in a total of L. e_{jl} represents the expected frequency.
2. **Mutual Information:** It is a measure of mutual dependence between the features [39]. It is defined on the range $[0, +\infty)$. It is 0 when the features are truly independent. It is based on the entropy measure $H(X)$:

$$H(X) = -\sum_{a \in A} P(X = a) \log P(X = a) \tag{2}$$

   where a is a feature/discrete level/sensor reading in the set of all features A. Hence, the mutual information $I(.)$ may now be defined by:

$$I(X, Y) = H(X) - H(X|Y) \tag{3}$$

   where Y represents the labels.

3. **Cramer's V:** It is based on the Chi^2 but is able to provide an idea of the degree of association between the features [40]. It may be viewed as a transformation of the χ^2 as follows:

$$V = \sqrt{\frac{\chi^2/n}{min(J, L) - 1}} \tag{4}$$

where $n = \sum_{j,l} n_{jl}$ represents the sample size. Its range is (0, 1] whereby 1 shows the strongest association and is a symmetrical measure.

Other metrics for discrete data are also available such as Kendall's Tau or Spearman's correlations [41]; however, these are only suitable for ordinal data. The latter refers to data where order is needed. Given the characteristics of the data that we use for activity recognition in Sect. 3.2, these are not applicable.

2.2.2 Wrapper-Based Techniques

Two famous methodologies in wrapper feature selection are forward stepwise and backward stepwise feature selection. In the prior, we begin by the smallest subset of features, i.e., 1 and train the model on each of these subsets. Next, we evaluate the model and increase the subset by 1 in every iteration choosing to move forward with the highest performing subset.

Backward stepwise feature selection is also referred to as backward elimination and is the technique of interest in this chapter in terms of wrapper methods. While both of the wrapper techniques discussed here are computationally expensive, the backward elimination is chosen given the results that will be discussed shortly in the following section. The approach begins by training the model on feature subsets of a number lower than the complete set by 1.

Next, the performance of the trained models is compared and the feature that does not exist in the highest performing model is removed in the next stage. This is consecutively repeated until the performance degrades in comparison to the benchmark, i.e., the model trained on the full feature set. This results in a feature subset that is on bar of or has a better performance than the benchmark.

3 Experimental Setup and Results

This section details the various aspects in relation to the experiments carried out for our proposed framework. We first present the experimental setup of the proposed investigation in Sect. 3.1. Next, the activity recognition datasets that are utilized are discussed in Sect. 3.2. The evaluation metrics applied for the performance of the model and its improvement are examined in Sect. 3.3. Finally, the analysis of the results is carried out in Sect. 3.4.

3.1 Experimental Setup

The experimental setup that we employ in our investigations in this chapter is based on training a model on the data of each activity, i.e., class, independently. The testing of a new observation/collected reading of sensors is based on finding the log likelihood by applying the forward algorithm for each of the models. The final label is assigned according to the maximum resultant value.

This is an automation-friendly setup as it easily allows the addition of new activities by simply incorporating the additional trained models within the framework. Moreover, such flexibility in the model further facilitates an improved online model deployment. This also aids in the robustness of the model as each HMM is trained on a single class allowing for a better modelling of the underlying distribution of the data and elegantly handles the data imbalance.

Furthermore, usually data-driven models as well as knowledge-based ones that utilize machine learning techniques under which HMMs fall suffer from scalability problems [10]. This setup alleviates this constraint. Additionally, it presents flexibility in the model to incorporate the variability in interpretation that characterizes this paradigm. In particular, so long as common labelled data have a single trained model on them, other definitions may be added as separate models. For instance, if all previously labelled data were of eating sandwiches, then new data became available for eating though of spaghetti then two different models may be trained on them incorporating both variabilities. Depending on the application, these may be potentially then pooled.

3.2 Data

In this chapter, we use the OPPORTUNITY Activity Recognition Dataset [42]. This dataset constitutes of various sensor-based (wearable, object, and ambient) interleaved and hierarchical naturalistic activities of 4 subjects. It is considered as the benchmark for various tasks such as activity recognition and automatic data segmentation. In addition to its public availability that enables future researchers access to compare to our investigations, it is indeed considered as the benchmarking activity recognition dataset in [13]. We especially focus on two available subsets on UCI Machine Learning Repository:[1] Ordonez A and Ordonez B.

These are collected from two users in their own homes for a total of 35 days of labelled real unsimulated data. Ordonez A is a 4-room house with 14 days of labelled data. The labels/activities constitute of: Leaving, Toileting, Showering, Sleeping, Breakfast, Lunch, Snack, Spare_Time/TV, and Grooming. In contrast, the Ordonez B data is collected in a 5-room house setting with 21 labelled days of the following activities: Leaving, Toileting, Showering, Sleeping, Breakfast, Lunch,

[1] https://archive.ics.uci.edu/ml/datasets/opportunity+activity+recognition.

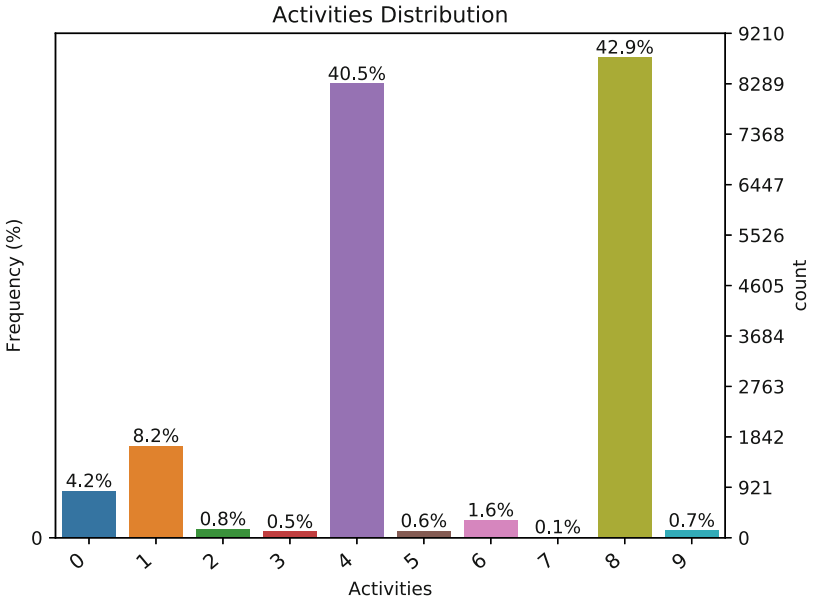


Fig. 2 The distribution and frequency of activities in Ordonez A dataset

Dinner, Snack, Spare_Time/TV, and Grooming. That is Ordonez B has the addition Dinner activity. Moreover, we add the Idle label for both datasets in the times when no activity is carried out.

The distribution of the data across the labels is not balanced; that is, we are modelling an imbalanced dataset. This is shown in Figs. 2 and 3 for Ordonez A and Ordonez B datasets, respectively. Our experimental setup allows us to tackle this hindrance in an elegant manner. This is due to the nature of splitting the labels for individual training of the model. Indeed, we no longer require an equal balance of the data across the labels due to this. In addition, HMMs are generative models, which learn the underlying representative distribution rather than a decision boundary between the various classes as in the case of discriminative models. This further solidifies the stability of the model in addition to its flexibility as discussed in Sect. 3.1.

In terms of the features that make up the datasets, they are based on collected data from binary sensors. The available data is in form of a log that we sample at a 1-minute quantum (sampling rate). This results in a total of 20,456 data instances for Ordonez A and 30,470 data instances for Ordonez B datasets.

The histograms for the distribution of the binary sensor values for the Ordonez A dataset are shown in Figs. 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15. On the other hand, Figs. 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, and 27 show the sensor value distributions for the Ordonez B dataset. There is a clear imbalance between the recorded values of the binary sensors; that is, 10 of the sensors were turned off most of the time in both of the datasets.

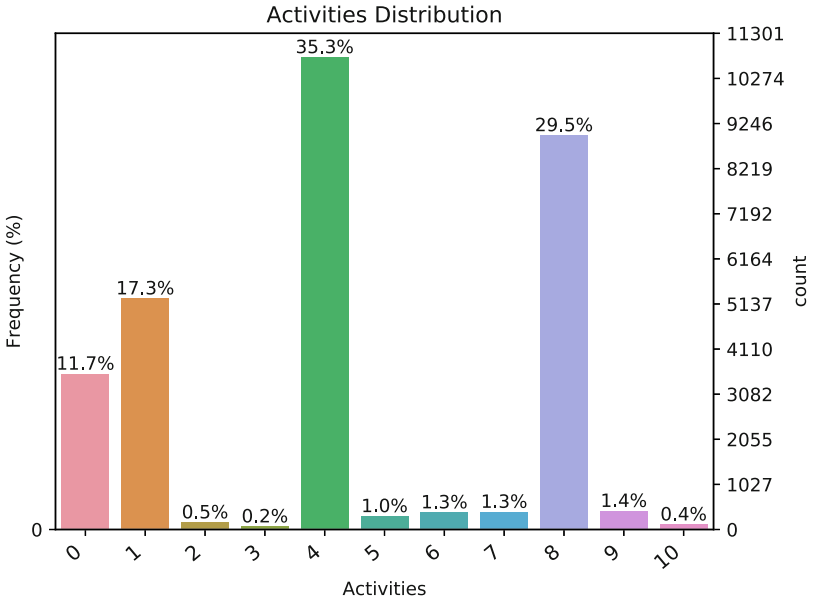


Fig. 3 The distribution and frequency of activities in Ordonez B dataset

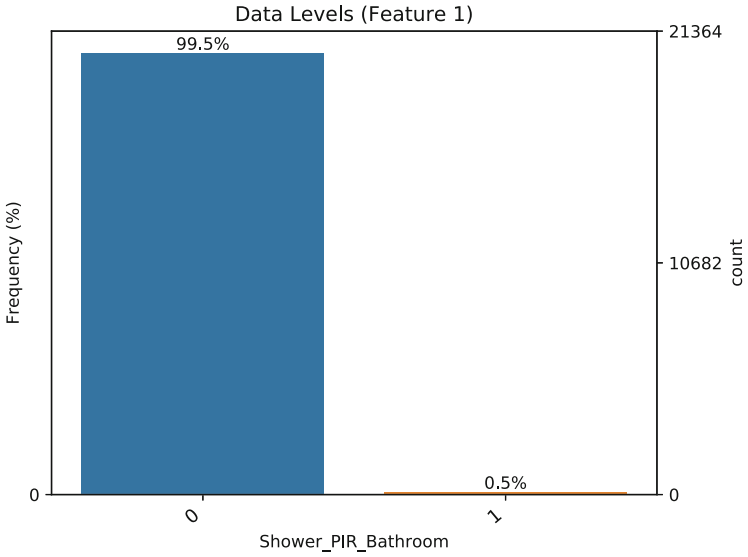


Fig. 4 Histogram of the values of the shower PIR sensor in bathroom in Ordonez A dataset (encoded feature 0 in backward elimination feature subsets)

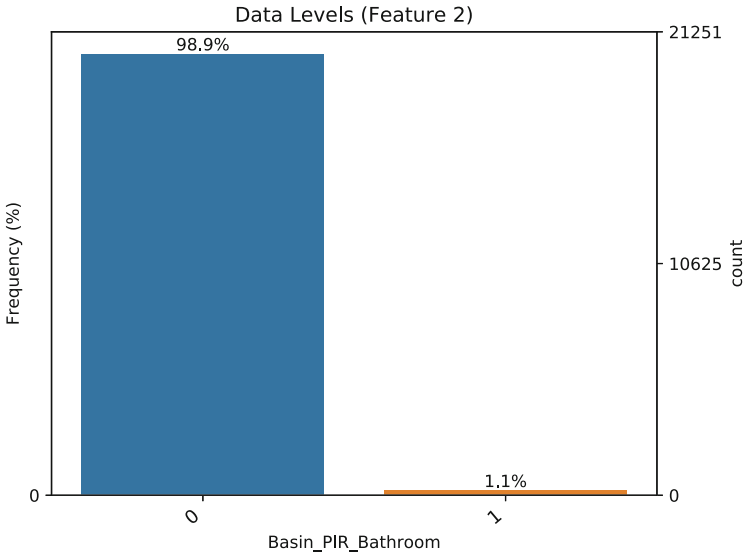


Fig. 5 Histogram of the values of the basin PIR sensor in bathroom in Ordonez A dataset (encoded feature 1 in backward elimination feature subsets)

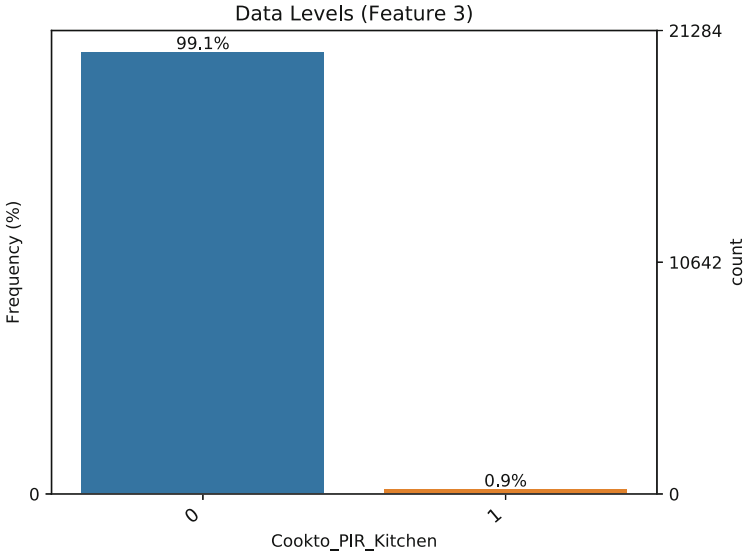


Fig. 6 Histogram of the values of the cooktop PIR sensor in kitchen in Ordonez A dataset (encoded feature 2 in backward elimination feature subsets)

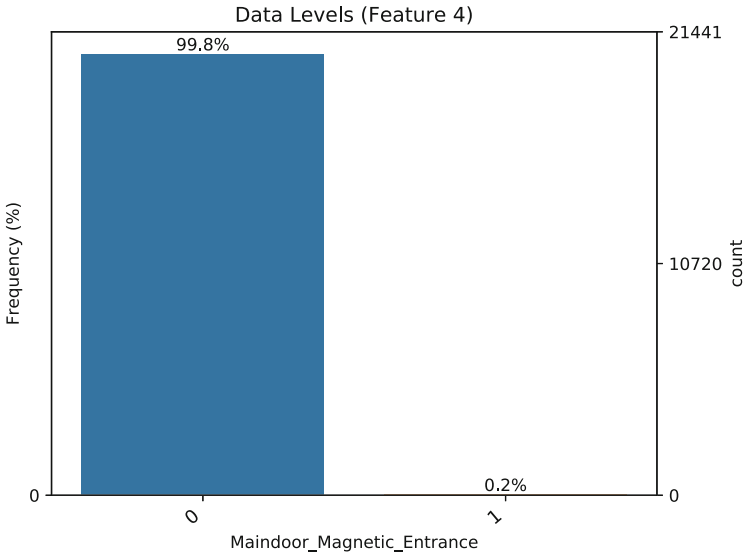


Fig. 7 Histogram of the values of the maindoor magnetic sensor in entrance in Ordonez A dataset (encoded feature 3 in backward elimination feature subsets)

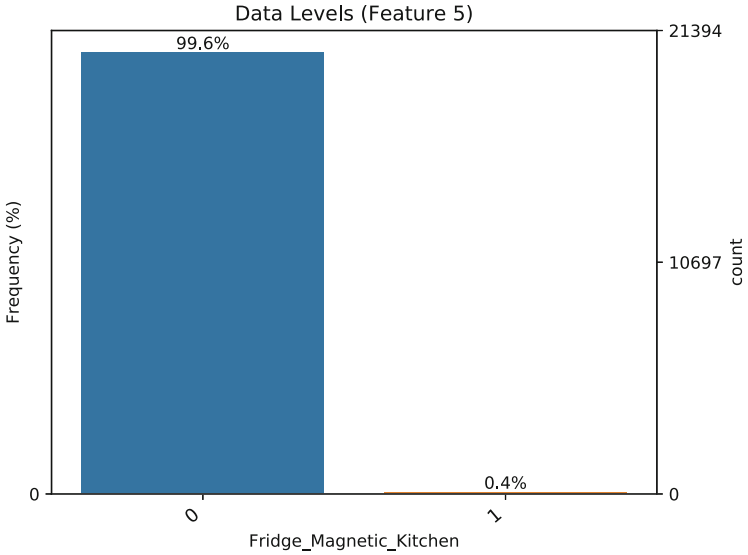


Fig. 8 Histogram of the values of the fridge magnetic sensor in kitchen in Ordonez A dataset (encoded feature 4 in backward elimination feature subsets)

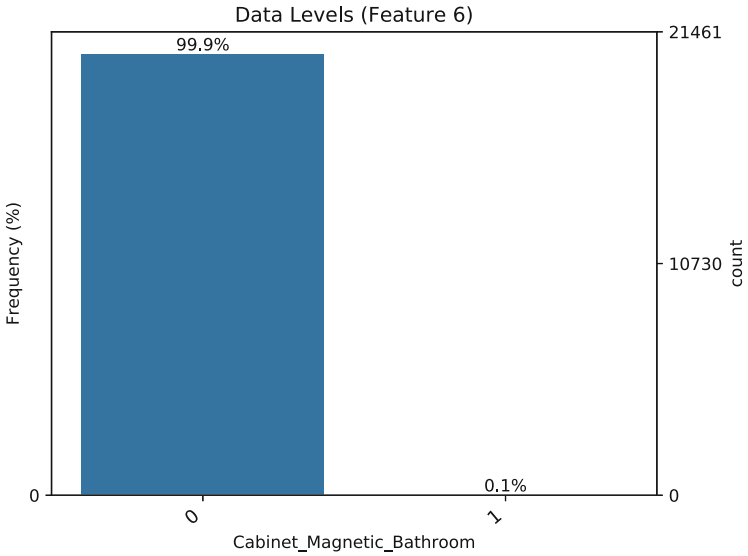


Fig. 9 Histogram of the values of the cabinet magnetic sensor in bathroom in Ordonez A dataset (encoded feature 5 in backward elimination feature subsets)

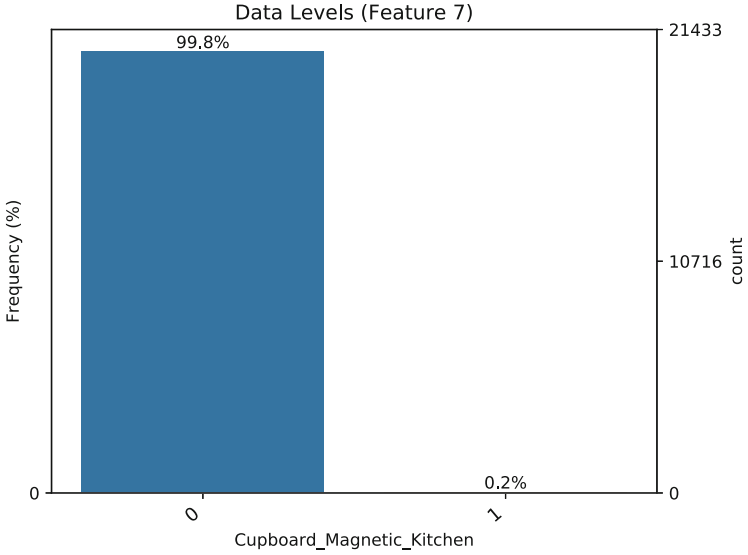


Fig. 10 Histogram of the values of the cupboard magnetic sensor in kitchen in Ordonez A dataset (encoded feature 6 in backward elimination feature subsets)

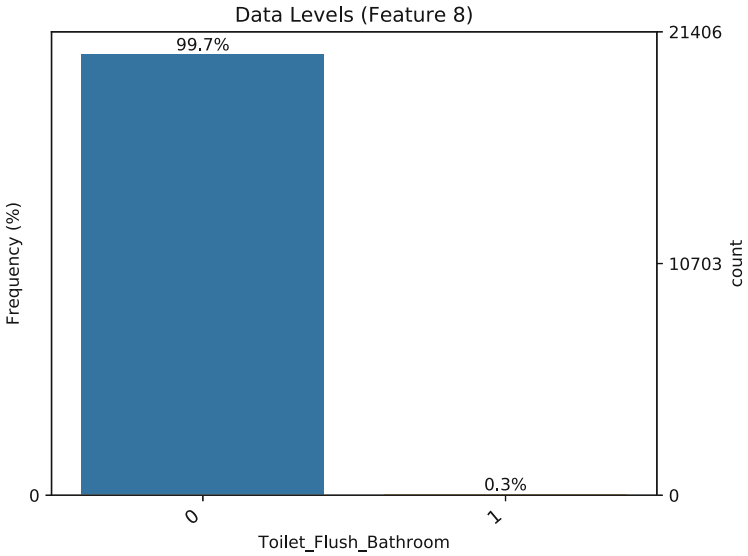


Fig. 11 Histogram of the values of the toilet flush sensor in bathroom in Ordonez A dataset (encoded feature 7 in backward elimination feature subsets)

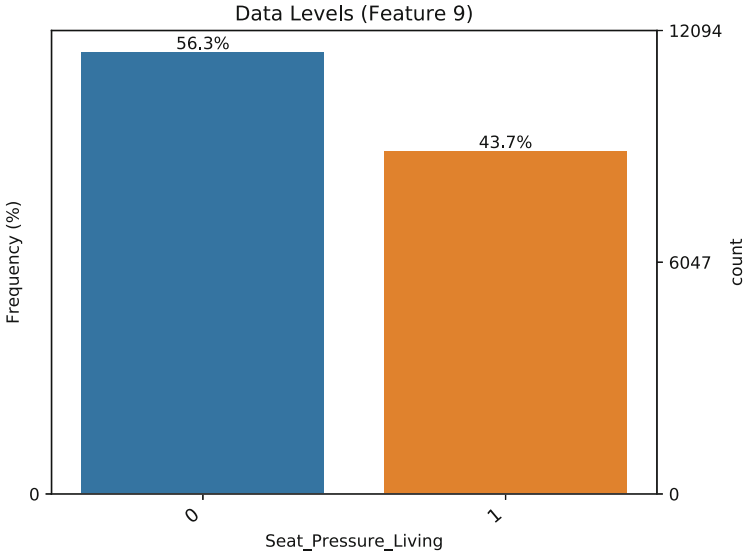


Fig. 12 Histogram of the values of the seat pressure sensor in living room in Ordonez A dataset (encoded feature 8 in backward elimination feature subsets)

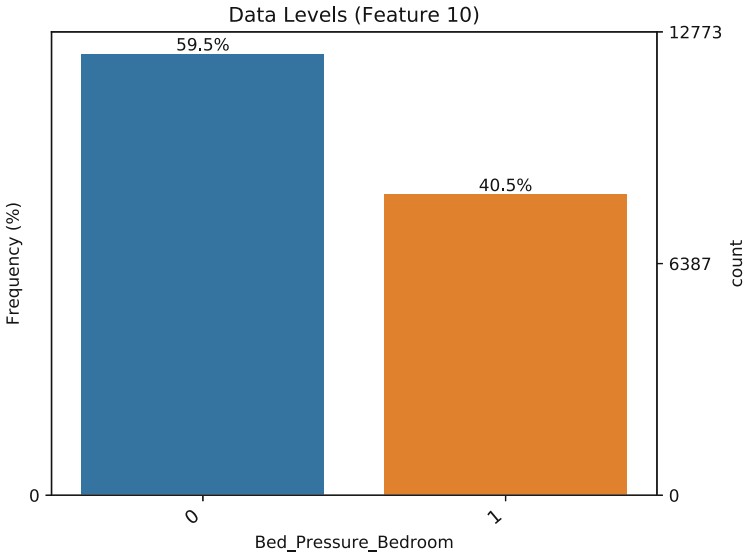


Fig. 13 Histogram of the values of the bed pressure sensor in bedroom in Ordonez A dataset (encoded feature 9 in backward elimination feature subsets)

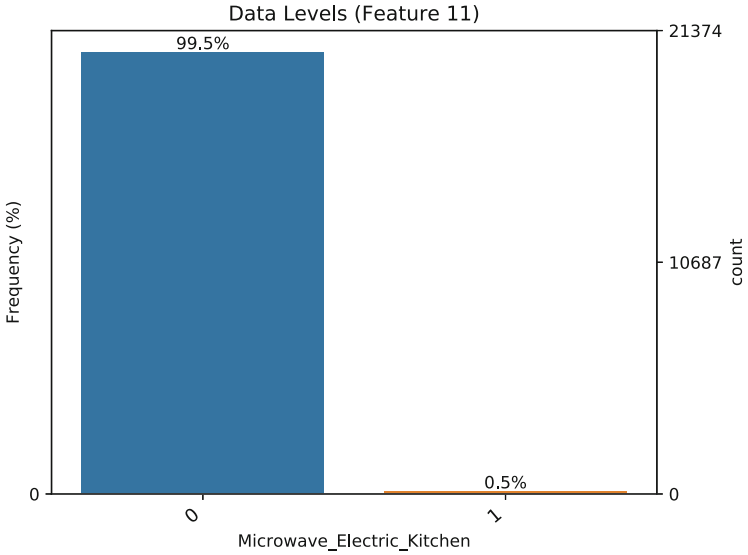


Fig. 14 Histogram of the values of the microwave electric sensor in kitchen in Ordonez A dataset (encoded feature 10 in backward elimination feature subsets)

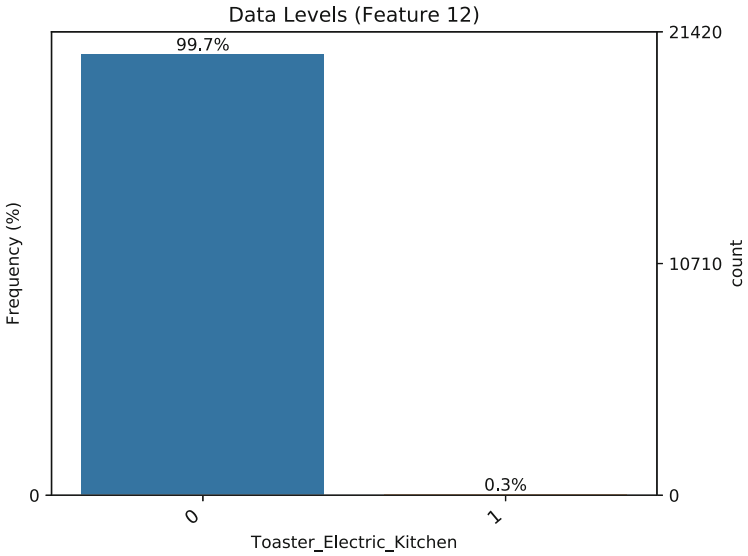


Fig. 15 Histogram of the values of the toaster electric sensor in kitchen in Ordonez A dataset (encoded feature 11 in backward elimination feature subsets)

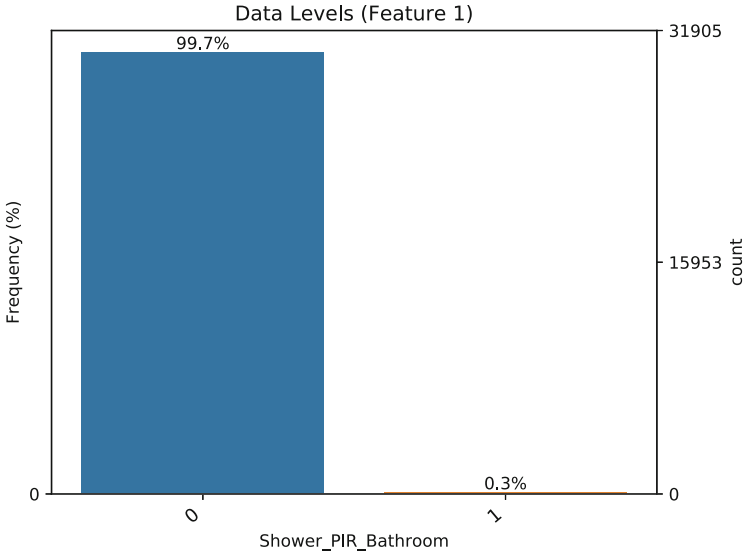


Fig. 16 Histogram of the values of the shower PIR sensor in bathroom in Ordonez B dataset (encoded feature 0 in backward elimination feature subsets)

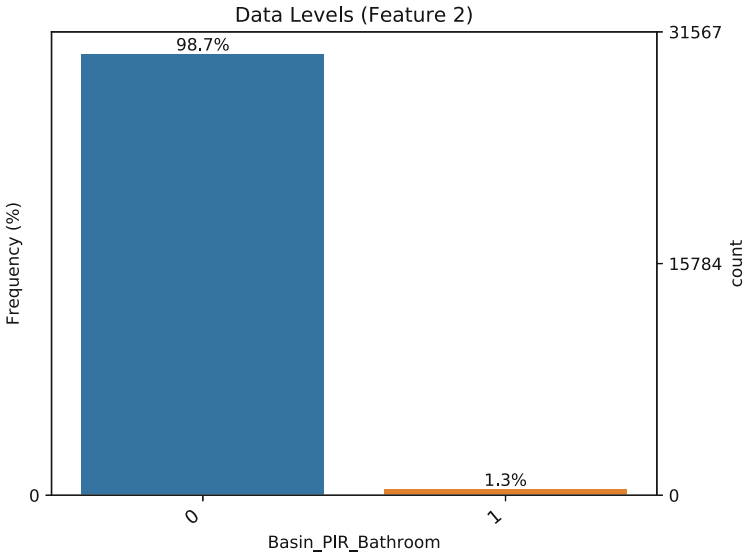


Fig. 17 Histogram of the values of the basin PIR sensor in bathroom in Ordonez B dataset (encoded feature 1 in backward elimination feature subsets)

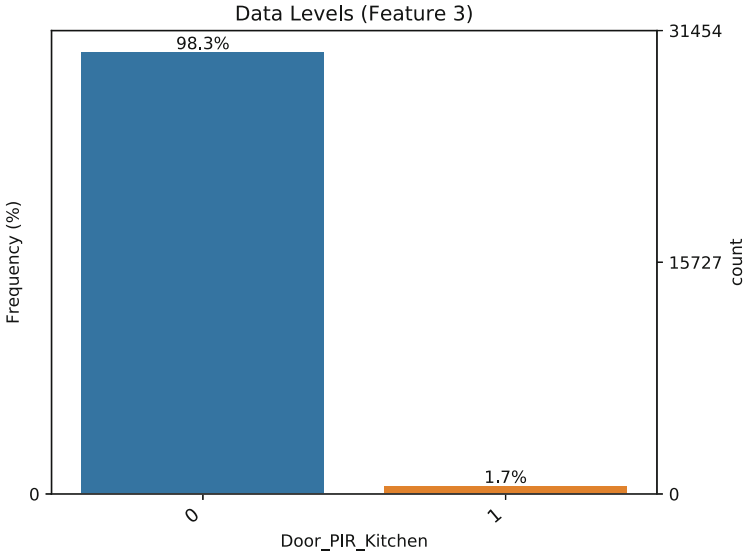


Fig. 18 Histogram of the values of the door PIR sensor in kitchen in Ordonez B dataset (encoded feature 2 in backward elimination feature subsets)

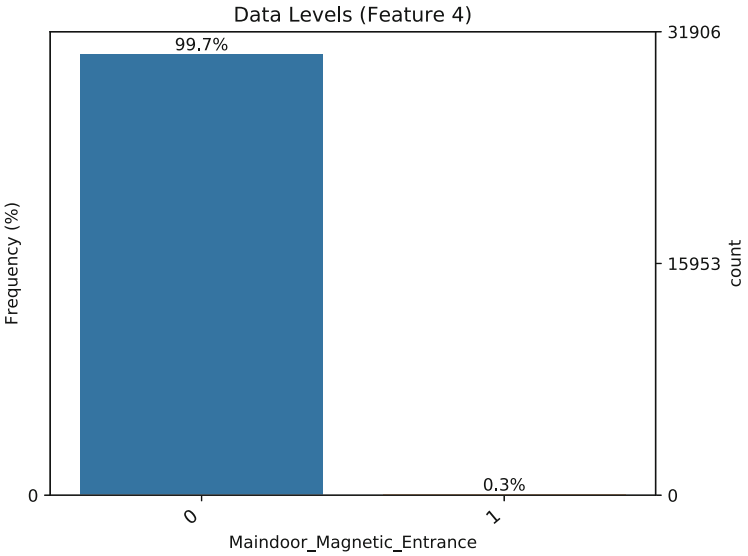


Fig. 19 Histogram of the values of the maindoor magnetic sensor in entrance in Ordonez B dataset (encoded feature 3 in backward elimination feature subsets).

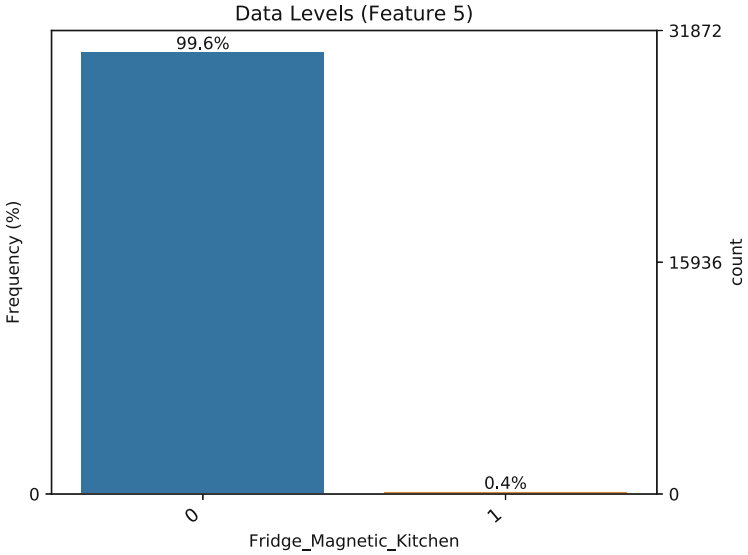


Fig. 20 Histogram of the values of the fridge magnetic sensor in kitchen in Ordonez B dataset (encoded feature 4 in backward elimination feature subsets)

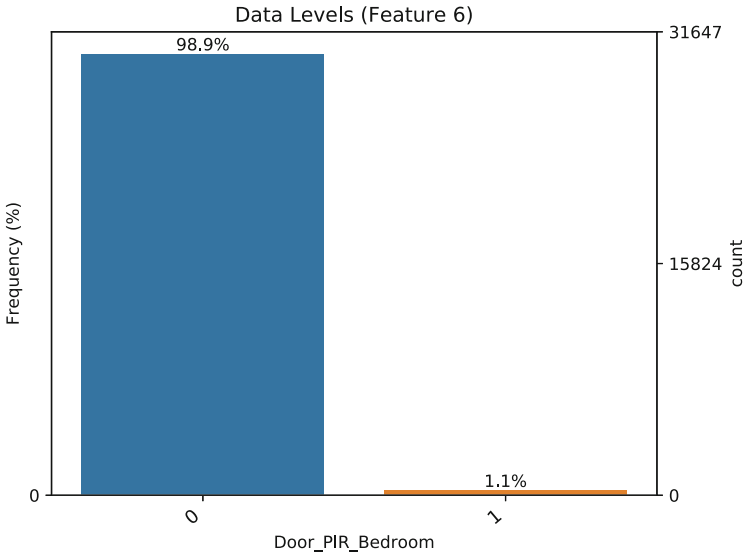


Fig. 21 Histogram of the values of the door PIR sensor in bedroom in Ordonez B dataset (encoded feature 5 in backward elimination feature subsets)

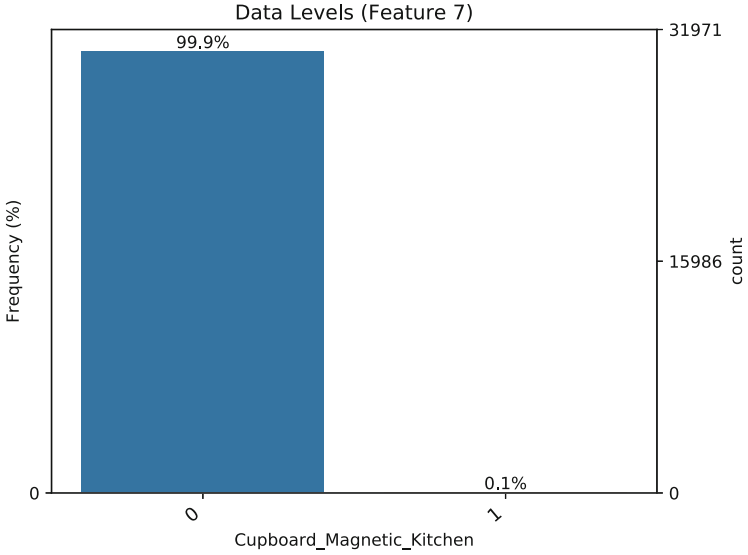


Fig. 22 Histogram of the values of the cupboard magnetic sensor in kitchen in Ordonez B dataset (encoded feature 6 in backward elimination feature subsets)

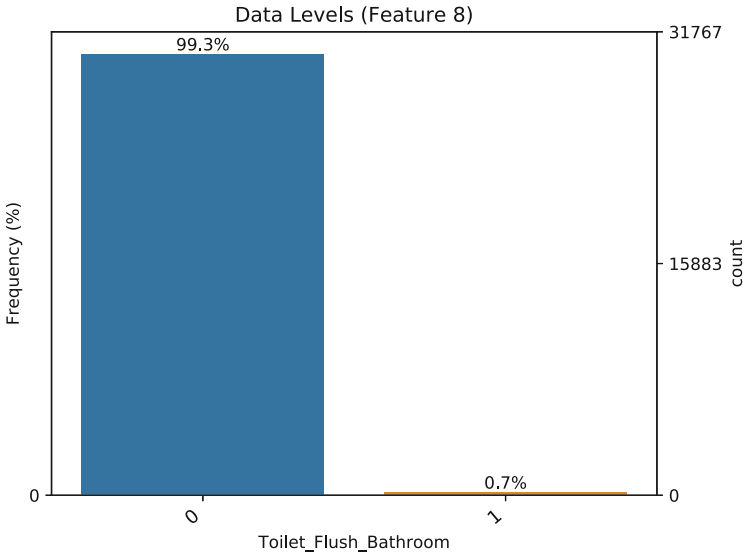


Fig. 23 Histogram of the values of the toilet flush sensor in bathroom in Ordonez B dataset (encoded feature 7 in backward elimination feature subsets)

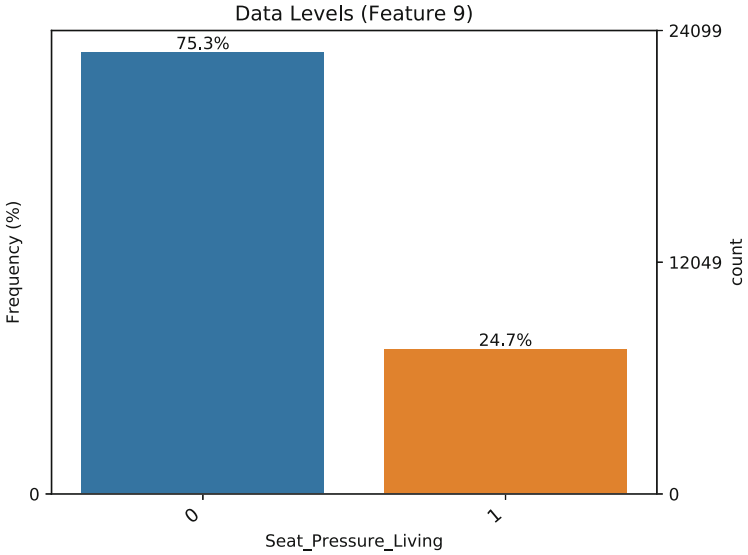


Fig. 24 Histogram of the values of the seat pressure sensor in living room in Ordonez B dataset (encoded feature 8 in backward elimination feature subsets)

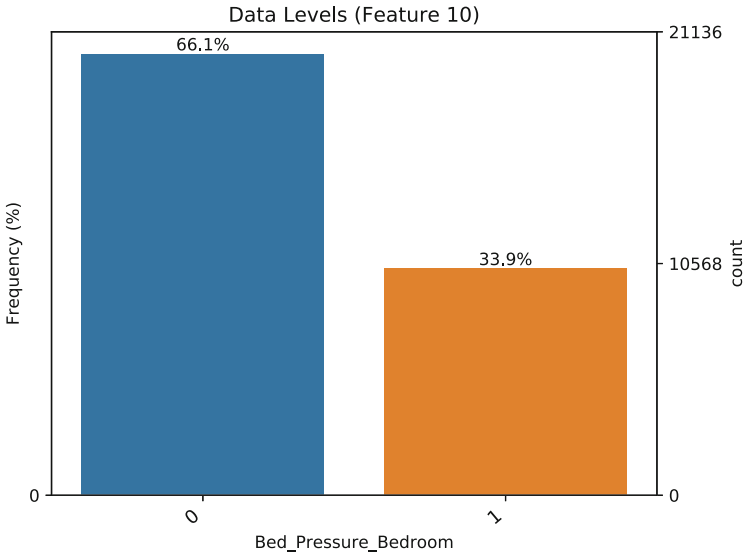


Fig. 25 Histogram of the values of the bed pressure sensor in bedroom in Ordonez B dataset (encoded feature 9 in backward elimination feature subsets)

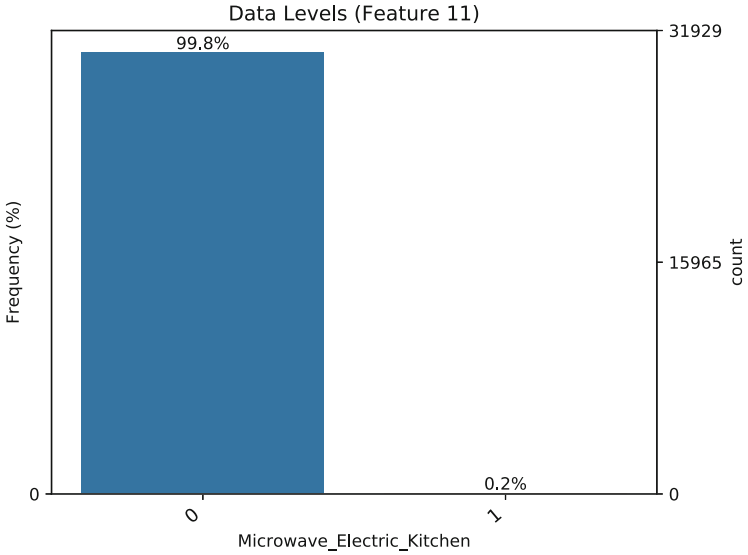


Fig. 26 Histogram of the values of the microwave electric sensor in kitchen in Ordonez B dataset (encoded feature 10 in backward elimination feature subsets)

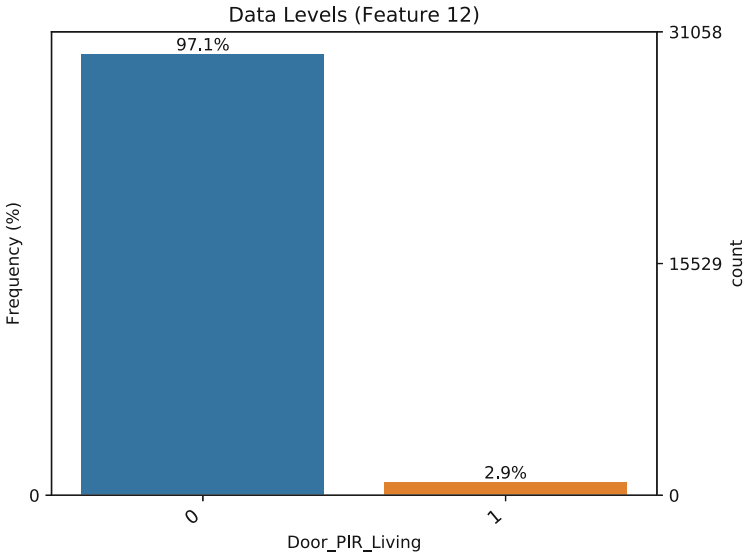


Fig. 27 Histogram of the values of the door PIR sensor in living room in Ordonez B dataset (encoded feature 11 in backward elimination feature subsets)

3.3 Evaluation Metrics

Accuracy may be defined as the measure that indicates the detection rate of the model. Mathematically, that is:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

where TP is the number of true positives, TN is the number of true negatives, and FP and FN represent the number of false positives and negatives, respectively. On the other hand, *precision* and *recall* allow us to measure the rate of true positives detection and the rate of missed positives, correspondingly. These may be defined by:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

The latter is particularly useful in imbalanced datasets. A balance of the two measures in terms of their harmonic mean is the *F-measure*. This is computed by:

$$\text{F-Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

Together, these form the set of reported evaluation metrics that we report for our investigations in the following subsection.

3.4 Results

In this section, we discuss and analyse our results. In particular, Sect. 3.4.1 presents the results for the filter-based discrete methods, whereas Sect. 3.4.2 details the results of the backward elimination technique. We also show the confusion matrix of the model trained on the chosen feature subset at each stage in the latter. We benchmark our feature selection investigations with the multinomial HMM trained on the complete feature set.

We also vary the number of states (K) in order to allow for more flexibility in the feature selection approaches and better model selection. While it is usually the case that model selection for probabilistic models, such as HMMs, follow an information criterion, this is not applicable in this chapter. Indeed, it was proven that in instances where the physical meaning of the labels is not found to correspond to the states, as is the case in our investigations, that the Akaike and the Bayesian information criteria are no longer appropriate [43].

It is also noteworthy to mention that we vary the training and testing data splits in order to investigate the best one for the training of the data and our investigations. We have experimentally set it to 70% training data with the remainder used for testing data. The resultant performance metrics are shown in Figs. 28 and 29 for the Ordonez A and the Ordonez B datasets, respectively.

The confusion matrix showing the benchmark results is displayed in Figs. 30 and 31 for Ordonez A and Ordonez B datasets, respectively.

3.4.1 Filter-Based Methods

In the filter-based feature selection techniques, we are basically following a best subset selection paradigm. Unfortunately, both the Chi^2 and Cramer's V methods were found to be inapplicable. In the case of the Ordonez A dataset, there was no sufficient data for a representative split after applying the Chi^2 method and little association was found with the Cramer's V measure. This is logical because the latter is based on the prior. For the Ordonez B dataset, considerable degrade was observed after applying the Chi^2 method even with 11 features and the Cramer's V measure's results were also showing little association.

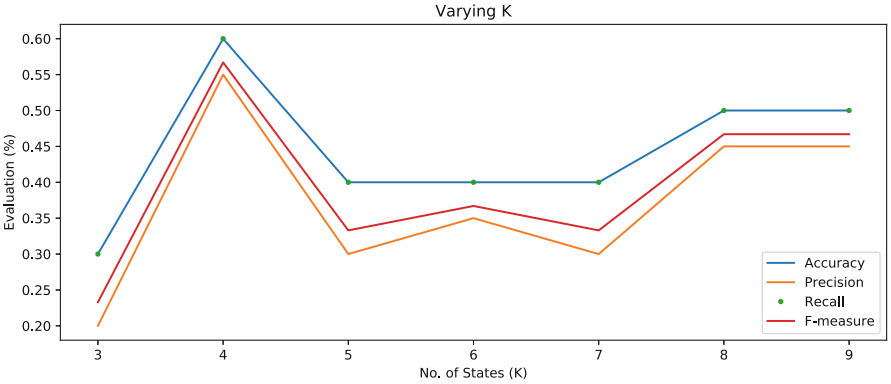


Fig. 28 Performance metrics across increasing number of states for the multinomial HMM trained on the complete feature set of Ordonez A dataset; K is ranging from 3 to 9

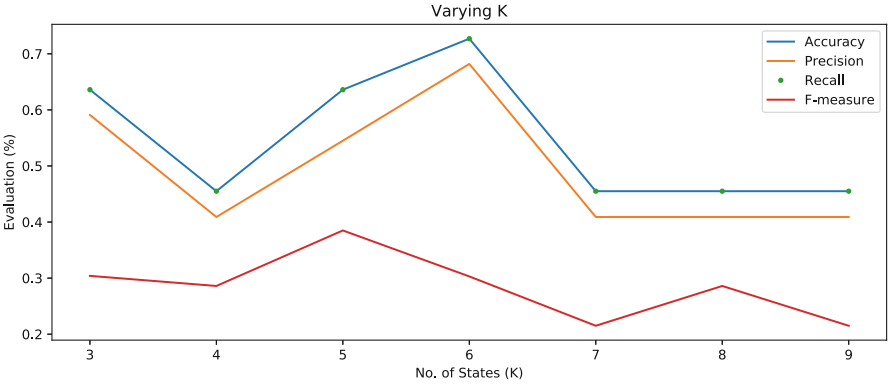


Fig. 29 Performance metrics across increasing number of states for the multinomial HMM trained on the complete feature set of Ordonez A dataset; K is ranging from 3 to 9

On the other hand, the mutual information approach performed slightly better. On the Ordonez A dataset, no significant improvement was found as shown in Fig. 32 with 11 features, except of course for the reduced number of features. Otherwise, it was found inapplicable on smaller subsets. This was similar for the Ordonez B dataset though it ran for an even smaller subset of 10 features. The confusion matrices for model trained with the resultant 11 and 10 features selected are, respectively, illustrated in Figs. 33 and 34.

3.4.2 Backward Elimination

The binary nature of the data and its low number of features motivate the utilization of the backward elimination in comparison to the forward stepwise feature selection

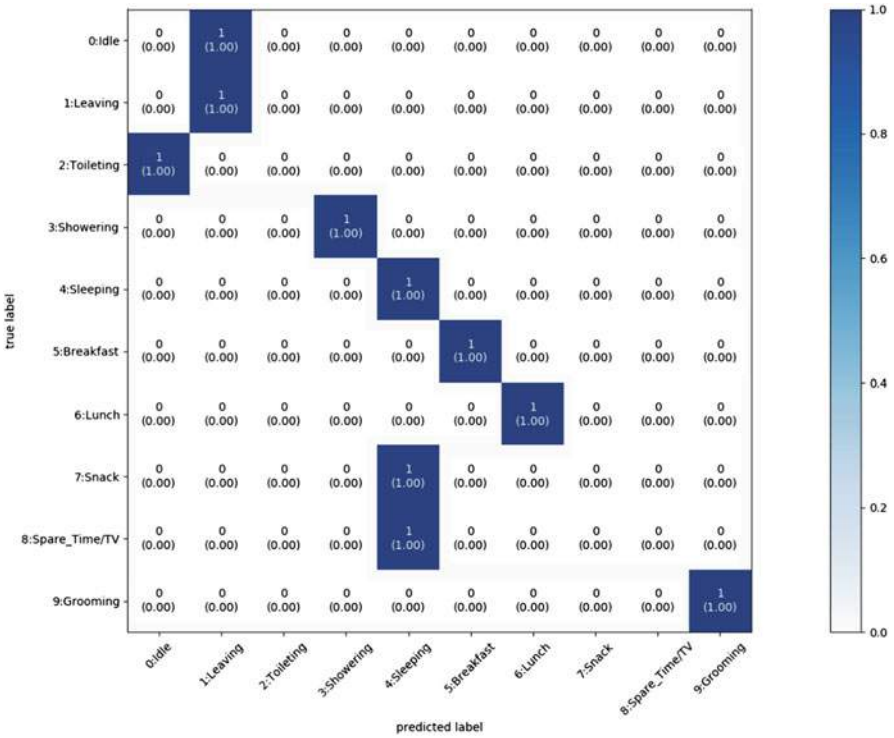


Fig. 30 Confusion matrix of the optimum performing benchmarking model ($K = 4$) trained on the complete feature set for Ordonez A dataset

technique. Additionally, the relatively poor performance of the filter-based methods as discussed further enforces the choice of the algorithm. This is due to the inability of the model to train properly on smaller feature subsets. In this section, we further explore the capabilities of this wrapper technique on the utilized two datasets and discuss our findings. It is noteworthy to mention that the performance metrics are zeroed out for the combinations where the models fail to train.

At 11 features of the Ordonez A dataset, the first feature subset combination to be generated in the process, we find out that the optimum performance is depicted by the model trained with $K = 3$ on feature subset 4: [0 1 2 3 4 5 6 7 9 10 11]. This is shown for the accuracy, precision, recall, and F-measure in Figs. 35, 36, 37, and 38, respectively. These are recorded as 60.00, 50.00, 60.00, and 53.33%, correspondingly. The confusion matrix of this optimized model can be seen in Fig. 39. Given this result, we use these features in the next round to create combinations of smaller size (10 features).

However, these are lower than the performance of the benchmark model across the board. This leads us to terminate the procedure for the Ordonez A dataset. It is

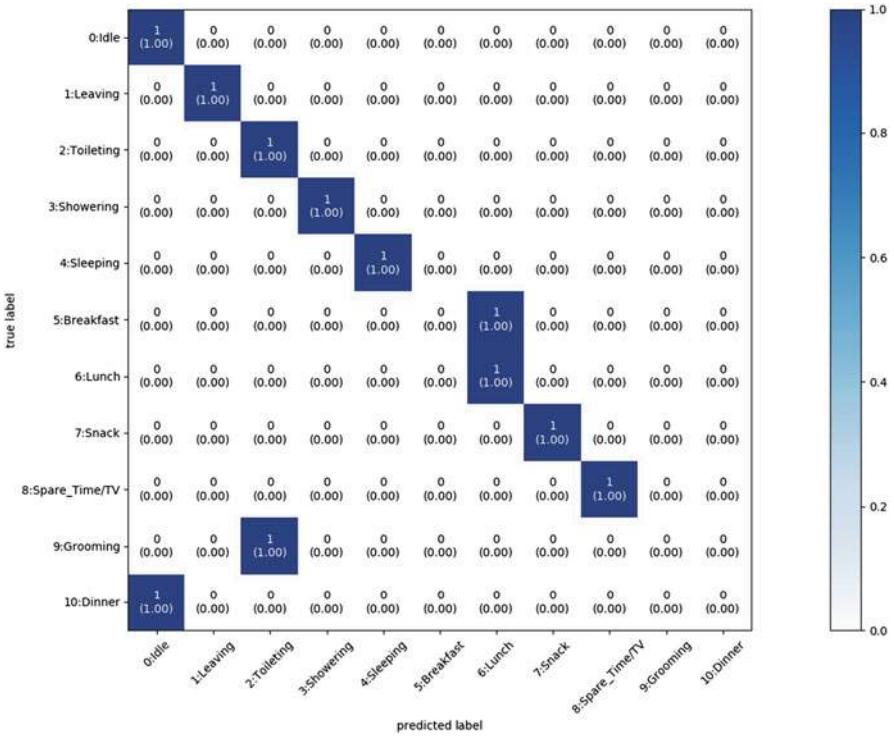


Fig. 31 Confusion matrix of the optimum performing benchmarking model ($K = 6$) trained on the complete feature set for Ordonez B dataset

noteworthy to mention that at this level, the model that was trained with optimum set of features had a lower complexity of 1 less state than the benchmark.

On the other hand, graphs depicting the results over varying number of states for 11 features selected for the Ordonez B dataset are shown in Figs. 40, 41, 42, and 43 for the accuracy, precision, recall, and F-measure metrics, respectively. The best configuration to move forward in this particular case would be feature subset 3 or 4 with a lower complexity of 3 states instead of the benchmark of $K = 6$ with the performance metrics maintained at the same level. The confusion matrices for these models are shown in Figs. 44 and 45.

From these results, we can conclude that the features encoded 8 (seat pressure sensor in the living room) and 9 (bed pressure sensor in the bedroom) are interchangeable. Upon inspection of their distribution across the two labels as shown in Figs. 24 and 25, we decide to remove feature 8 in the next backward elimination step. This results in an update in the encoding of the features as well to [f1, f2, f3, f4, f5, f6, f7, f8, f10, f11, f12] where the count starts at 0.

In the next step of the backward elimination, we further optimize the model not only by reducing the feature subset to 10 features but also find out a lower

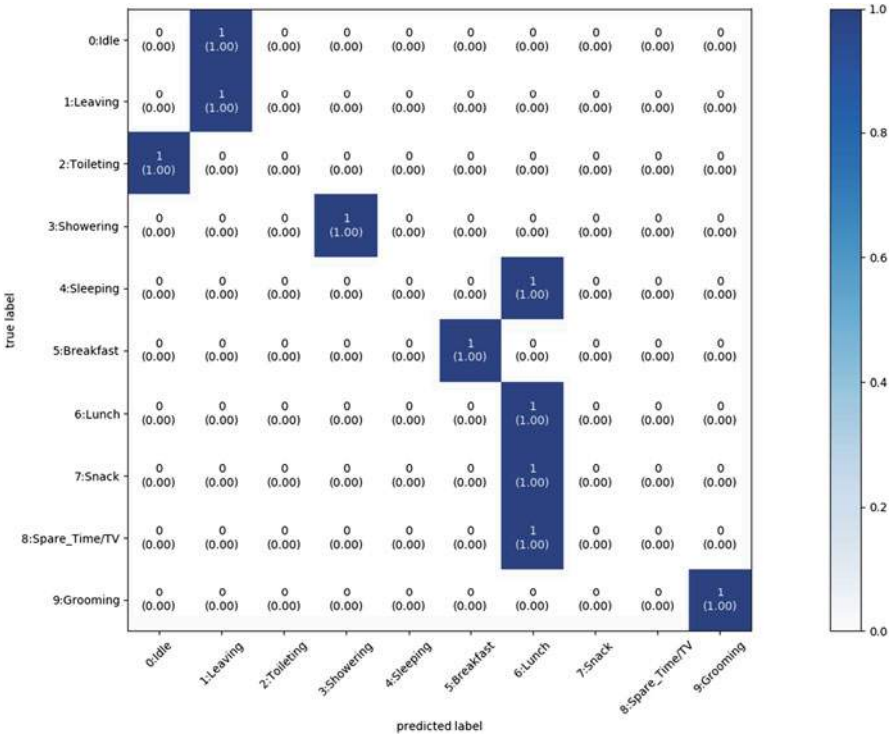


Fig. 32 Confusion matrix of the model trained on the mutual information selected set of 11 features for Ordonez A dataset

complexity model to have the optimum performance. In particular, the optimum feature subset is 6: [0 1 2 3 4 6 7 8 9 10] as shown in Figs. 46, 47, 48, and 49 for the accuracy, precision, recall, and F-measure metrics, respectively, with the confusion matrix illustrated in Fig. 50. This leads us to drop the encoded feature 5. This results in an update in the encoding of the features as well to [f1, f2, f3, f4, f5, f7, f8, f10, f11, f12] where the count starts at 0.

For 9 feature subsets, Figs. 51, 52, 53, and 54 show the accuracy, precision, recall, and F-measure metrics respectively with the confusion matrix illustrated in Fig. 55. The precision now improves in this round of feature selection in comparison to other higher numbered feature subsets and consequently the F-measure follows. That comes at the expense of higher complexity of the model (1 more state larger than the previous iteration). Nonetheless, this remains lower than the benchmark. The highest performing subset excludes the encoded feature 4. In the next iteration, we now encode features [f1, f2, f3, f4, f7, f8, f10, f11, f12] where the count starts at 0.

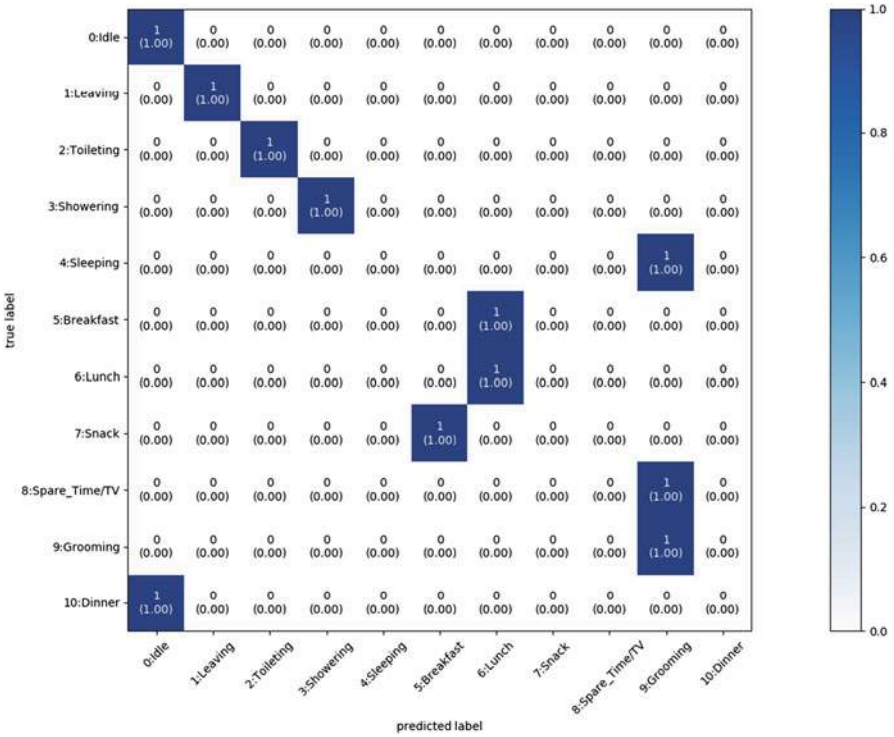


Fig. 33 Confusion matrix of the model trained on the mutual information selected set of 11 features for Ordonez B dataset

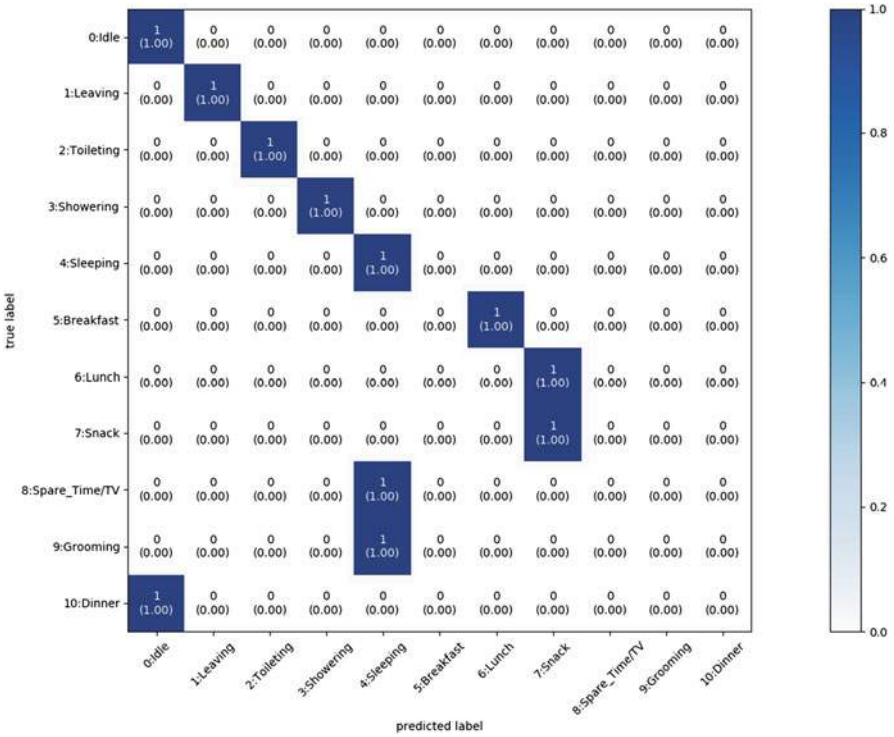


Fig. 34 Confusion matrix of the model trained on the mutual information selected set of 10 features for Ordonez B dataset

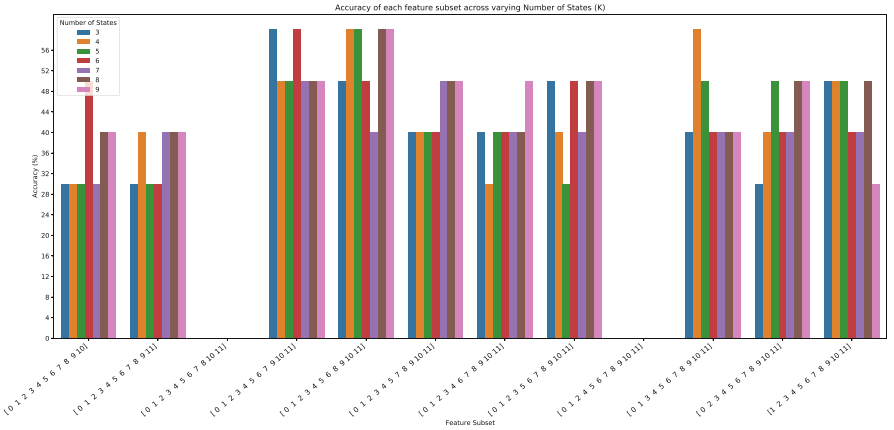


Fig. 35 Accuracy of the backward elimination selected 11 features across trained HMMs with various number of states for Ordonez A dataset

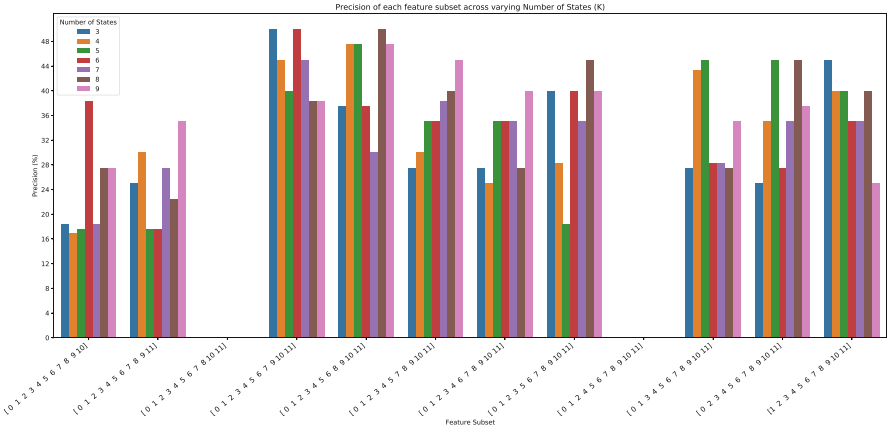


Fig. 36 Precision of the backward elimination selected 11 features across trained HMMs with various number of states for Ordonez A dataset

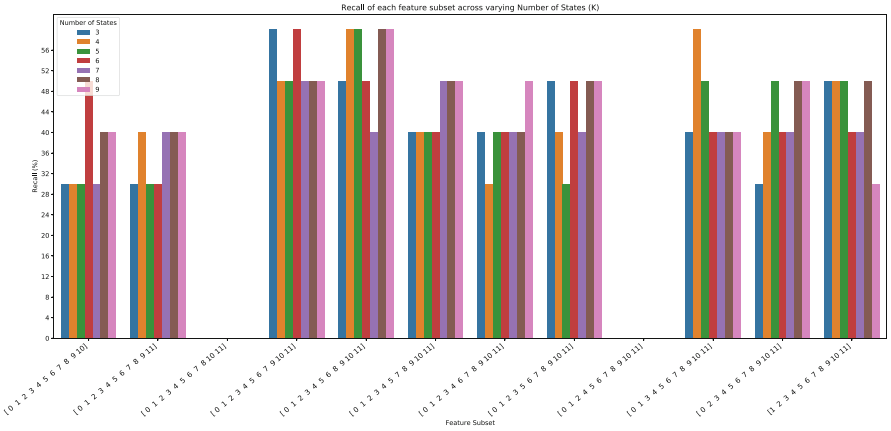


Fig. 37 Recall of the backward elimination selected 11 features across trained HMMs with various number of states

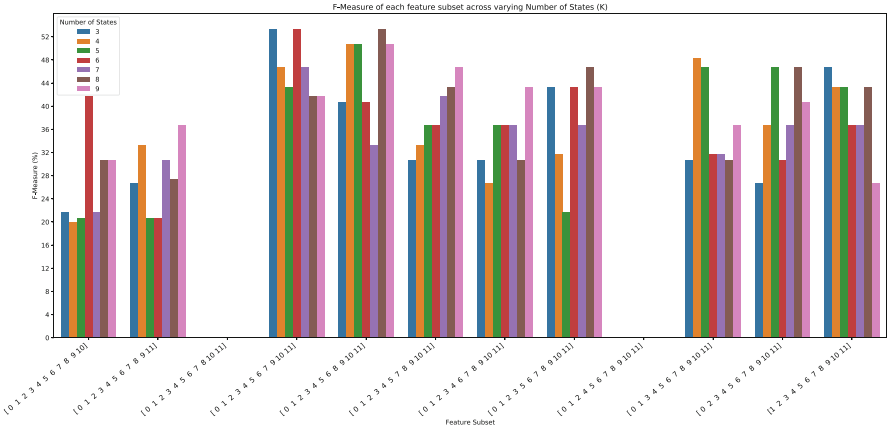


Fig. 38 F-measure of the backward elimination selected 11 features across trained HMMs with various number of states for Ordonez A dataset

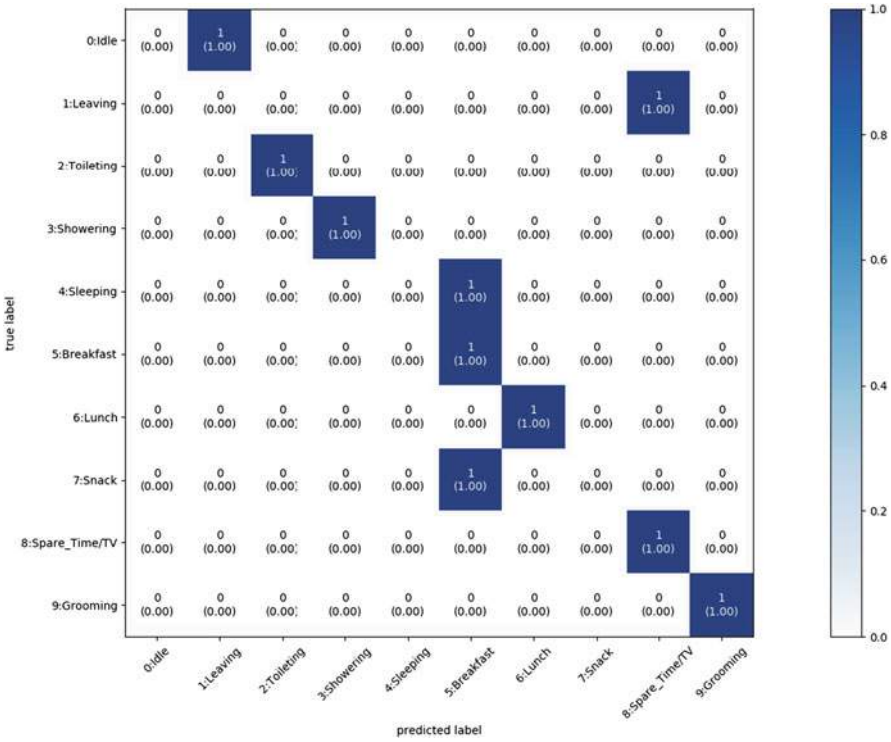


Fig. 39 Confusion matrix of the model trained with backward elimination selected set of 11 features (subset 4: [0 1 2 3 4 5 6 7 9 10 11]) for Ordonez A dataset ($K = 3$)

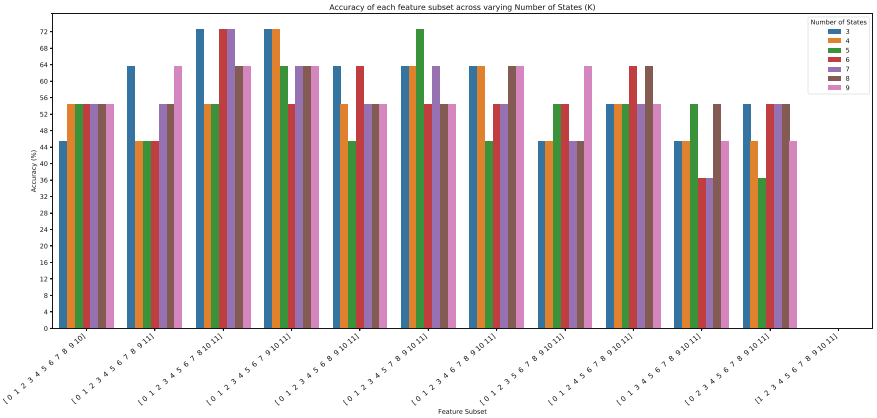


Fig. 40 Accuracy of the backward elimination selected 11 features across trained HMMs with various number of states for Ordonez B dataset

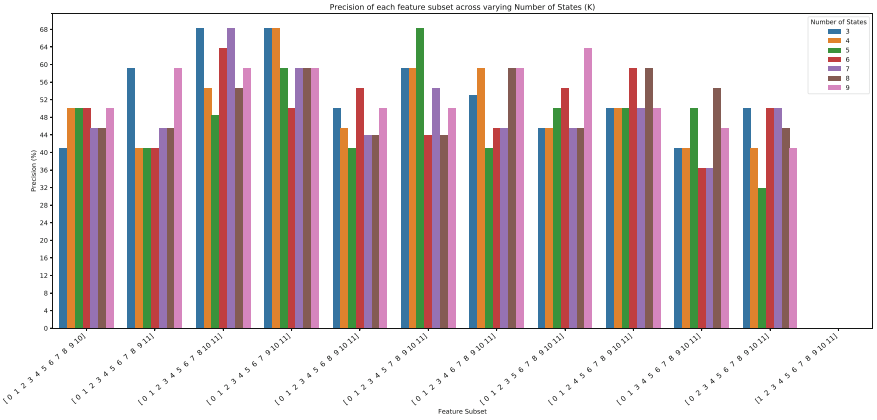


Fig. 41 Precision of the backward elimination selected 11 features across trained HMMs with various number of states for Ordonez B dataset

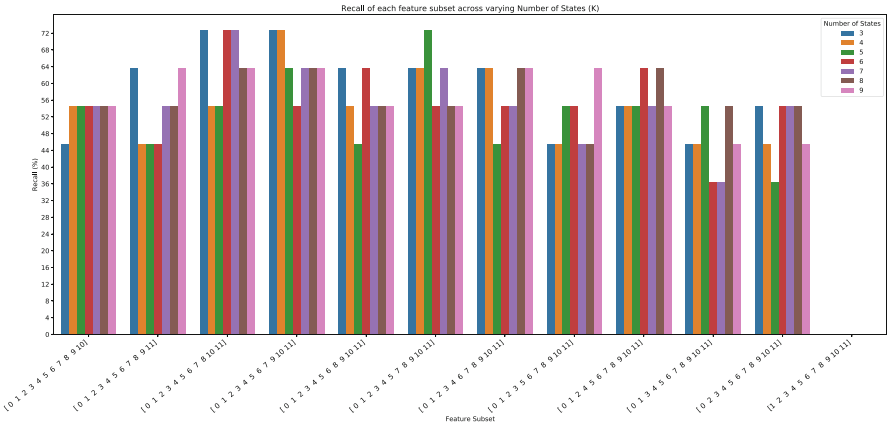


Fig. 42 Recall of the backward elimination selected 11 features across trained HMMs with various number of states for Ordonez B dataset

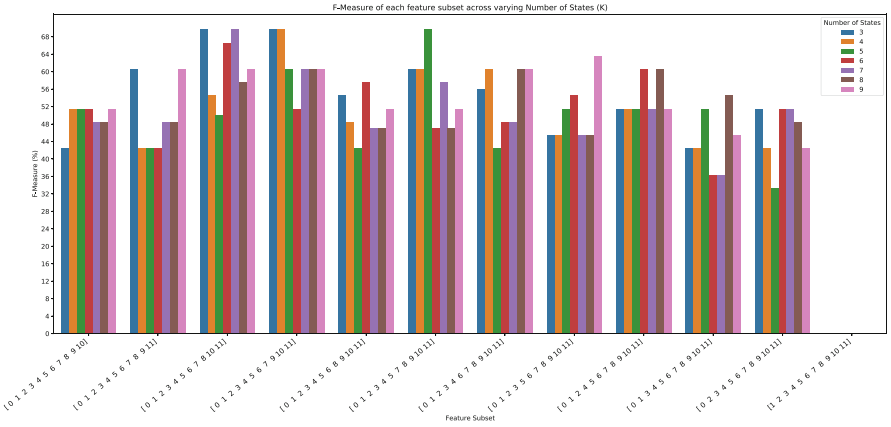


Fig. 43 F-measure of the backward elimination selected 11 features across trained HMMs with various number of states for Ordonez B dataset

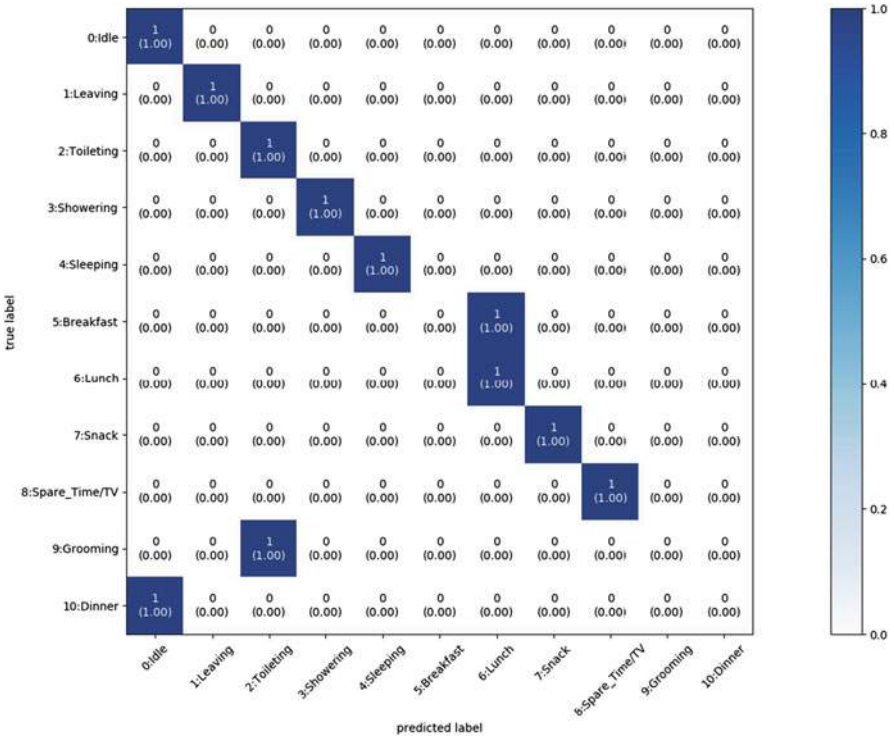


Fig. 44 Confusion matrix of the model trained with backward elimination selected set of 11 features (subset 3: [0 1 2 3 4 5 6 7 8 10 11]) for Ordonez B dataset ($K = 3$)

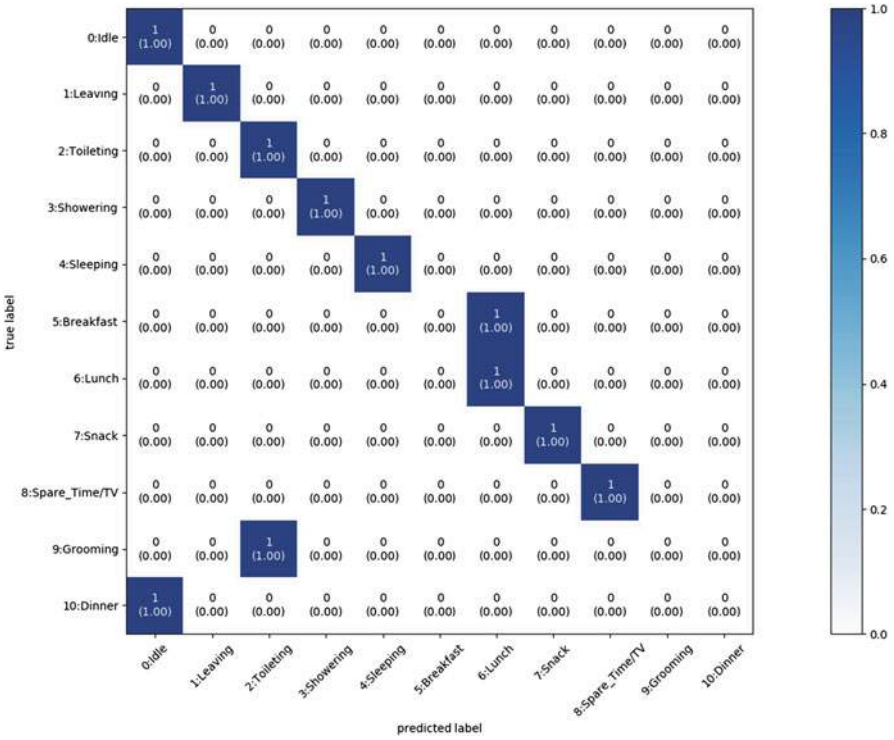


Fig. 45 Confusion matrix of the model trained with backward elimination selected set of 11 features (subset 4: [0 1 2 3 4 5 6 7 9 10 11]) for Ordonez B dataset ($K = 3$)

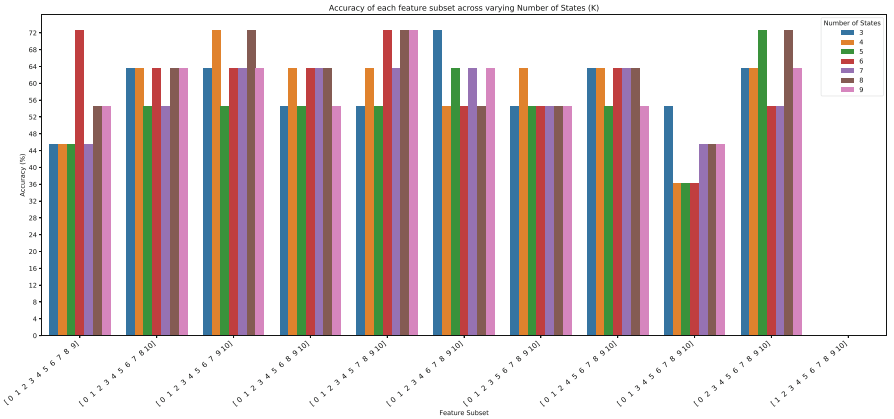


Fig. 46 Accuracy of the backward elimination selected 10 features across trained HMMs with various number of states for Ordonez B dataset

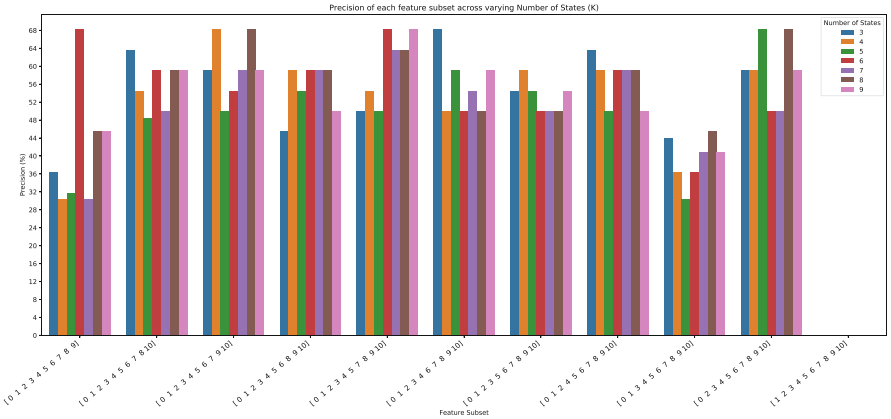


Fig. 47 Precision of the backward elimination selected 10 features across trained HMMs with various number of states for Ordonez B dataset

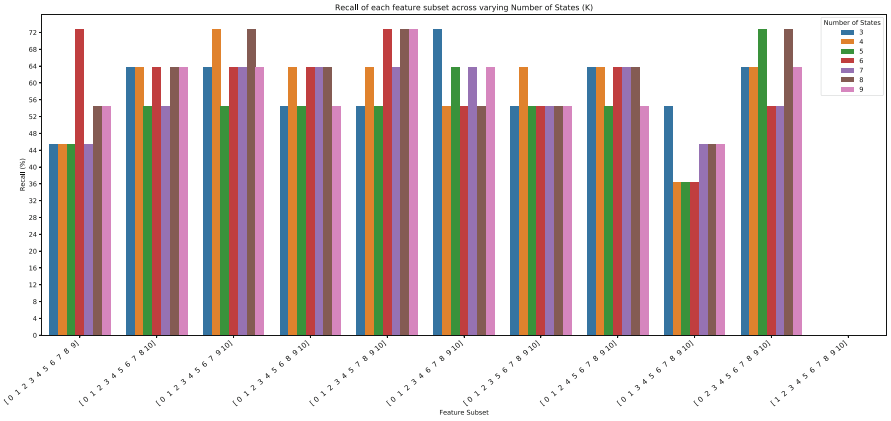


Fig. 48 Recall of the backward elimination selected 10 features across trained HMMs with various number of states for Ordonez B dataset

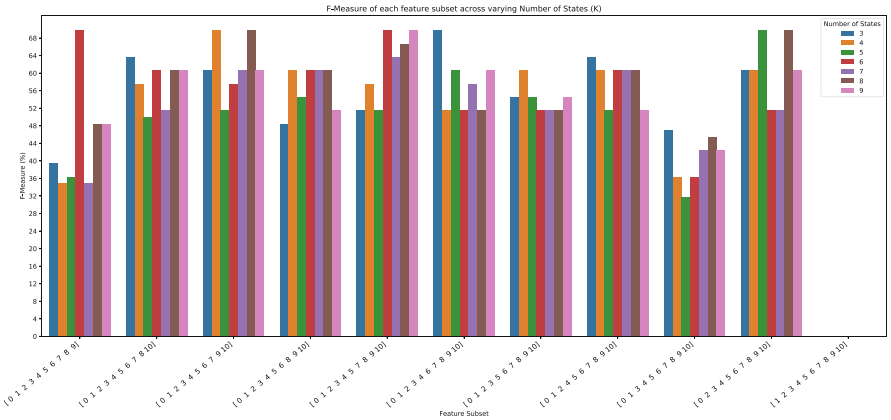


Fig. 49 F-measure of the backward elimination selected 10 features across trained HMMs with various number of states for Ordonez B dataset

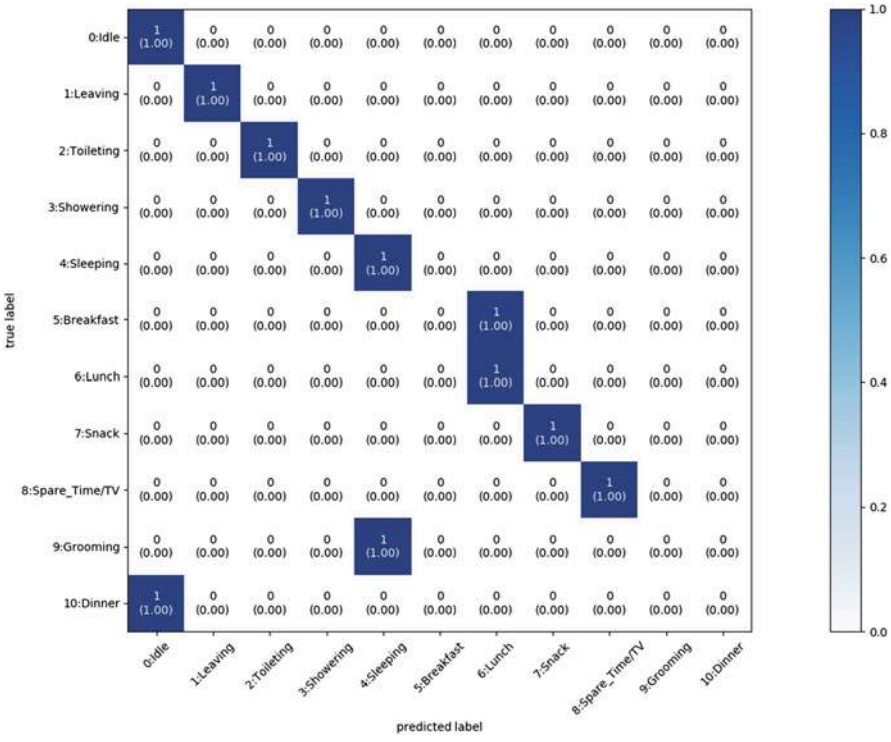


Fig. 50 Confusion matrix of the model trained with backward elimination selected set of 10 features (subset 6: [0 1 2 3 4 6 7 8 9 10]) for Ordonez B dataset ($K = 3$)

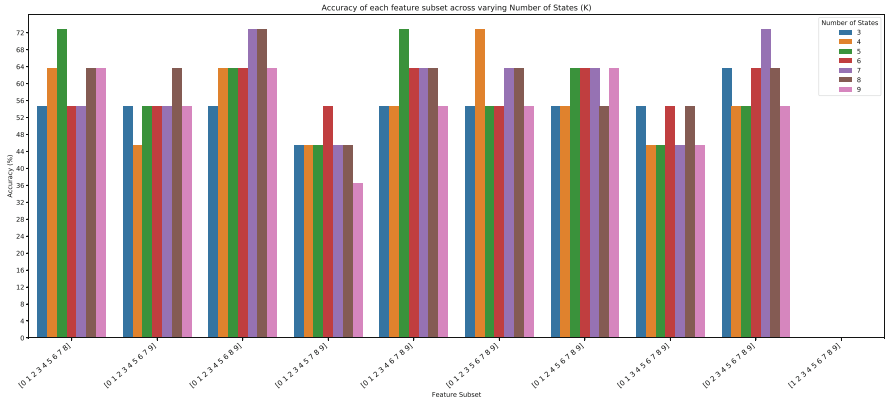


Fig. 51 Accuracy of the backward elimination selected 9 features across trained HMMs with various number of states for Ordonez B dataset

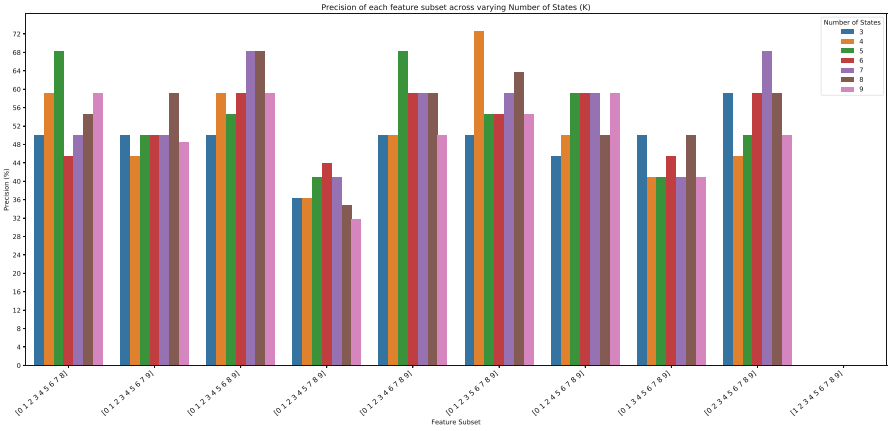


Fig. 52 Precision of the backward elimination selected 9 features across trained HMMs with various number of states for Ordonez B dataset

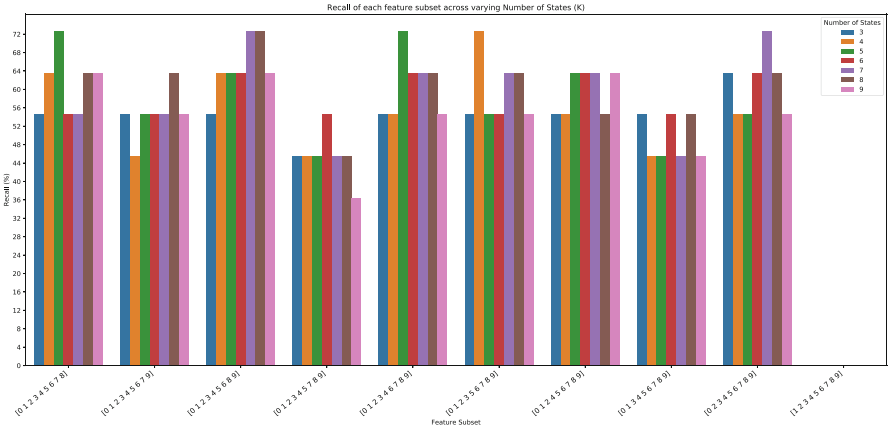


Fig. 53 Recall of the backward elimination selected 9 features across trained HMMs with various number of states for Ordonez B dataset

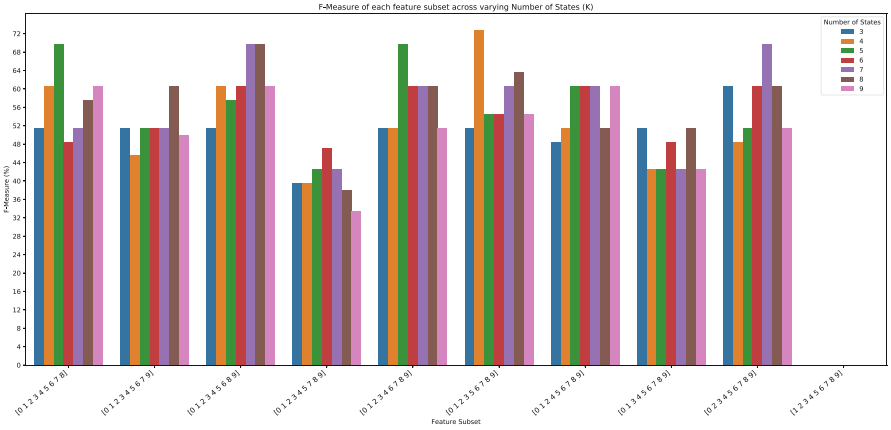


Fig. 54 F-measure of the backward elimination selected 9 features across trained HMMs with various number of states for Ordonez B dataset

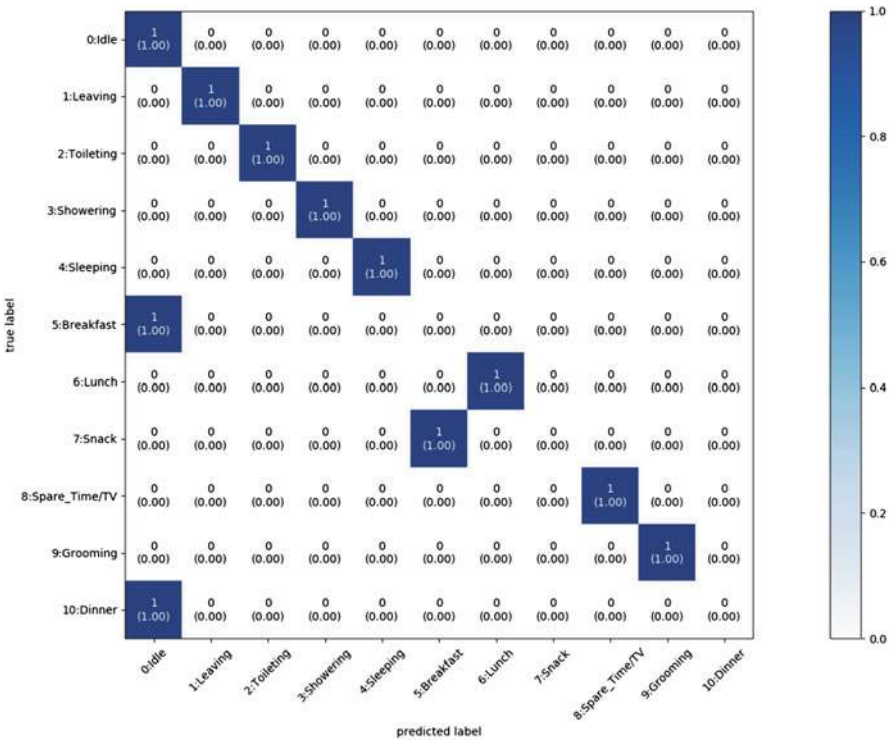


Fig. 55 Confusion matrix of the model trained with backward elimination selected set of 9 features (subset 6: [0 1 2 3 5 6 7 8 9]) for Ordonez B dataset ($K = 4$)

When selecting 8 features in the next step, we are able to improve the accuracy, the precision, the recall, and the F-measure to 81.82% across the board. This is a significant improvement. It comes with a more complex model of $K = 9$ for feature subset 3: [0 1 2 3 4 5 7 8]. This may be observed in Figs. 52, 53, and 54 across the performance metrics (Figs. 55, 56, 57, 58, and 59) with the confusion matrix of the optimum performing feature subset shown in Fig. 60. The encoded feature 6 is dropped in the next stage.

When selecting a feature subset of 7 features in the next stage, the results of the backward elimination technique in terms of accuracy, precision, recall, and F-measure are shown, respectively, in Figs. 61, 62, 63, and 64. The optimum feature subset is the fourth one ([0 1 2 3 5 6 7]) with $K = 3$ now. This yields a performance that is on bar with the benchmark at a lower model complexity as well as training on nearly half the number of features. Given this feature subset, we now remove the encoded feature 4 in the next round of feature selection (Fig. 65).

The performance of the model remains consistent when selecting 6 features. However, the number of states is increased to 4 for the chosen subset. The accuracy, precision, recall, and F-measure performance metrics of all trained models on all

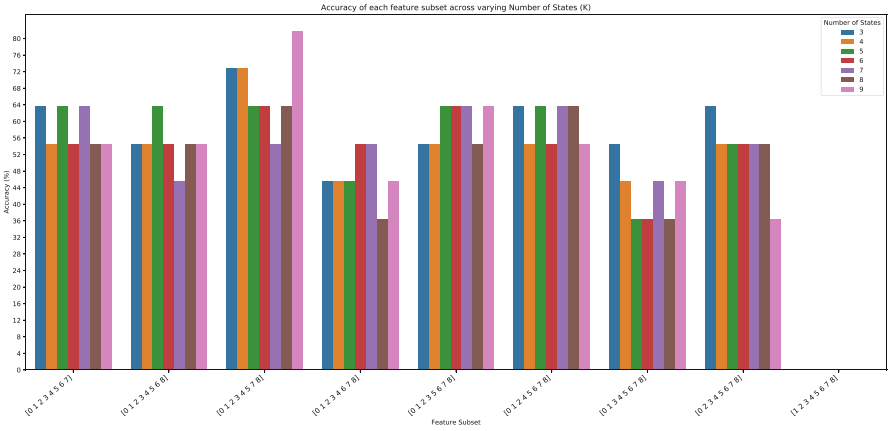


Fig. 56 Accuracy of the backward elimination selected 8 features across trained HMMs with various number of states for Ordonez B dataset

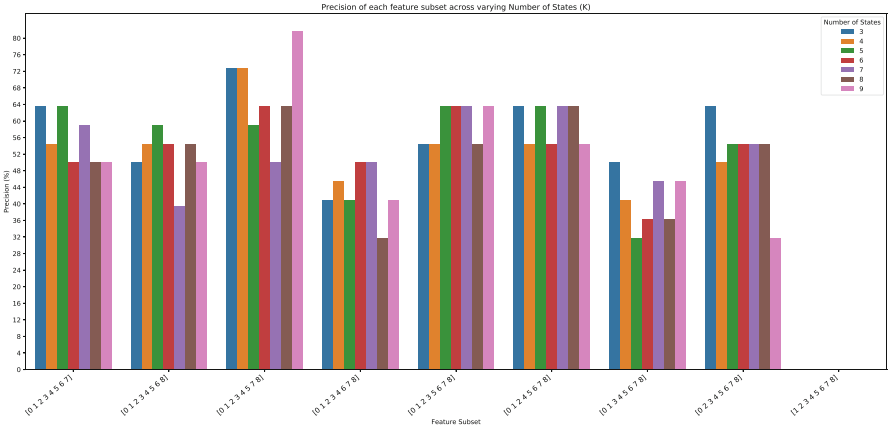


Fig. 57 Precision of the backward elimination selected 8 features across trained HMMs with various number of states for Ordonez B dataset

subsets of 6 features are shown in Figs. 66, 67, 68, and 69, respectively. The confusion matrix of the optimum model is shown in Fig. 70. The selected feature set does not have the encoded feature 6, so we remove it in the next stage of feature selection.

The next stage witnesses an improvement in the performance at 81.82% accuracy, 77.27% precision, 81.82% recall, and 78.79% F-measure levels for the optimum chosen feature subset 2: [0 1 2 3 5] as shown in Figs. 71, 72, 73, and 74, respectively. We compute the next feature subsets accordingly without the encoded feature 4. Nonetheless, it is noteworthy to mention that the complexity of the model now becomes larger given that $K = 8$. The confusion matrix of the chosen trained model is illustrated in Fig. 75.

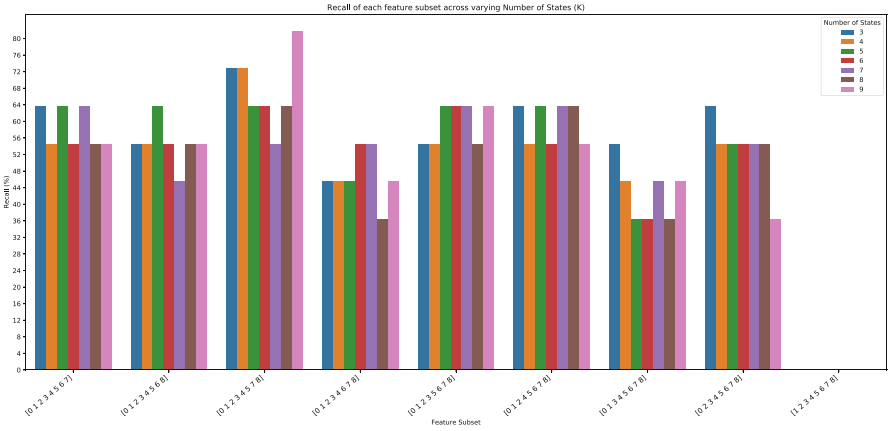


Fig. 58 Recall of the backward elimination selected 8 features across trained HMMs with various number of states for Ordonez B dataset

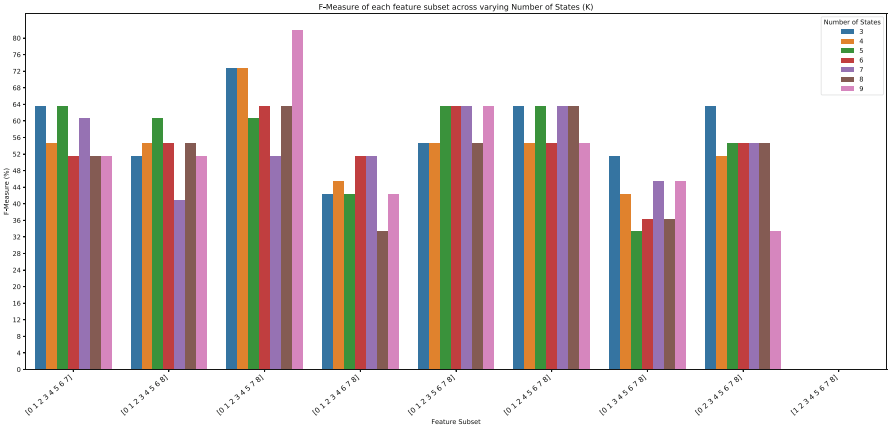


Fig. 59 F-measure of the backward elimination selected 8 features across trained HMMs with various number of states for Ordonez B dataset

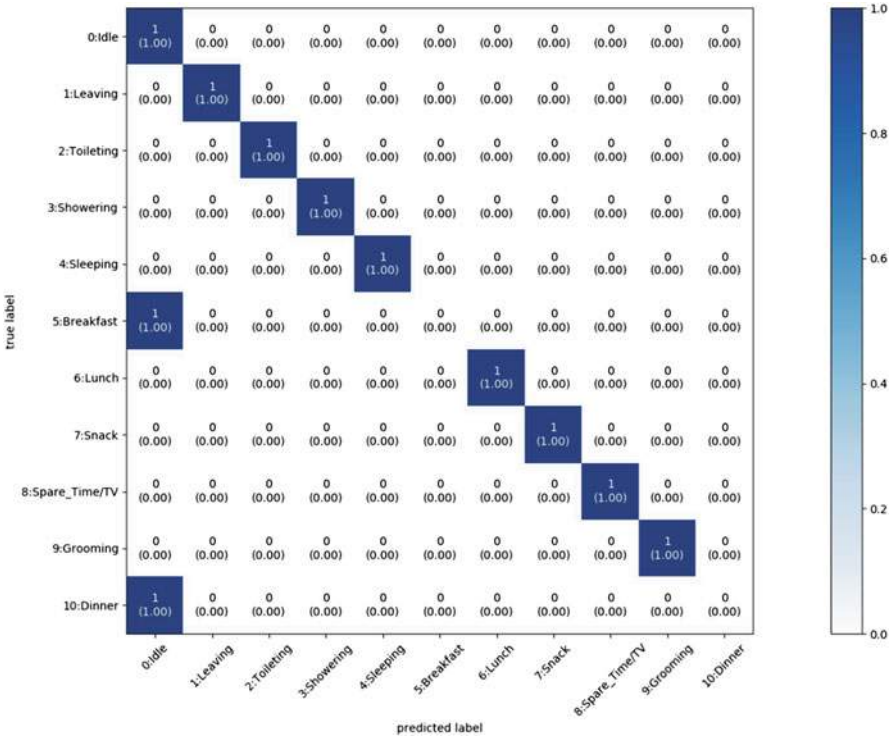


Fig. 60 Confusion matrix of the model trained with backward elimination selected set of 8 features (subset 3: [0 1 2 3 4 5 7 8]) for Ordonez B dataset ($K = 9$)

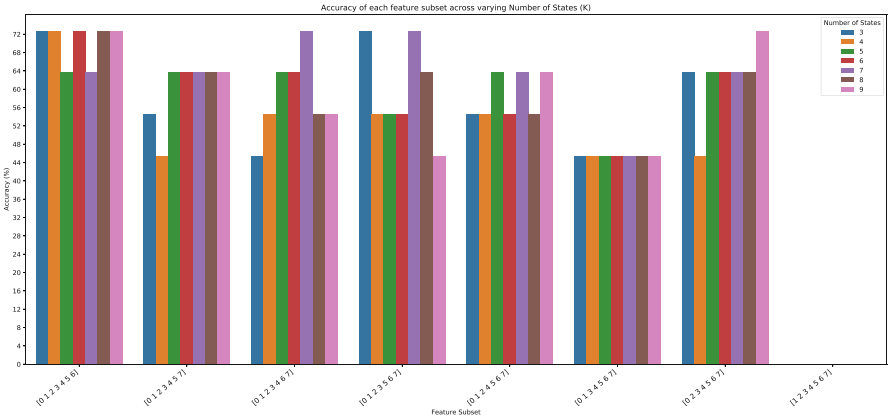


Fig. 61 Accuracy of the backward elimination selected 7 features across trained HMMs with various number of states for Ordonez B dataset

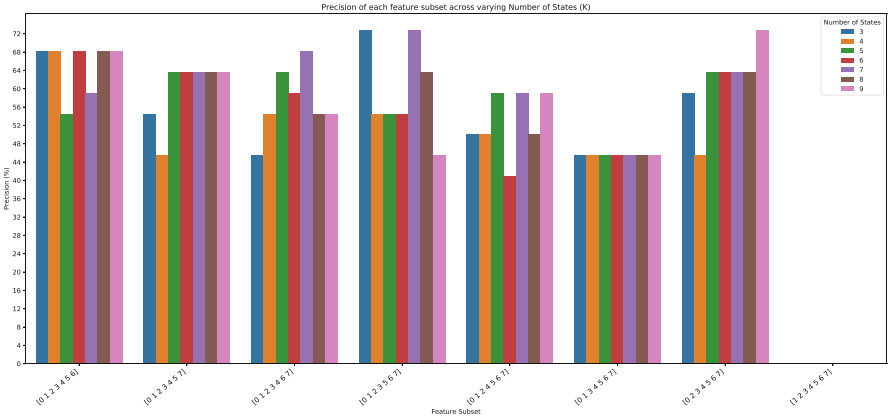


Fig. 62 Precision of the backward elimination selected 7 features across trained HMMs with various number of states for Ordonez B dataset

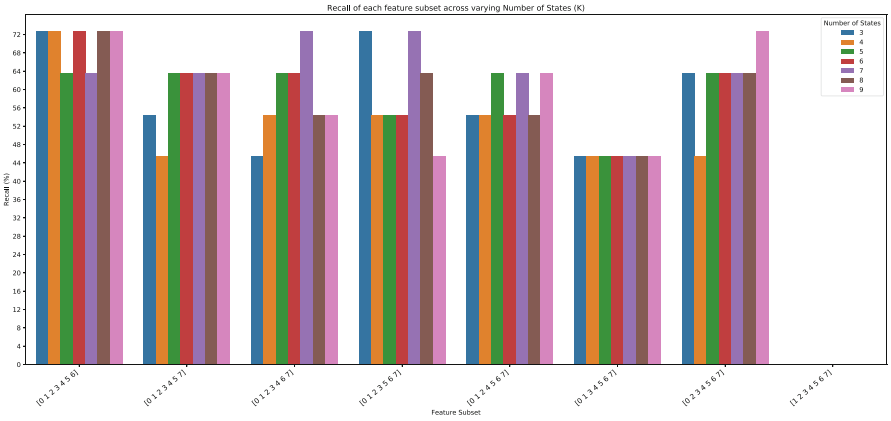


Fig. 63 Recall of the backward elimination selected 7 features across trained HMMs with various number of states for Ordonez B dataset

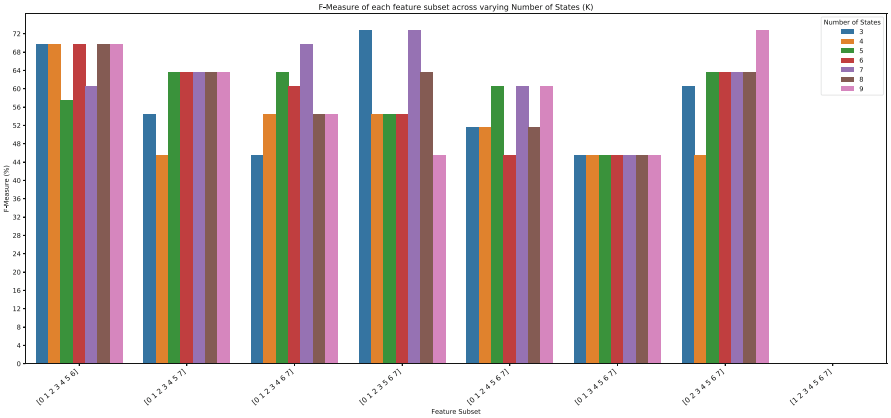


Fig. 64 F-measure of the backward elimination selected 7 features across trained HMMs with various number of states for Ordonez B dataset

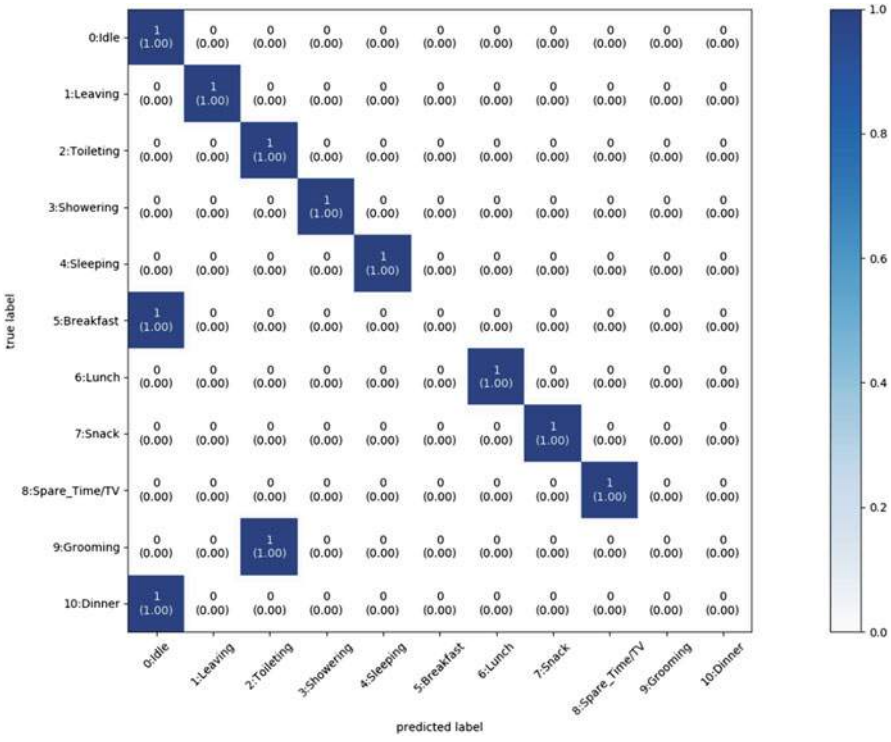


Fig. 65 Confusion matrix of the model trained with backward elimination selected set of 7 features (subset 4: [0 1 2 3 5 6 7]) for Ordonez B dataset ($K = 3$)

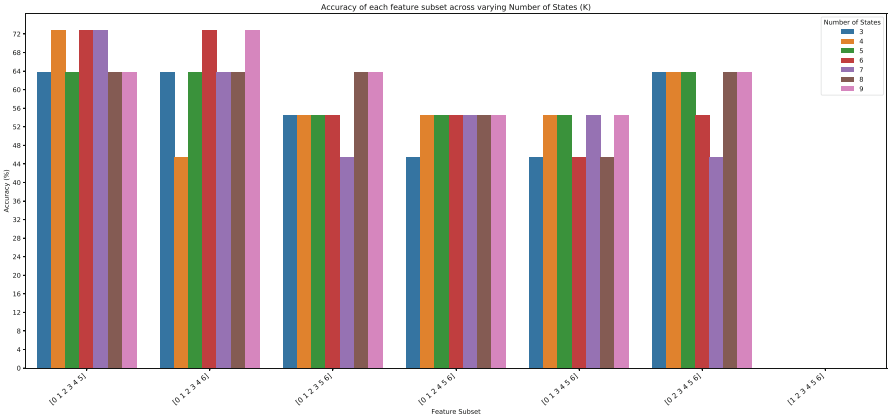


Fig. 66 Accuracy of the backward elimination selected 6 features across trained HMMs with various number of states for Ordonez B dataset

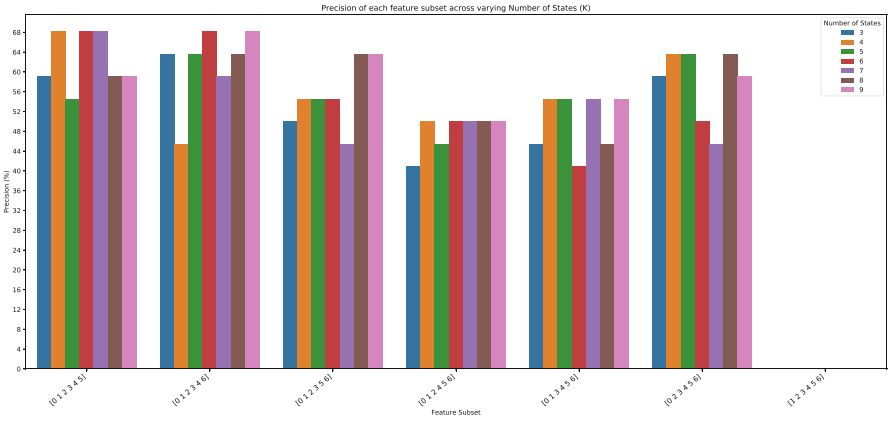


Fig. 67 Precision of the backward elimination selected 6 features across trained HMMs with various number of states for Ordonez B dataset

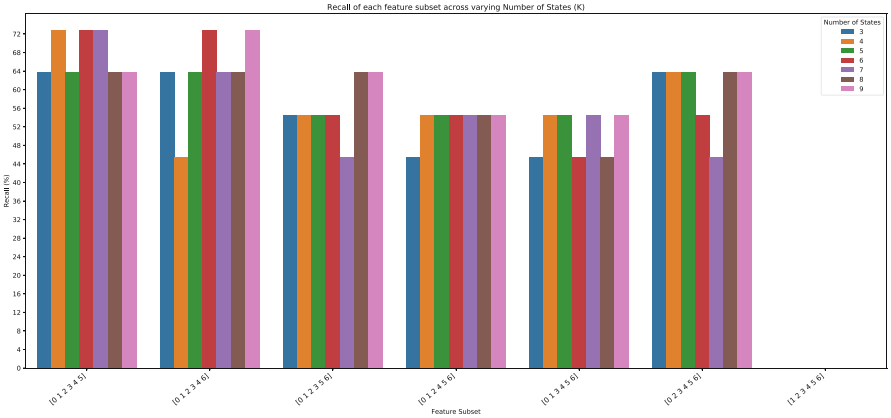


Fig. 68 Recall of the backward elimination selected 6 features across trained HMMs with various number of states for Ordonez B dataset

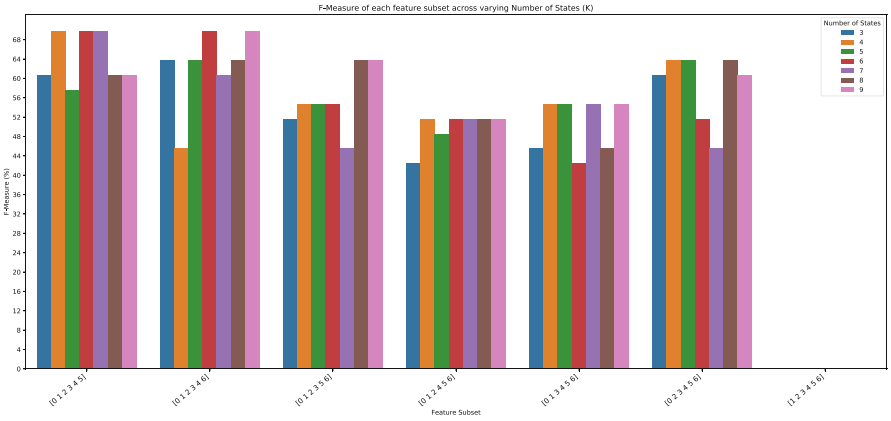


Fig. 69 F-measure of the backward elimination selected 6 features across trained HMMs with various number of states for Ordonez B dataset

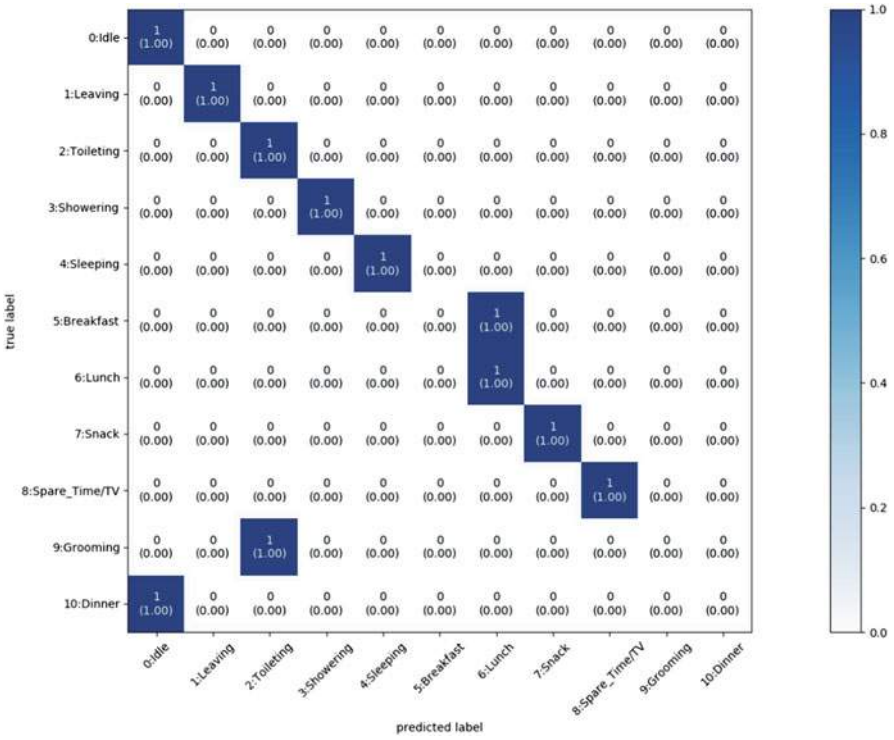


Fig. 70 Confusion matrix of the model trained with backward elimination selected set of 6 features (subset 1: [0 1 2 3 4 5]) for Ordonez B dataset ($K = 4$)

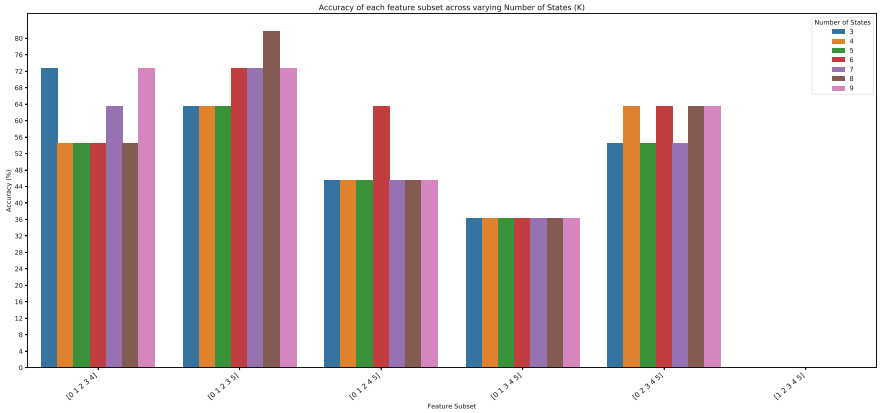


Fig. 71 Accuracy of the backward elimination selected 5 features across trained HMMs with various number of states for Ordonez B dataset

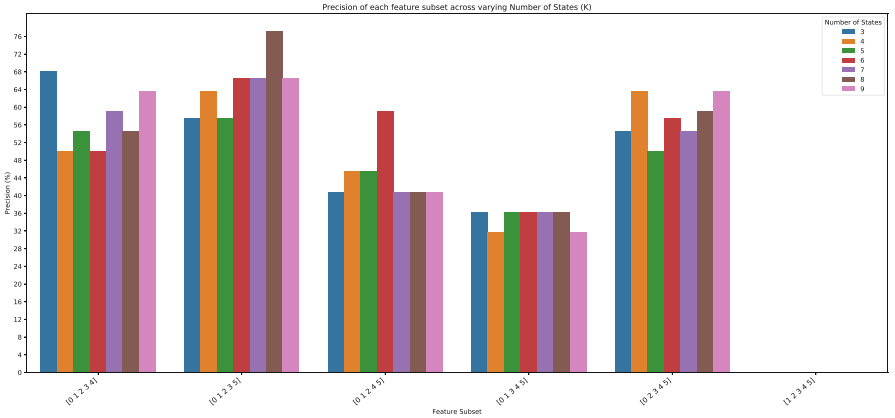


Fig. 72 Precision of the backward elimination selected 5 features across trained HMMs with various number of states for Ordonez B dataset

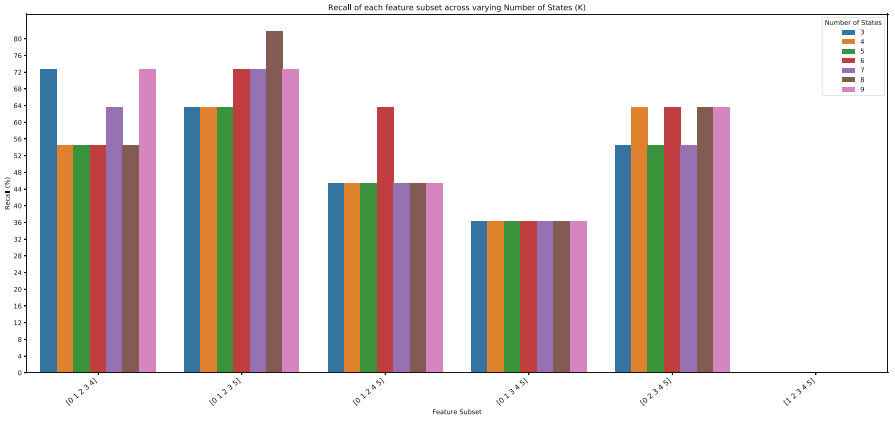


Fig. 73 Recall of the backward elimination selected 5 features across trained HMMs with various number of states for Ordonez B dataset

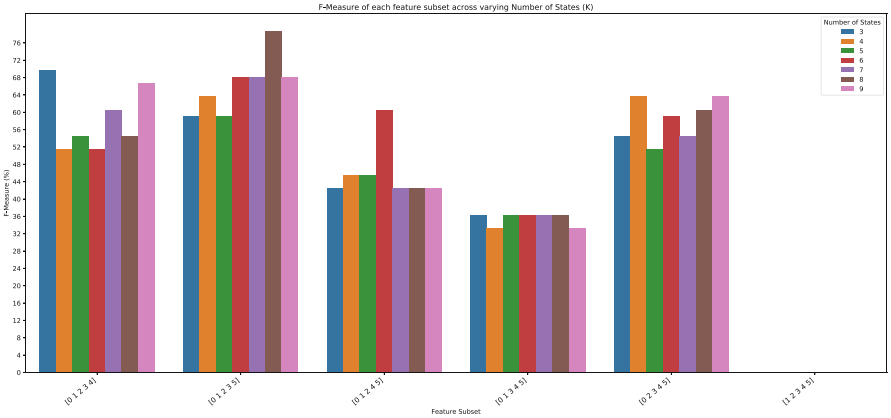


Fig. 74 F-measure of the backward elimination selected 5 features across trained HMMs with various number of states for Ordonez B dataset

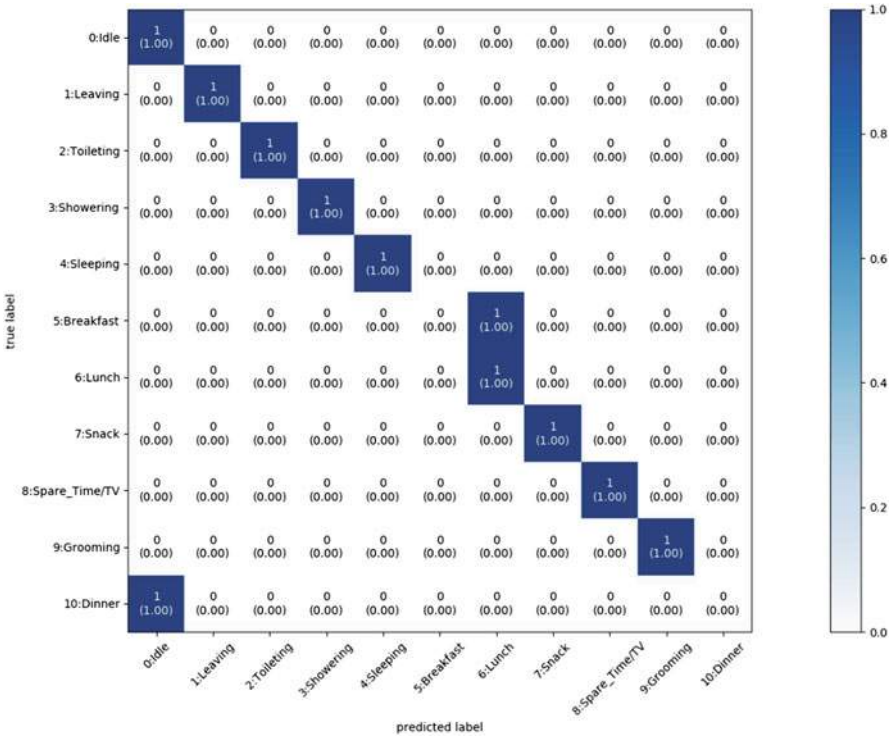


Fig. 75 Confusion matrix of the model trained with backward elimination selected set of 5 features (subset 2: [0 1 2 3 5]) for Ordonez B dataset ($K = 8$)

In the next stage, we note a degradation in all the performance metrics as shown in Figs. 76, 77, 78, and 79. This indicates that the optimum feature subset has been achieved in the previous step. The physical correspondence of the final chosen features are the following for the Ordonez B dataset: shower PIR sensor in bathroom, basin PIR sensor in bathroom, door PIR sensor in kitchen, maindoor magnetic sensor in the entrance, and microwave electric sensor in the kitchen.

We can also infer a general conclusion that the subset that does not include the first feature is always incapable of training properly on both levels for all the models (i.e., features). In the final iteration, we also note that this phenomenon is also shown by another feature subset. Overall, this technique was capable of optimizing the

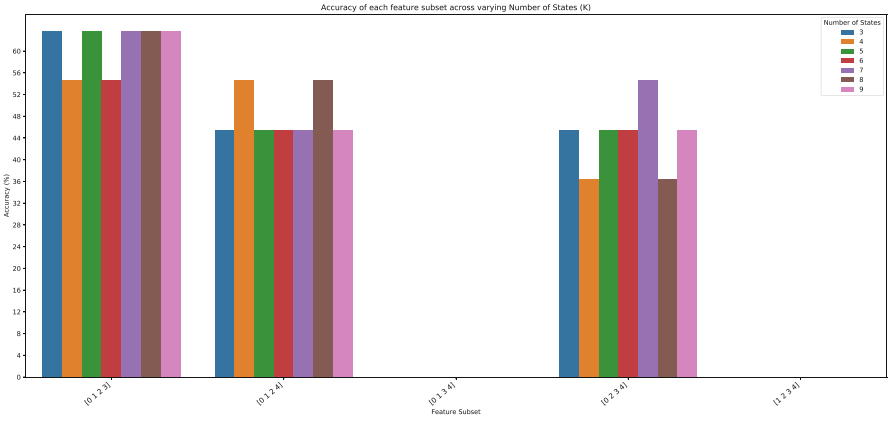


Fig. 76 Accuracy of the backward elimination selected 4 features across trained HMMs with various number of states for Ordonez B dataset

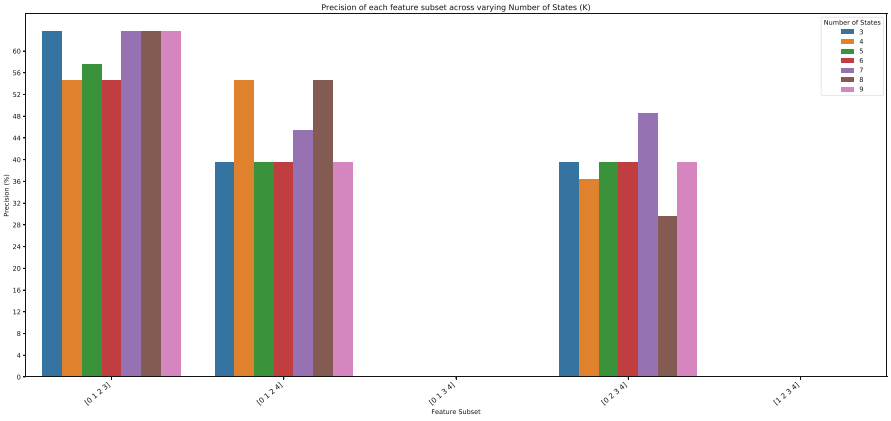


Fig. 77 Precision of the backward elimination selected 4 features across trained HMMs with various number of states for Ordonez B dataset

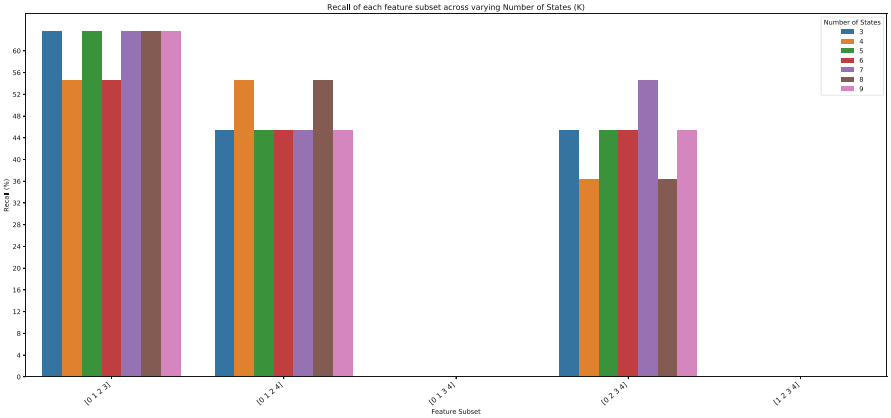


Fig. 78 Recall of the backward elimination selected 4 features across trained HMMs with various number of states for Ordonez B dataset

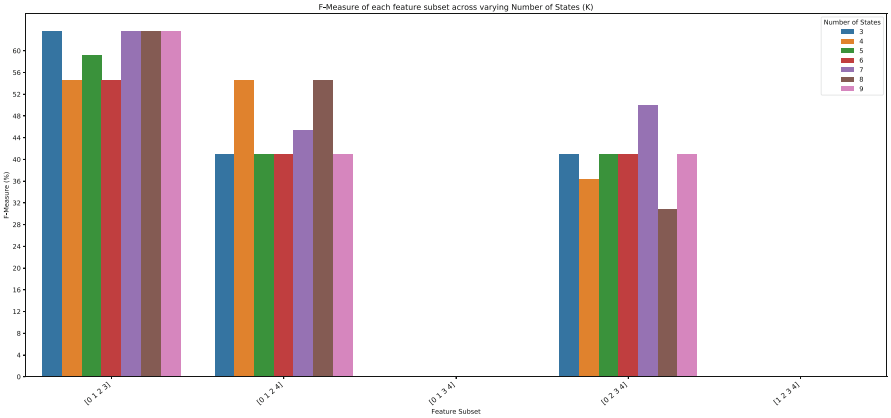


Fig. 79 F-measure of the backward elimination selected 4 features across trained HMMs with various number of states for Ordonez B dataset

feature set for the Ordonez B to be lower than the half of the original number; a no small feat.

4 Conclusion

In conclusion, this chapter investigated filter- and wrapper-based feature selection techniques for discrete data. The modelling of the data was carried out with a multinomial HMM and appropriate model selection was performed. The investigations were executed for the challenging indoor activity recognition task

with ambient sensor binary features. Though the data was highly imbalanced, the presented experimental setup is able to tackle this issue in an elegant manner. It also facilitates flexibility, scalability, and robustness of the model. Overall, we found out that the backward elimination, i.e., the wrapper-based method applied, was the most appropriate for the dataset at hand. Its results on the Ordonez B dataset are particularly encouraging. Future works may include the fusion of the sensor data as well as relevant feature engineering as study cases with simplistic derived features have shown promise.

Acknowledgments This work was funded and supported by Ericsson—Global Artificial Intelligence Accelerator in Montreal and a Mitacs Accelerate fellowship.

References

1. M. Angelidou, Smart cities: a conjuncture of four forces. Cities **47**, 95–106 (2015). Current Research on Cities (CRoC). https://www.sciencedirect.com/science/article/pii/S0264275115000633
2. M.M. Rathore, A. Ahmad, A. Paul, S. Rho, Urban planning and building smart cities based on the internet of things using big data analytics. Comput. Netw. **101**, 63–80 (2016). Industrial Technologies and Applications for the Internet of Things. https://www.sciencedirect.com/science/article/pii/S1389128616000086
3. W. Shen, G. Newsham, B. Gunay, Leveraging existing occupancy-related data for optimal control of commercial office buildings: a review. Adv. Eng. Inf. **33**, 230–242 (2017). http://www.sciencedirect.com/science/article/pii/S1474034616301987
4. X. Dai, J. Liu, X. Zhang, A review of studies applying machine learning models to predict occupancy and window-opening behaviours in smart buildings. Energy Build. **223**, 110159 (2020). http://www.sciencedirect.com/science/article/pii/S0378778820303017
5. K. Akkaya, I. Guvenc, R. Aygun, N. Pala, A. Kadri, IoT-based occupancy monitoring techniques for energy-efficient smart buildings, in *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)* (2015), pp. 58–63
6. M. Yoshida, S. Kleisarchaki, L. Gtirgen, H. Nishi, Indoor occupancy estimation via location-aware hmm: an IoT approach, in *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)* (2018), pp. 14–19
7. H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-khah, P. Siano, Iot-based smart cities: a survey, in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)* (2016), pp. 1–6
8. E. Kim, S. Helal, D. Cook, Human activity recognition and pattern discovery. IEEE Pervasive Comput. **9**(1), 48–53 (2010)
9. J. Guo, Y. Li, M. Hou, S. Han, J. Ren, Recognition of daily activities of two residents in a smart home based on time clustering. Sensors **20**(5), 1457 (2020)
10. L. Chen, J. Hoey, C.D. Nugent, D.J. Cook, Z. Yu, Sensor-based activity recognition. IEEE Trans. Syst. Man, Cybern. C **42**(6), 790–808 (2012)
11. M.H. Kolekar, D.P. Dash, Hidden Markov model based human activity recognition using shape and optical flow based features, in *2016 IEEE Region 10 Conference (TENCON)* (2016), pp. 393–397
12. M.H. Siddiqi, M. Alruwaili, A. Ali, S. Alanazi, F. Zeshan, Human activity recognition using gaussian mixture hidden conditional random fields. Comput. Intell. Neurosci. **2019**, 8590560 (2019). https://doi.org/10.1155/2019/8590560

13. H. Sagha, S.T. Digumarti, J. del R. Millaan, R. Chavarriaga, A. Calatroni, D. Roggen, G. Trauster, Benchmarking classification techniques using the opportunity human activity dataset, in *2011 IEEE International Conference on Systems, Man, and Cybernetics* (2011), pp. 36–40
14. L.R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
15. E. Epaillard, 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 Recognit. **55**, 125–136 (2016). https://doi.org/10.1016/j.patcog.2016.02.004
16. E. Epaillard, N. Bouguila, Variational Bayesian learning of generalized dirichlet-based hidden Markov models applied to unusual events detection. IEEE Trans. Neural Netw. Learn. Syst. **30**(4), 1034–1047 (2019). https://doi.org/10.1109/TNNLS.2018.2855699
17. E. Epaillard, N. Bouguila, Data-free metrics for dirichlet and generalized dirichlet mixture-based hmms - A practical study. Pattern Recognit. **85**, 207–219 (2019). https://doi.org/10.1016/j.patcog.2018.08.013
18. S. Ali, N. Bouguila, Variational learning of beta-liouville hidden Markov models for infrared action recognition, in *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, June 16–20, 2019* (Computer Vision Foundation/IEEE, Piscataway, 2019), pp. 898–906. http://openaccess.thecvf.com/content_CVPRW_2019/html/PBVS/Ali_Variational_Learning_of_Beta-Liouville_Hidden_Markov_Models_for_Infrared_Action_CVPRW_2019_paper.html
19. E. Epaillard, N. Bouguila, Hidden Markov models based on generalized dirichlet mixtures for proportional data modeling, in *Artificial Neural Networks in Pattern Recognition*, ed. by N. El Gayar, F. Schwenker, C. Suen (Springer, Cham, 2014), pp. 71–82
20. R. Nasfi, M. Amayri, N. Bouguila, A novel approach for modeling positive vectors with inverted dirichlet-based hidden markov models. Knowl.-Based Syst. **192**, 105335 (2020). http://www.sciencedirect.com/science/article/pii/S0950705119306057
21. S. Ali, N. Bouguila, Hybrid generative-discriminative generalized dirichlet-based hidden Markov models with support vector machines, in *2019 IEEE International Symposium on Multimedia (ISM)* (IEEE, Piscataway, 2019), pp. 231–2311
22. S. Ali, 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, Piscataway, 2019), pp. 1–5
23. S. Ali, N. Bouguila, Online learning for beta-liouville hidden markov models: incremental variational learning for video surveillance and action recognition, in *2020 IEEE International Conference on Image Processing (ICIP)* (IEEE, Piscataway, 2020), pp. 3249–3253
24. Y. Kim, B. Kang, D. Kim, Hidden Markov model ensemble for activity recognition using tri-axis accelerometer, in *2015 IEEE International Conference on Systems, Man, and Cybernetics* (2015), pp. 3036–3041
25. C. Yang, Z. Wang, B. Wang, S. Deng, G. Liu, Y. Kang, H. Men, Char-hmm: an improved continuous human activity recognition algorithm based on hidden Markov model, in *International Conference on Mobile Ad-Hoc and Sensor Networks* (Springer, Berlin, 2017), pp. 271–282
26. M.H. Kabir, M.R. Hoque, K. Thapa, S.-H. Yang, Two-layer hidden Markov model for human activity recognition in home environments. Int. J. Distrib. Sens. Netw. **12**(1), 4560365 (2016). https://doi.org/10.1155/2016/4560365
27. T. van Kasteren, A. Noulas, G. Englebienne, B. Kröse, Accurate activity recognition in a home setting, in *Proceedings of the 10th International Conference on Ubiquitous Computing, ser. UbiComp'08*. (Association for Computing Machinery, New York, 2008), pp. 1–9. https://doi.org/10.1145/1409635.1409637
28. N.A. Capela, E.D. Lemaire, N. Baddour, Feature selection for wearable smartphone-based human activity recognition with able bodied, elderly, and stroke patients. *Plos One* **10**(4), 1–18 (2015). https://doi.org/10.1371/journal.pone.0124414

29. M. Shafiq, Z. Tian, A.K. Bashir, X. Du, M. Guizani, IoT malicious traffic identification using wrapper-based feature selection mechanisms. Comput. Secur. **94**, 101863 (2020). https://www.sciencedirect.com/science/article/pii/S0167404820301358
30. S. Boutemedjet, D. Ziou, N. Bouguila, Unsupervised feature selection for accurate recommendation of high-dimensional image data, in *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, December 3–6, 2007*, ed. by J.C. Platt, D. Koller, Y. Singer, S.T. Roweis (Curran Associates, New York, 2007), pp. 177–184. https://proceedings.neurips.cc/paper/2007/hash/073b00ab99487b74b63c9a6d2b962ddc-Abstract.html
31. M.A. Mashrgy, T. Bdiri, N. Bouguila, Robust simultaneous positive data clustering and unsupervised feature selection using generalized inverted dirichlet mixture models. Knowl. Based Syst. **59**, 182–195 (2014). https://doi.org/10.1016/j.knosys.2014.01.007
32. T. Bdiri, N. Bouguila, D. Ziou, Variational bayesian inference for infinite generalized inverted dirichlet mixtures with feature selection and its application to clustering. Appl. Intell. **44**(3), 507–525 (2016). https://doi.org/10.1007/s10489-015-0714-6
33. N. Bouguila, K. Almakadmeh, S. Boutemedjet, A finite mixture model for simultaneous high-dimensional clustering, localized feature selection and outlier rejection. Exp. Syst. Appl. **39**(7), 6641–6656 (2012). https://doi.org/10.1016/j.eswa.2011.12.038
34. T. Elguebaly, N. Bouguila, Simultaneous high-dimensional clustering and feature selection using asymmetric gaussian mixture models. Image Vis. Comput. **34**, 27–41 (2015). https://doi.org/10.1016/j.imavis.2014.10.011
35. N. Bouguila, D. Ziou, A countably infinite mixture model for clustering and feature selection. Knowl. Inf. Syst. **33**(2), 351–370 (2012). https://doi.org/10.1007/s10115-011-0467-4
36. The SAGE encyclopedia of educational research, measurement, and evaluation AU - Frey, Bruce B. Thousand Oaks, California (2018). https://doi.org/10.4135/9781506326139
37. S. Balakrishnama, A. Ganapathiraju, Linear discriminant analysis-a brief tutorial. Inst. Signal Inf. Process. **18**(1998), 1–8 (1998)
38. M. Cherrington, F. Thabtah, J. Lu, Q. Xu, Feature selection: filter methods performance challenges, in *2019 International Conference on Computer and Information Sciences (ICCIS)* (2019), pp. 1–4
39. J.R. Vergara, P.A. Estévez, A review of feature selection methods based on mutual information. Neural Comput. Appl. **24**(1), 175–186 (2014). https://doi.org/10.1007/s00521-013-1368-0
40. B. Wu, L. Zhang, Y. Zhao, Feature selection via cramer's v-test discretization for remote-sensing image classification. IEEE Trans. Geosci. Remote Sens. **52**(5), 2593–2606 (2014)
41. X. Geng, T.-Y. Liu, T. Qin, H. Li, Feature selection for ranking, in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ser. SIGIR'07*. (Association for Computing Machinery, New York, 2007), pp. 407–414. https://doi.org/10.1145/1277741.1277811
42. D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Faurster, G. Trauster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, J.D.R. Millaan, Collecting complex activity datasets in highly rich networked sensor environments, in *2010 Seventh International Conference on Networked Sensing Systems (INSS)* (2010), pp. 233–240
43. J. Pohle, R. Langrock, F.M. van Beest, N.M. Schmidt, Selecting the number of states in hidden Markov models: pragmatic solutions illustrated using animal movement. J. Agric. Biol. Environ. Stat. **22**(3), 270–293 (2017)

Bayesian Inference of Hidden Markov Models Using Dirichlet Mixtures

Ravi Teja Vemuri, Muhammad Azam, Zachary Patterson, and Nizar Bouguila

1 Introduction

Artificial Intelligence (AI) has become an integral part of many technical solutions due to the generation of large amounts of data from many sources. In general, AI is used as an umbrella term to represent any machine or program capable of showing characteristics such as self-optimization, learning, or inference. It can also be defined as a system that is programmed to perform cognitive tasks similar to those of human beings such as image recognition, natural language processing, speech processing, etc.

Machine learning (ML) [2, 17] is a form of AI that enhances computers to go beyond their capacities by involving various programming techniques and algorithms. In ML, large amounts of data [27, 40] are processed to help "machines" evolve with each iteration, which is referred to as learning. As data are essential to learning, having organized data flow is critical for any ML algorithm to function. The more clean, varied, machine-readable the data is, the more efficient is the learning process. In ML, there are different types of learning algorithms that fall under these three categories: supervised learning [24], unsupervised learning [11, 20, 25], and reinforcement learning [44]. There are two kinds of data: labeled, where the data is already segregated into groups based on certain identified characteristics, and unlabelled, where the data is not segregated or tagged, are fed to these algorithms based on their type.

R. T. Vemuri · M. Azam · Z. Patterson · N. Bouguila (✉)
Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, QC, Canada
e-mail: r_vemur@encs.concordia.ca; mu_azam@encs.concordia.ca; zachary.patterson@concordia.ca; nizar.bouguila@concordia.ca

N. Bouguila et al. (eds.), *Hidden Markov Models and Applications*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-99142-5_6

In supervised learning, models are trained on labeled data for tasks such as classification [26] or regression. Here the algorithm is given a data set that is sampled from the whole population of the data. The data is then divided into training and testing sets, with a majority of the data in the training set. In the training phase, using the algorithm, the machine is trained to identify and classify the data as per labels. In the testing phase, the machine's learning ability is tested using a testing subset.

In unsupervised learning, data are unlabelled, and the goal is generally to find data clusters [38, 46] based on various factors like data similarity, etc., resulting in knowledge discovery. In this type of learning, the algorithm is left to run for several iterations on the data to find out the hidden structures in the data, and the learning is concluded when the learning is not progressing anymore, i.e., the algorithm reached a convergence point.

In reinforcement learning, the algorithm tries to mimic the learning process in human beings, learning from data in their lives using a trial-and-error method, i.e., favorable outputs are encouraged, and others are discouraged. This type of learning process puts the algorithm in a test environment directly with a reward system that decides whether each output is favorable. When the outcome is favorable, the algorithm is rewarded, and in all other cases, the output is passed.

In this chapter, we are interested in exploring and proposing novel unsupervised learning techniques. Nowadays, a great deal of data are mostly unlabelled, which motivated us to propose our approach. In machine learning also there are two kinds of learning approaches: stochastic and deterministic [32]; in the former, it is believed that randomness in the learning process is efficient, and in the latter, the learning process involves no randomness. Stochastic learning is further applied to two families of approaches: generative and discriminative [12] modeling; in the first method, learning is based on joint probabilities, and in the other, it is based on conditional probabilities. Our proposed approach follows stochastic generative modeling in a Bayesian framework involving HMMs and mixture models.

HMMs [19, 33] are advanced statistical models that follow a Markov process [29] in system modeling. Such processes assume that an observable sequence of observations is dependent on some hidden information and try to learn about that based on visible observations. HMMs are efficient in speech recognition applications and in any sequence analysis and time series analysis. HMMs are usually used to represent dependent heterogeneous events because of which, they are applied in various fields such as: econometrics [23], biology [28], genetics [18], speech processing [33], and particularly finance [39].

HMMs are associated with a significant problem of choosing the number of states. According to the classical approach solving this problem would require hypothesis testing with complex parameters. This approach is usually not recommended and considered only an alternative because of regularity conditions, and when asymptotic theory is not applicable. The likelihood ratio test needs to be approximated using simulation techniques, which demand high computation. In addition, the Akaike and Bayesian information criteria can be used, but they fail to produce required confidence in the results as they are susceptible to over

fitting and cannot handle high dimensional data. In a Bayesian setting, there are different approaches for choosing the number of states in an HMM. For instance, a Bayesian nonparametric methodology is presented in [31] in view of the Dirichlet process. A downside of this methodology is that a single parameter controls the clustering, making it difficult for prior specifications. A characteristic option in contrast to the Dirichlet process model is to utilize mixture models with multinomial allocations, which is the Dirichlet mixture model. Following [37] and [35], we utilize a completely Bayesian framework, in light of the Reversible Jump (RJMCMC) algorithm, proposed in [21], which permits, for the change in dimension of the parameter space, changing the number of states from one iteration to the next. The algorithm also permits estimating the joint posterior distribution of the number of states and all the parameters. In this chapter, we demonstrate parameter estimation and model selection with HMMs using MCMC [1] and Reversible Jump techniques [22].

The rest of the chapter is organized as follows: Sect. 2 describes our HMM modeling approach using mixture models in a Bayesian setting. Section 3 explains the MCMC and RJMCMC algorithms employed for parameter estimation and model selection in HMM. In Sect. 4, we experiment with a few applications to demonstrate the effectiveness of our model in fitting real-world data. Finally, in Sect. 5, we conclude our research.

2 The Learning Model

In this section, we present the proposed model, which is a combination of various components, and we elaborate on each component and its contribution to the parameter estimation and model selection processes. As we proceed, the following sections and subsections will give more details about our modeling approach.

2.1 The Bayesian Model

As previously mentioned in the chapter, we choose to implement a Bayesian approach [36] in our modeling, in which Θ represents a vector of parameters describing the model. For a given data set $\mathcal{Y}$, Bayes' theorem is

$$p(\Theta|\mathcal{Y}) \propto p(\mathcal{Y}|\Theta)p(\Theta) \tag{1}$$

where $p(\mathcal{Y}|\Theta)$ is the likelihood and $p(\Theta)$ is the prior distribution of the parameter set. Later on, in this section, we discuss in detail the prior distributions of our parameter set and the complete hierarchical model, which is the heart of the proposed modeling approach where the joint probability is computed.

2.2 Mixture Model

Mixture models are probabilistic models which aim to fit data containing a given number of clusters using the same or different probabilistic distributions where k represents the number of clusters, often called a mixture of k components, where each component fits a cluster of the whole data. A d-dimensional random variable $y = [y_1, \ldots, y_d]^T$ is said to follow a mixture of k components if its probability density function takes following form:

$$\phi(y; \Theta) = \sum_{j=1}^{k} \pi_i \phi(y; \xi_j) \tag{2}$$

where $\xi_j = (\mu_j, \alpha_j)$ is the set of parameters for the jth component, π_j are the mixing probabilities, which are always positive and sum to 1. $\mu_j = (\mu_{j1}, \ldots, \mu_{jd})$ is the mean and $\alpha_j = (\alpha_{j1}, \ldots, \alpha_{jd})$ is the sharpness parameter. Θ, the set of all parameters, is represented as $\Theta = \{\pi_1, \ldots, \pi_k, \xi_1, \ldots, \xi_k\}$. Here, k represents the number of components in the mixture and is always greater than or equal to 1. $\phi(y; \mu_j, \alpha_j)$ in Eq. (2) is the density of the Dirichlet distribution [15]:

$$\phi(y_t; \mu_j, \alpha_j) = \frac{\Gamma(|\alpha_j|)}{\prod_{i=1}^{d} \Gamma\left(\mu_{ji}|\alpha_j|\right)} \prod_{i=1}^{d} y_i^{\mu_{ji}|\alpha_j|-1} \tag{3}$$

where $\sum_{i=1}^{d} y_i = 1$ and $|\alpha_j| = \sum_{i=1}^{d} \alpha_{ji},\ \alpha_{ji} > 0\ \forall i = 1, \ldots, d$. Given α the mean and variance of the Dirichlet distribution can be given as follows:

$$\mu_d = E(y_d) = \frac{\alpha_d}{|\alpha|} \tag{4}$$

$$\sigma_d^2 = Var(y_d) = \frac{\alpha_d(|\alpha| - \alpha_d)}{|\alpha|^2(|\alpha| + 1)} \tag{5}$$

2.3 Hidden Markov Model

Given, $y = (y_t)_{t=1}^{T}$ which are the vectors of observations with respect to time T, HMMs assume that the distribution of each data point y_t depends on hidden states, which are unobserved and are denoted by s_t and can take values from 1 to k. The hidden variable $s = (s_t)_{t=1}^{T}$ is often called a "regime" or "state"—we adopt the former word throughout the rest of the chapter. In HMMs that follow the Markov chain property, it is assumed that the hidden state variable s_t always depends on past

realizations of y and s as shown in Eq. (6):

$$p(s_t = j|s_{t-1} = i) = a_{ij} \tag{6}$$

where a_{ij} is the element of a transition probability matrix denoted by $A = (a_{ij})$. A transition probability matrix A, where each row is a vector of stationary probabilities given by π and satisfies $\pi^{'} A = \pi^{'}$, and stationary probabilities decide the initial state of the model from which, with time, the state transition takes place. As it is assumed in HMMs that, for every state change s_t an observation y_t is noticed, which follows a marginal probability distribution given in Eq. (2). The same equation can also be represented as:

$$y_t|\pi, \mu, \alpha \sim \Sigma_{i=1}^{k} \pi_i \phi(y_t; \mu_i, \alpha_i) \tag{7}$$

Equation (7) can also be expressed as follows involving s_t:

$$y_t|s, \pi, \mu, \alpha \sim \Sigma_{i=1}^{k} \pi_i \phi(y_t; \mu_{s_t}, \alpha_{s_t}) \tag{8}$$

Here, we assume that the number of components k (i.e., the number of states) is unknown and subject to inference and we can observe that for $k = 1$ the model in Eq. (7) reduces to a simple random walk with drift.

2.3.1 Prior Distributions

In any Bayesian modeling approach [13, 36, 37], prior information is one's belief about an unknown quantity before considering any evidence about it. Usually, prior information would be a probability distribution describing the unknown parameters in a model. Since the prior information is a probability distribution describing a parameter, the parameters of such a prior distribution are called hyper-parameters. In our case, we have three prior distributions for three unknown parameters $(\mu_i, \alpha_i, a_{ij})$ of the model, found in Eqs. (9), (10), and (11) as follows:

$$\mu_i \sim \mathcal{D}(\delta_1, \ldots, \delta_k) \sim \frac{\Gamma(\sum_{j=1}^{k} \delta_j)}{\prod_{j=1}^{k} \Gamma(\delta_j)} \prod_{j=1}^{k} \mu_{ij}^{\delta_j - 1} \tag{9}$$

$$a_{ij} \sim \mathcal{D}(\eta_1, \ldots, \eta_k) \tag{10}$$

where the mean μ_i and each row of the transition probability matrix a_{ij} has a Dirichlet distribution as prior with $\delta = \{\delta_1, \ldots, \delta_k\}$ and $\eta = \{\eta_1, \ldots, \eta_k\}$ as the hyper-parameters. The sharpness parameter $\alpha = \{\alpha_1, \ldots, \alpha_k\}$ has an inverse Gamma as a prior as follows:

$$\alpha_i \sim |\alpha_i|^{-3/2} \exp\left(-1/(2|\alpha_i|)\right) \tag{11}$$

2.3.2 Complete Hierarchical Model

The joint probability distribution like in [37] for all the variables including their hyper-parameters can be represented according to Eq. (12) as follows:

$$p(k, A, \mu, \alpha, s, y) = p(k)p(A|k, \eta)p(\mu|\delta, k)p(\alpha)p(s|A)p(y|\mu, \alpha, s) \tag{12}$$

$$\text{where} \qquad p(s|A) = p(s_1|A)\prod_{t=2}^{T} p(s_t|s_{t-1}, A) \tag{13}$$

The term $p(s_t|s_{t-1}, A)$ from Eq. (13) is given by Eq. (6) and $p(s_1 = i|A) = \pi_i$, and from Eq. (12)

$$p(y|s, \mu, \alpha) = \prod_{t=1}^{T} \phi(y_t; \mu_{s_t}, \alpha_{s_t}) \tag{14}$$

Figure 1 is a directed acyclic graph (DAG) representing the complete hierarchical model in which the usual convention is followed where the square boxes represent fixed or observed quantities and the circles represent the unknowns.

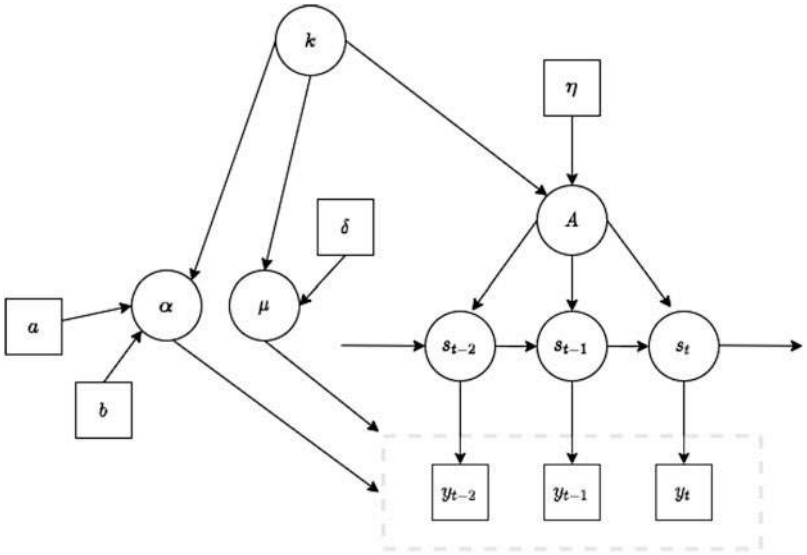


Fig. 1 Directed acyclic graph for the complete hierarchical model

3 Markov Chain Monte Carlo Methodology

The mixture model [14] approach taken in this chapter is a fairly complex one and requires MCMC techniques to approximate the posterior distribution. A detailed description of these computational techniques can be found in [16, 37]. To get the realizations from the posterior joint distribution with all the parameters, we use the following moves at each sweep of the MCMC algorithm :

1. Update the transition probability matrix A
2. Update the allocations s
3. Update the mean μ
4. Update the sharpness parameter α
5. Update standard deviation σ
6. Consider split or combine moves
7. Consider birth or death moves

3.1 Gibbs Moves

Moves from (1–5) presented in the algorithm are called Gibbs moves and follow [37]. In move 1, the ith row of A is sampled from a Dirichlet distribution $D(\eta + n_{i1}, \ldots, \eta + n_{ik})$ where:

$$n_{ij} = \sum_{t=1}^{T-1} I\{s_t = i, s_{t+1} = j\} \tag{15}$$

is the number of jumps from state i to state j. In move 2, the state allocations $s_1, \ldots s_T$ are sampled one at a time from $t = 1, .., T$ by drawing new values from the full conditional distribution in Eq. (16). For $t = 1$, the first factor is replaced by the stationary probability π_i, and for $t = T$, the rightmost factor is replaced by 1. Here $\phi(y_t; \mu_i, \alpha_i)$ is the density of a Dirichlet random variable with mean μ_i and sharpness parameter α_i.

$$p(s_t = i|S) \sim a_{s_{t-1}i}\phi(y_t; \mu_i, \alpha_i)a_{is_{t+1}} \tag{16}$$

where $S = \{s_1, \ldots, s_T\}$. In move 3, the mean μ_i is updated by sampling from a log-normal distribution whose mean is the natural log of the mean from the previous iteration with a transformation $\mu^*_{il} = \log(\frac{\mu_{il}}{(1-\mu_{il})})$ and with Σ^2 as co-variance matrix where $\Sigma^2 = diag[0.01, \ldots, 0.01]$. The whole equation is represented in Eq. (17).

$$\mu^*_i \sim \mathcal{LN}(log(\mu_i^{*(t-1)}), \Sigma^2) \tag{17}$$

In move 4, α_i is updated in a similar fashion like μ_i with the only difference being where the parameters of the log-normal distribution are changed, and Σ^2 is replaced by σ^2 whose value is 0.01, the equation is given as Eq. (18). As α_i is updated, in move 5, σ is updated using the new values of α according to Eq. (5).

$$|\alpha_i| \sim \mathcal{LN}(log(|\alpha_i|^{t-1}), \sigma^2) \tag{18}$$

3.2 Split and Combine Moves

In the above mentioned moves, 6 and 7 are considered as reversible jump MCMC moves which allow the number of components to increase or decrease by 1. In move 6, a state at a given point of time is chosen to split or combine with probabilities b_k and $d_k = 1 - b_k$, respectively. As $d_1 = b_{k_{max}} = 0$. We use $b_k = d_k = \frac{1}{2}$ for $k = 2, 3, \ldots, k_{max} - 1$. In the combine move we suppose the current state of the MCMC algorithm is $\tilde{x}$ with $\tilde{a}_{ij}$, etc., as parameters with a total of $k+1$ states. Then, we randomly choose a pair (j_1, j_2) which are two adjacent states and combine them to a single new state j_* resulting in an MCMC with x states and k components.

For the combine move, the parameters are updated as follows Eqs. (19), (20), and (21):

$$\mu_{j^*} = \frac{\tilde{\pi}_{j_1}\mu_{j_1} + \tilde{\pi}_{j_2}\mu_{j_2}}{\tilde{\pi}_{j_1} + \tilde{\pi}_{j_2}} \tag{19}$$

$$\mu_{j^*}^2 + \sigma_{j^*}^2 = \frac{\tilde{\pi}_{j_1}(\mu_{j_1}^2 + \sigma_{j_1}^2) + \tilde{\pi}_{j_2}(\mu_{j_2}^2 + \sigma_{j_2}^2)}{\tilde{\pi}_{j_1} + \tilde{\pi}_{j_2}} \tag{20}$$

$$\begin{aligned} a_{j^*j} &= \frac{\tilde{\pi}_{j_1}}{\tilde{\pi}_{j_1} + \tilde{\pi}_{j_2}}\tilde{a}_{j_1 j} + \frac{\tilde{\pi}_{j_2}}{\tilde{\pi}_{j_2} + \tilde{\pi}_{j_2}}\tilde{a}_{j_2 j} \quad \text{for } j \neq j*, \\ a_{ij_*} &= \tilde{a}_{ij_1} + \tilde{a}_{ij_2} \quad \text{for } i \neq j*, \end{aligned} \tag{21}$$

and for any t with $\tilde{s}_t$ equal to $j_1 or j_2$, s_t is set to j_* and the remaining $\tilde{s}_t$ are simply copied. Similarly, a state j_* is selected at random in the split move and split into two new components j_1 and j_2.

In the split move, a state j_* is randomly selected and split into two new states j_1 and j_2. In the old representation, we assume that x is the current state with a total of k states and after the move is executed the system is represented with $\tilde{x}$ with a total of $k+1$ states. The goal of the split move is to split j_* in a way that the stationary probabilities for the chain of hidden states satisfy the following: $\tilde{\pi}_j = \pi_j \ \ for\ j \neq j_1, j_2, \tilde{\pi}_{j_1} = u_0\pi_{j_*}\ \ and\ \ \tilde{\pi}_{j_2} = (1-u_0)\pi_{j_*}$. This can be achieved by sampling $u_0 \sim Be(2,2), u_j \sim Be(r,s)\ for\ each\ j \neq j_1, j_2$ and $v_i \sim Be(r,s)\ for\ each\ i \neq j_1, j_2$. The parameters of the Beta distribution r and s are given from Eq. (23). The

transition probabilities $\tilde{A}$ after the split are updated according to the Eq. (22) where $K_i = \frac{\pi_i}{\pi_{j_*}}$:

$$\begin{aligned}
\tilde{a}_{j_1 j} &= \frac{u_j}{u_0} a_{j_* j}, \qquad \tilde{a}_{j_2 j} = \frac{1-u_j}{1-u_0} a_{j_* j} \qquad \text{for } j \neq j_1, j_2, \\
\tilde{a}_{i j_1} &= v_i a_{i j_*}, \qquad \tilde{a}_{i j_2} = (1-v_i) a_{i j_*} \qquad \text{for } i \neq j_1, j_2, \\
\tilde{a}_{j_1 j_2} &= u_1 \left(1 - \sum_{i \neq j_*} \frac{u_j}{u_0} a_{j_* j}\right), \\
\tilde{a}_{j_2 j_1} &= \left\{(1-u_1) \sum_{j \neq j_*} u_j a_{j_* j} + u_0 u_1 - \sum_{i \neq j_*} K_i v_i a_{i j_*}\right\} \Big/ (1-u_0)
\end{aligned} \tag{22}$$

$$\begin{aligned}
r &= \frac{1-u_0(1+c^2)}{c^2}, \quad s = r\frac{1-u_0}{u_0} \qquad if\ u_0 \leq \frac{1}{2} \\
s &= \frac{1-(1-u_0)(1+c^2)}{c^2}, \quad r = s\frac{u_0}{1-u_0} \qquad if\ u_0 > \frac{1}{2}
\end{aligned} \tag{23}$$

Here c^2 is known as the squared coefficient of variation of the Beta distribution, and for the reasons mentioned in [37] we assume it to be $c^2 = 0.5$ for numerical result stability. We discuss u_1, which is the range for $\tilde{A} : [u_1^L, u_1^U]$ and is given as follows:

$$\begin{aligned}
u_1^L &= max\left(1 - \frac{1 - \sum_{i \neq j_1, j_2} K_i/u_0 \times \tilde{a}_{i j_1}}{1 - \sum_{j \neq j_1, j_2} \tilde{a}_{j_1, j}}, 0\right), \\
u_1^U &= min\left\{1 - \frac{1 - \sum_{i \neq j_1, j_2} K - i/u_0 \times \tilde{a}_{i j_1} - (1-u_0)/u_0 \times (1 - \sum_{j \neq j_1, j_2} \tilde{a}_{j_2 j})}{1 - \sum_{j \neq j_1, j_2} \tilde{a}_{j_1, j}}, 1\right\}
\end{aligned} \tag{24}$$

During the split move if $u_1^L > u_1^U$, it means that there is no valid range for $\tilde{A}$ and the move is rejected. If $u_1^L < u_1^U$, the move is not rejected and we can get the u_1 by $u_1 \sim u_1^L + (u_1^U - u_1^L) Be(1,1)$.

In the split move after splitting new parameters μ_{j_*}, σ_{j_*} are computed as follows:

$$\begin{aligned}
\tilde{\mu}_{j_1} &= \mu_{j_*} - z_1 \sigma_{j_*} \sqrt{\frac{\tilde{\pi}_{j_2}}{\tilde{\pi}_{j_1}}}, \qquad \tilde{\mu}_{j_2} = \mu_{j_*} - z_1 \sigma_{j_*} \sqrt{\frac{\tilde{\pi}_{j_1}}{\tilde{\pi}_{j_2}}} \\
\tilde{\sigma}_{j_1}^2 &= z_2 (1 - z_1^2) \sigma_{j_*}^2 \frac{\pi_{j_*}}{\tilde{\pi}_{j_1}}, \qquad \tilde{\sigma}_{j_2}^2 = (1 - z_2)(1 - z_1^2) \sigma_{j_*}^2 \frac{\pi_{j_*}}{\tilde{\pi}_{j_2}}
\end{aligned} \tag{25}$$

In this process of splitting, we make use of a two-dimensional random vector z which is sampled from a Beta distribution as $z_1 \sim z_1^U Be(1,1)$ and $z_2 \sim Be(1,1)$ and z_1^U is given in Eq. (26), which is the upper bound for z_1 and in which $\mu_i' s$ are

properly sorted.

$$z_1^U = min\left\{ \frac{\mu_{j_*} - \mu_{j_*-1}}{\sigma_{j_*}} \sqrt{\frac{\tilde{\pi}_{j_1}}{\tilde{\pi}_{j_2}}} \ , \ \frac{\mu_{j_*+1} - \mu_{j_*}}{\sigma_{j_*}} \sqrt{\frac{\tilde{\pi}_{j_2}}{\tilde{\pi}_{j_1}}} \ , \ 1 \right\} \tag{26}$$

At last, we choose to reallocate the observation that belongs to $s_t = j_*$ before splitting to j_1 and j_2. We achieve this by using a restricted backward algorithm. Let us assume that $s_t = j_*$ for $t_1 \leq t \leq t_2$ with $s_{t_1-1} \neq j_*$ and $s_{t_2+1} \neq j_*$. Then we sample $\tilde{s}_{t_1}, \ldots, \tilde{s}_{t_2}$ one at a time from $t = t_1$ to $t = t_2$ with conditional probabilities given as follows:

$$p(\tilde{s}_t = j|\Delta) \sim \tilde{a}_{\tilde{s}_{t-1},j} \phi(y_t; \tilde{\mu}_j, \tilde{\sigma}_j^2) b_t(i) \quad \text{for } j = j_1, j_2 \tag{27}$$

where $\Delta = \{y, \tilde{s}_{t_1-1}, \tilde{s}_{t_1}, \ldots, \tilde{s}_{t_1+1} \in [j_1, j_2], \ldots, \tilde{s}_{t_1} \in [j_1, j_2], \tilde{s}_{t_2+1}, \tilde{A}, \tilde{\mu}, \tilde{\sigma}\}$ and $b_t(i) = p(y_{t+1}, \ldots, y_{t_2}, \tilde{s}_{t_1+1} \in [j_1, j_2], \ldots, \tilde{s}_{t_1} \in [j_1, j_2], \tilde{s}_{t_2+1}|\tilde{s}_t = j, \tilde{A}, \tilde{\mu}, \tilde{\sigma})$ and for $j = j_1, j_2$

$$b_{t_2}(i) = \tilde{a}_{i,\tilde{s}_{t_2+1}} \tag{28}$$

For $t = t_2 - 1, \ldots, t_1$, $b_t(i)$ is given as follows:

$$b_t(i) = \sum_{j=i_1,i_2} b_{t+1}(j) \tilde{a}_{ij} \phi(y_{t+1}; \tilde{\mu}_j, \tilde{\sigma}_j^2) \tag{29}$$

when $t_1 = 1$, the $a_{\tilde{s}_{t-1},j}$ from Eq. (27) is replaced by $\tilde{\pi}_j$ which is the stationary probability and when $t_2 = T$, $\tilde{a}_{i,\tilde{s}_{t_2+1}}$ from Eq. (28) is replaced by 1.

As per the reverse jump algorithm, the acceptance probability of a split move is given as $min(1, R)$, and it is $min(1, R^{-1})$ for the combine move.

$$R = \frac{p(y|\tilde{s}, \tilde{\mu}, \tilde{\alpha})}{p(y|s, \mu, \alpha)} \times \frac{p(k+1)}{p(k)} \times \frac{p(\tilde{A}|k+1, \eta)}{p(A|k, \eta)} \times \frac{p(\tilde{s}|\tilde{A})}{p(s|A)} \times \frac{d_{k+1}}{b_k P_{\text{alloc}}} \times J \tag{30}$$

$$\times \left[\frac{1}{z_1^U} g_{1,1}\left(\frac{z_1}{z_1^U}\right) g_{1,1}(z_2) g_{2,2}(u_0) \frac{1}{u_1^U - u_1^L} g_{1,1}\left(\frac{u_1 - u_1^L}{u_1^U - u_1^L}\right) \prod_j g_{r,s}(u_j) \prod_i g_{r,s}(v_i) \right]^{-1}$$

where $g_{r,s}$ is the $Be(r, s)$ density, P_{alloc} is the probability of allocation for $\tilde{s}_t$, and J is the Jacobian determinant (explained in the Appendix).

3.3 Birth and Death Moves

Now we talk about birth and death moves as our final step in the RJMCMC algorithm. In this move, we randomly choose between birth and death with

probabilities b_k and d_k, respectively. An empty state is selected at random in the death move among all the empty states and deleted. Then the remaining rows A are normalized, and the s_t is not changed.

In birth move the aim is to create a new state j_*. To do this we sample j_* row which will be a new row of A from a Dirichlet prior $D(\delta, \ldots, \delta)$. Then we draw v_i $i \neq j_*$ from $Be(1, k)$ and set:

$$\begin{aligned} \tilde{a}_{ij} &= (1 - v_i)a_{ij} \qquad for j \neq j_*, \\ a_{ij_*} &= v_i \end{aligned} \tag{31}$$

The new parameters for this state are generated in the same way as previously mentioned, and the s_t remains untouched as the new state is empty. Similar to split and combine move the acceptance probability is computed to satisfy the rule of reversible jump where at any time t, the number of states can be increased or reduced. So, the acceptance probability for these moves is: $min(1, R)$ for birth and $min(1, R^{-1})$ for death.

$$R = \frac{p(k+1)}{p(k)} \times k^k \times \frac{p(\tilde{s}|\tilde{A})}{p(s|A)} \times (k+1) \times \frac{d_{k+1}}{b_k(k_0+1)} \left\{ \prod_i g_{1,k}(v_i) \right\}^{-1} \times J \tag{32}$$

where k_0 is the number of states before birth and J is the Jacobian determinant given by

$$J = \sum_{i \neq j_*} (1 - v_i)^{k-1} \tag{33}$$

The entire MH-within-Gibbs learning for HMM-DMM can be summarized as follows:
Input: Observations $\mathcal{X}$ with k number of components
Output: HMM-DMM parameter set, k components

1. Initialization
2. Step at time t: For $t = \{1, \ldots, n\}$

   Gibbs Sampling Part
   - Generate s from Eq. (16)
   - Generate $n_i j$ from Eq. (15)
   - Generate a_{ij} from Eq. (10)

Metropolis-Hastings and RJMCMC

- Sample μ_i, α_i from Eq. (17), (18)
- Compute σ_i from Eq. (5)
- Compute acceptance ratio (R) for split and combine move from Eq. (30)
- Compute acceptance ratio (R) for birth and death move from Eq. (32)

4 Experiments

In this section, we provide experiments to validate the proposed model with real-world applications. According to the literature [33], in many cases, HMMs are known to work well with sequential or time series data. In this chapter, we conducted experiments using our model with video and speech data sets which are both time series in nature.

4.1 Human Activity Recognition

The outcome of this experiment is to recognize various human activities, cluster the appropriate activities, and check the appropriateness of the clustering process with various available metrics.

The motivation behind choosing this application as our clustering task is to highlight the importance of recognizing human activities in daily life and its applications in various real-life scenarios. Much learning and knowledge can be derived from this task. Its application can be further extended to trending research areas such as human behavior analysis, criminal activity recognition, gait recognition, etc.

We choose two well-known activity recognition data sets, namely: KTH [41] (See Fig. 2) and UCF101 [43] (See Fig. 3). Both data sets contain human activities of different kinds. KTH contains actors performing six types of outdoor activities. Each of these activities is performed by 25 people in a similar background setup captured with a static camera with 25 frames per second (fps). Each image video sequence has a resolution of 160 x 120 pixels with an average length of 4 seconds. In UCF101, we have 101 human actions of 25 categories. Each action category has around 4–7 videos.

For our model to effectively cluster the human activities, we process the video data according to the following experimental setup: First, we extract the frames from the video using video prepossessing techniques, then a feature extraction technique called SIFT [30] is applied to the extracted frames which are images to generate BoVW (Bag of Visual Words).

For this experiment, we consider four actions from the KTH data set: walking, jogging, running, and boxing. From UCF data set, we consider pull-ups, push-ups, swing, and haircut. To begin, a BoVW [47, 49] is generated for each of these actions and fed to the model by combining BoVW for all the actions belonging to one data

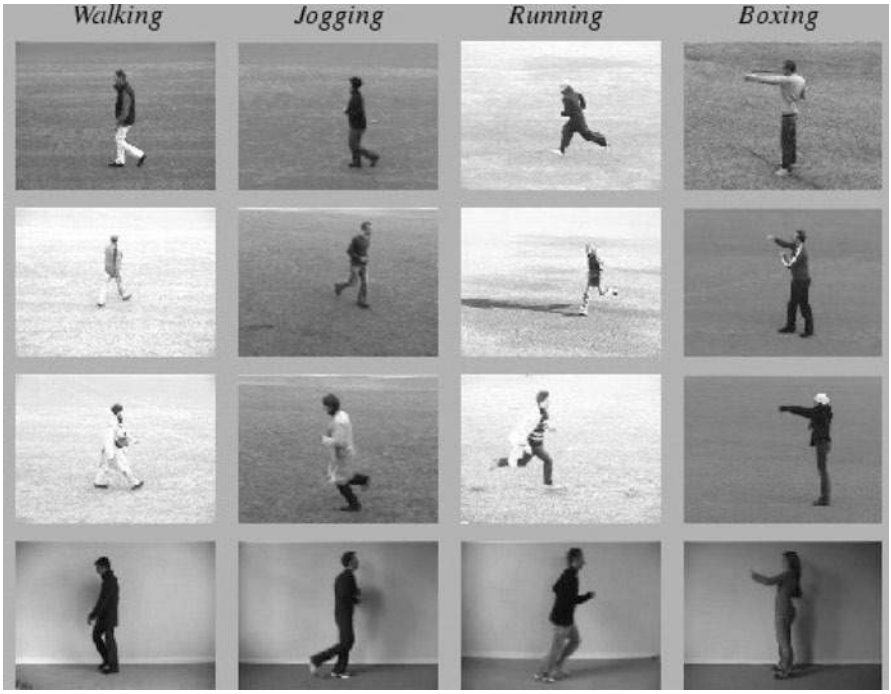


Fig. 2 KTH data set

Fig. 3 UCF data set

set without disturbing the sequence, i.e., avoiding the shuffle. The model is then left to iterate until it converges, that is, until the average value of the latest batch of iterations for the parameters is approximately equal to zero or remains constant. Each iteration computes all seven stages of the MCMC algorithm, including the reversible jump, and the parameters are used to evaluate the model's performance using the four performance metrics listed in Table 1 [8, 10]. Then for the same BoVW, we use HMM with GMM [34] and standard normal distributions as base models to compare our results.

Table 1 Activity recognition with KTH

States	HMM-DMM				HMM-GMM				HMM-std. norm			
K	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
2	49.07	49.07	49.87	49.32	47.28	47.21	47.28	42.79	43.52	28.12	43.92	43.34
3	72.35	72.38	72.35	71.89	64.74	66.65	64.14	69.97	61.71	65.92	61.71	69.54
4	82.92	83.97	82.92	81.28	72.35	72.38	72.35	71.89	66.42	74.16	66.42	53.18

Table 2 Activity recognition with UCF

States	HMM-DMM				HMM-GMM				HMM-std. norm			
K	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
2	54.94	61.30	54.94	67.23	52.55	74.61	52.55	67.79	49.55	47.44	49.55	65.31
3	67.27	79.47	67.27	75.24	62.72	76.15	62.72	72.55	59.27	59.34	59.27	57.40
4	83.22	85.72	83.22	85.18	75.88	78.93	75.88	79.25	68.94	72.62	68.94	61.09

In Tables 1 and 2, we have displayed the relevant results of our model for the learning task of activity recognition. As can be observed, we started the model with two states and left the learning model to figure out the right number of states. After many iterations, it is clearly visible that our model could arrive at the correct number of states with the best parameters and produce better results than the other benchmark models. From Tables 1 and 2, it can be seen that our model score for a number of states gradually increases from a non-optimal number of states to the optimal number of states, with the following accuracies: 49.07 and 72.94, 82.92% and 54.94, 67.27, 83.22%, respectively, for KTH and UCF activities. It can also be observed that our model outperforms the benchmark models in both cases.

4.2 Speaker Recognition

Speaker recognition is the task of automatically detecting the speaker by exploiting the speaker-specific information included in speech waves to validate the identities claimed by persons accessing systems; in other words, it enables voice access control of various services. Voice dialing, banking over a telephone network, telephone shopping, database access services, information and reservation services, voice mail, security control for private information, and remote computer access are all applicable services. Another key use for speech recognition technology is as a forensic tool. Speaker recognition also has several significant advantages over other types of identity identification, such as iris scans, facial recognition, and fingerprint scans. To begin, because all phones have microphones, it is commonly utilized for verification on mobile phones. Second, it is inexpensive to incorporate into other devices like home appliances and automobiles; third, because of the rapid proliferation of IoT devices, it is convenient and familiar to most users. Finally, it has been demonstrated to be extremely accurate in some conditions.

The goal of this experiment is to cluster and identify various voices in a speech sample. In this process, we take several steps to make the speech into a machine-understandable format to be fed to the model.

According to the literature, HMMs have proved their prominence in efficiently processing and clustering speech data on multiple occasions. This is our main motivation to experiment with speech data. Another reason to work with speech data is to showcase the learning efficiency of the model and thereby establish a scope for the model to extend applications to advanced research domains such as emotion detection, speech verification [6, 9, 10] and speech classification [4], automatic speech recognition (ASR) [3, 5, 7], automatic audio transcription [48], etc.

In order to facilitate our experimentation, we have selected a prominent leaders speech data set[1] which contains speeches prominent leaders like Benjamin Netanyahu, Jens Stoltenberg, Julia Gillard, Margaret Thatcher, and Nelson Mandela as folder names. Each audio sample is of one-second 16,000 sample rate PCM encoded.

For this experiment, we have selected four speakers as mentioned in Fig. 4 as part of audio prepossessing for each speaker sample; we employ Mel-frequency cepstral coefficients (MFCCs) [45] for feature extraction and perform voice activity detection (VAD) [42] to eliminate pauses in the speech sample prior to the feature extraction step.

As a result of audio processing for each speaker sample, a feature matrix is obtained and is given as input to our model after excluding the labels for the clustering process without disturbing the sequence of the feature vectors.

The model is then left to iterate until it converges, that is, until the average value of the latest batch of iterations for the parameters is approximately equal to zero or remains constant. Each iteration computes all seven stages of the MCMC algorithm, including the reversible jump, and the parameters are used to evaluate the model's performance using the four performance metrics listed in Table 1. Then for the same feature vector of speech samples, we use HMM with GMM [34] and standard normal distributions as base models to compare our results.

The model is then run for a set number of iterations until it converges, that is, until the average value of the latest batch of iterations for the parameters is about equal or does not change: all seven stages of the MCMC, including the reverse jump component, are computed in each iteration, and the parameters are utilized to verify the model for performance using the four performance metrics shown in Table 3. Then, using the same feature vector of voice samples, we compare our findings using HMM with GMM [34] and HMM with standard normal distributions as base models.

In Table 3, in this speaker recognition learning problem, we have shown the appropriate findings of our model. As can be seen, we started the model with two states and left the learning model to determine the optimal number of states. After

[1] https://www.kaggle.com/kongaevans/speaker-recognition-dataset/version/1.

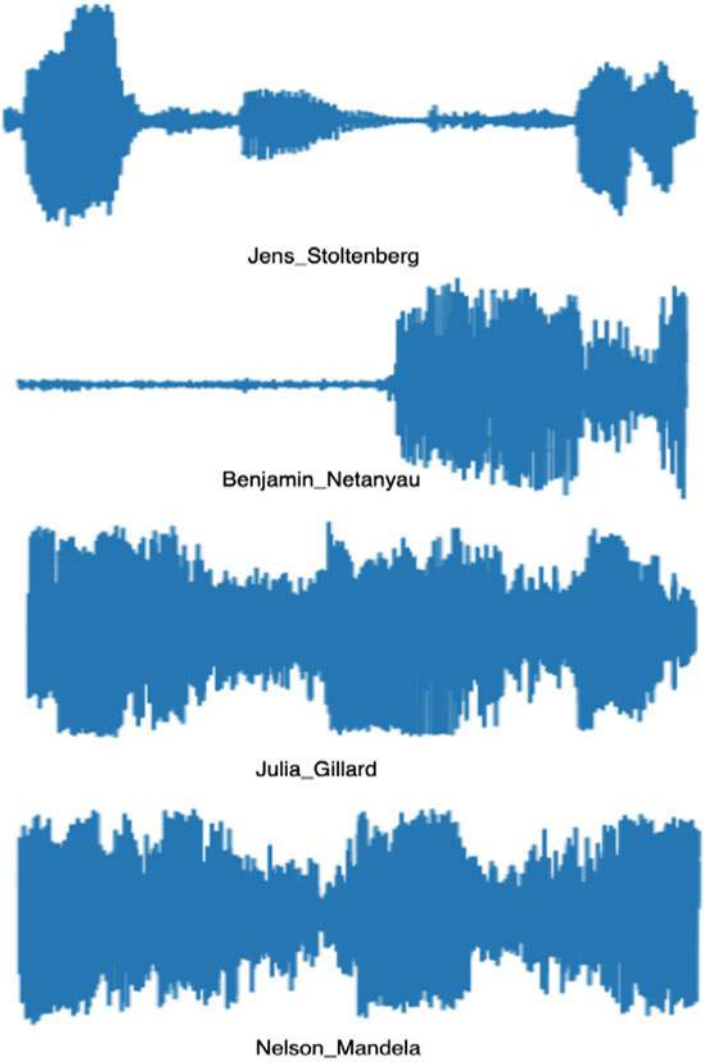


Fig. 4 Speaker speech samples

many iterations, it is evident that our model could reach the proper number of states with the appropriate parameters and produce better outcomes than other benchmark models.

From Table 3, it can be observed that our model performed poorly for non-optimal number of states with low accuracy: 55.07%, precision: 57.28% and we can also notice that the performance of the model gradually improved while approaching the optimal number of states and finally reaching a maximum accuracy and precision of 79.46 and 79.47%, respectively, outperforming the benchmark models.

Table 3 Speaker recognition

States	HMM-DMM				HMM-GMM				HMM-std. norm			
K	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
2	55.07	57.28	55.07	0.08	52.50	53.00	52.50	49.96	51.12	51.29	51.12	46.22
3	64.73	64.61	64.73	55.78	61.91	65.48	61.91	37.13	58.80	58.25	58.80	40.56
4	79.46	79.47	79.46	75.17	71.45	71.38	71.45	67.72	63.93	63.81	63.93	54.19

5 Conclusion

In this chapter, we provide a Bayesian learning framework for HMMs to efficiently learn the model parameters. In addition, we looked into a new reversible jump Markov chain Monte Carlo technique for determining the number of states in HMMs. We use a mixture modeling strategy to boost our model's learning capacity by introducing Dirichlet mixtures. We ran experiments on well-known data sets from video and voice domains to illustrate our model's usefulness in a number of tasks. Throughout this experimentation, we looked at a variety of pre-processing and domain-specific feature extraction algorithms to help with the model's learning process. Furthermore, we validated the learning efficiency of our models by comparing their outcomes and performance using well-known performance metrics, revealing that our models outperformed current benchmark models. We can incorporate a more complex mixture modeling technique to aid in parameter estimation and modeling choices in future studies. In the future, feature selection algorithms could be included to improve the generalization capabilities of the model.

Appendix

In this section, we explain the computation of Jacobian determinant which is a part of the acceptance ratio of split and combine move:

Table 4 presents a Jacobian matrix which has partly block diagonal structure, and our goal is to find out its determinant. For that, first, we identify sub-determinants across the diagonal and evaluate the sub-determinants individually and finally multiply the resultant to obtain the determinant of the whole matrix, and the same is shown below:

Table 4 Table of partial derivatives

	$\tilde{a}_{jj_1}$	$\tilde{a}_{jj_2}$	$\tilde{a}_{j_1 j}$	$\tilde{a}_{j_2 j}$	$\tilde{a}_{ij_2}$	$\tilde{a}_{j_2 j_1}$	$\tilde{\mu}_{j_1}$	$\tilde{\mu}_{j_2}$	$\tilde{\sigma}_{j_1}$	$\tilde{\sigma}_{j_2}$
$\tilde{a}_{ij_*}$	x	x	0	0	0	x	0	0	0	0
v_i	x	x	0	0	0	x	0	0	0	0
$\tilde{a}_{j_* j}$	0	0	x	x	x	x	0	0	0	0
u_j	0	0	x	x	x	x	0	0	0	0
u_0	0	0	x	x	x	x	x	x	x	x
u_1	0	0	0	0	x	x	0	0	0	0
$\tilde{\mu}_{j_*}$	0	0	0	0	0	0	x	x	0	0
z_1	0	0	0	0	0	0	x	x	x	x
$\tilde{\sigma}_{j_*}$	0	0	0	0	0	0	x	x	x	x
z_2	0	0	0	0	0	0	0	0	x	x

$$J_1 = \begin{vmatrix} \operatorname{diag}(v_i) & \operatorname{diag}(1-v_i) \\ \operatorname{diag}\left(\tilde{a}_{ij_*}\right) & -\operatorname{diag}\left(\tilde{a}_{ij_*}\right) \end{vmatrix} = \begin{vmatrix} \boldsymbol{I} & \operatorname{diag}(1-v_i) \\ \boldsymbol{0} & -\operatorname{diag}\left(\tilde{a}_{ij_*}\right) \end{vmatrix} = \prod_{i \neq j_*} \tilde{a}_{ij_*}$$

$$J_2 = \begin{vmatrix} \operatorname{diag}\left(\frac{u_j}{u_0}\right) & \operatorname{diag}\left(\frac{1-u_j}{1-u_0}\right) & -\operatorname{col}\left(\frac{u_1 u_j}{u_0}\right) & \operatorname{col}\left(\frac{(1-u_1)u_j - \sum_{i \neq j_*} v_i \tilde{a}_{ii} \star \partial \gamma_i / \partial \tilde{a}_{j_* j}}{1-u_0}\right) \\ \operatorname{diag}\left(\frac{\tilde{a}_{j_* j}}{u_0}\right) & -\operatorname{diag}\left(\frac{\tilde{a}_{j_* j}}{1-u_0}\right) & -\operatorname{col}\left(\frac{u_1 \tilde{a}_{j_* j}}{u_0}\right) & \operatorname{col}\left(\frac{(1-u_1)\tilde{a}_{j_*}}{1-u_0}\right) \\ -\operatorname{row}\left(\frac{u_j \tilde{a}_{j_* j}}{u_0^2}\right) & \operatorname{row}\left(\frac{(1-u_j)\tilde{a}_{j_* j}}{(1-u_0)^2}\right) & \frac{u_1\left(1-\tilde{\tilde{a}}_{i_1 i_1} - \tilde{\tilde{a}}_{i_1 i_2}\right)}{u_0} & \frac{u_1 + \tilde{\tilde{a}}_{i_2 i_1}}{1-u_0} \\ 0 & 0 & \tilde{\tilde{a}}_{i_1 i_1} + \tilde{\tilde{a}}_{i_1 i_2} & \frac{u_0\left(\tilde{\tilde{a}}_{i_1 i_1} + \tilde{\tilde{a}}_{i_1 i_2}\right)}{1-u_0} \end{vmatrix}$$

$$J_3 = \begin{vmatrix} 1 & 1 & 0 & 0 \\ -\sigma_{j_*}\sqrt{\frac{1-u_0}{u_0}} & \sigma_{j_*}\sqrt{\frac{u_0}{1-u_0}} & \frac{2z_1 u_0}{\sigma_{j_*}^2 z_2 \left(1-z_1^2\right)^2} & \frac{2z_1(1-u_0)}{\sigma_{j_*}^2 (1-z_2)\left(1-z_1^2\right)^2} \\ \frac{z_1 \sigma_{j_*}^3}{2}\sqrt{\frac{1-u_0}{u_0}} & -\frac{z_1 \sigma_{j_*}^3}{2}\sqrt{\frac{u_0}{1-u_0}} & \frac{u_0}{z_2\left(1-z_1^2\right)} & \frac{(1-u_0)}{(1-z_2)\left(1-z_1^2\right)} \\ 0 & 0 & \frac{u_0}{\sigma_{j_*}^2\left(1-z_1^2\right)} & \frac{1-u_0}{\sigma_{j_*}^2 (1-z_2)^2\left(1-z_1^2\right)} \end{vmatrix} = \frac{\sqrt{u_0(1-u_0)}}{\sigma_{j_* z_2^2 (1-z_2)^2 \left(1-z_1^2\right)^3}}$$

J_2 here is evaluated in the same way as shown in [37].

References

1. C. Andrieu, N. De Freitas, A. Doucet, M.I. Jordan, An introduction to mcmc for machine learning. Mach. Learn. **50**(1), 5–43 (2003)
2. T.O. Ayodele, Types of machine learning algorithms. New Adv. Mach. Learn. **3**, 19–48 (2010)
3. M. Azam, N. Bouguila, Unsupervised keyword spotting using bounded generalized Gaussian mixture model with ICA, in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)* (IEEE, Piscataway, 2015), pp. 1150–1154
4. M. Azam, N. Bouguila, Speaker classification via supervised hierarchical clustering using ICA mixture model, in *Proceedings of Image and Signal Processing - 7th International Conference, ICISP 2016, Trois-Rivières, May 30–June 1, 2016*, ed. by A. Mansouri, F. Nouboud, A. Chalifour, D. Mammass, J. Meunier, A. Elmoataz. Lecture Notes in Computer Science, vol. 9680 (Springer, Berlin, 2016), pp. 193–202. https://doi.org/10.1007/978-3-319-33618-3_20

5. M. Azam, N. Bouguila, Blind source separation as pre-processing to unsupervised keyword spotting via an ica mixture model, in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)* (IEEE, Piscataway, 2018), pp. 833–836
6. M. Azam, N. Bouguila, Speaker verification using adapted bounded Gaussian mixture model, in *2018 IEEE International Conference on Information Reuse and Integration, IRI 2018, Salt Lake City, July 6–9, 2018* (IEEE, Piscataway, 2018). https://doi.org/10.1109/IRI.2018.00053
7. M. Azam, N. Bouguila, Bounded generalized Gaussian mixture model with ICA. Neural Process. Lett. **49**(3), 1299–1320 (2019)
8. M. Azam, N. Bouguila, Multivariate bounded support laplace mixture model. Soft Comput. **24**, 1–30 (2020)
9. M. Azam, N. Bouguila, Multivariate-bounded gaussian mixture model with minimum message length criterion for model selection. Exp. Syst. **38**, e12688 (2021)
10. M. Azam, B. Alghabashi, N. Bouguila, Multivariate bounded asymmetric Gaussian mixture model, in *Mixture Models and Applications* (Springer, Berlin, 2020), pp. 61–80
11. H.B. Barlow, Unsupervised learning. Neural Comput. **1**(3), 295–311 (1989)
12. J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, M. West, Generative or discriminative? Getting the best of both worlds. Bayesian Stat. **8**(3), 3–24 (2007)
13. N. Bouguila, T. Elguebaly, A fully bayesian model based on reversible jump MCMC and finite beta mixtures for clustering. Expert Syst. Appl. **39**(5), 5946–5959 (2012). https://doi.org/10.1016/j.eswa.2011.11.122
14. N. Bouguila, W. Fan, *Mixture Models and Applications* (Springer, Berlin, 2020)
15. N. Bouguila, J.H. Wang, A. Ben Hamza, A Bayesian approach for software quality prediction, in *2008 4th International IEEE Conference Intelligent Systems*, vol. 2 (2008), pp. 11–49–11–54. https://doi.org/10.1109/IS.2008.4670508
16. N. Bouguila, J.H. Wang, A.B. Hamza, Software modules categorization through likelihood and Bayesian analysis of finite dirichlet mixtures. J. Appl. Stat. **37**(2), 235–252 (2010)
17. J. Burrell, How the machine thinks': understanding opacity in machine learning algorithms. Big Data Soc. **3**(1), 2053951715622512 (2016)
18. G.A. Churchill, Accurate restoration of dna sequences, in *Case Studies in Bayesian Statistics*, vol. II (Springer, Berlin, 1995), pp. 90–148
19. S.R. Eddy, Hidden Markov models. Curr. Opin. Struct. Biol. **6**(3), 361–365 (1996)
20. Z. Ghahramani, Unsupervised learning, in *Summer School on Machine Learning* (Springer, Berlin, 2003), pp. 72–112
21. P.J. Green, Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika **82**(4), 711–732 (1995)
22. P.J. Green, D.I. Hastie, Reversible jump MCMC. Genetics **155**(3), 1391–1403 (2009)
23. J.D. Hamilton, A new approach to the economic analysis of nonstationary time series and the business cycle. Econ. J. Econ. Soc. **57**, 357–384 (1989)
24. T. Hastie, R. Tibshirani, J. Friedman, Overview of supervised learning, in *The Elements of Statistical Learning* (Springer, Berlin, 2009), pp. 9–41
25. T. Hastie, R. Tibshirani, J. Friedman, Unsupervised learning, in *The Elements of Statistical Learning* (Springer, Berlin, 2009), pp. 485–585
26. S.B. Kotsiantis, I. Zaharakis, P. Pintelas, et al., Supervised machine learning: a review of classification techniques. Emerg. Artif. Intell. Appl. Comput. Eng. **160**(1), 3–24 (2007)
27. A. Labrinidis, H.V. Jagadish, Challenges and opportunities with big data. Proc. VLDB Endowment **5**(12), 2032–2033 (2012)
28. B.G. Leroux, M.L. Puterman, Maximum-penalized-likelihood estimation for independent and Markov-dependent mixture models. Biometrics **48**, 545–558 (1992)
29. S.E. Levinson, L.R. Rabiner, M.M. Sondhi, An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. Bell Syst. Tech. J. **62**(4), 1035–1074 (1983)
30. G. Lowe, Sift-the scale invariant feature transform. Int. J. Comput. Vis. **2**(91–110), 2 (2004)
31. E. Otranto, G.M. Gallo, A nonparametric Bayesian approach to detect the number of regimes in Markov switching models. Econ. Rev. **21**(4), 477–496 (2002)

32. Z. Pawlak, S.K.M. Wong, W. Ziarko, et al., Rough sets: probabilistic versus deterministic approach. Int. J. Man-Mach. Stud. **29**(1), 81–95 (1988)
33. L. Rabiner, B. Juang, An introduction to hidden Markov models. IEEE ASSP Mag. **3**(1), 4–16 (1986)
34. D.A. Reynolds, Gaussian mixture models. Encyclopedia Biom. **741**, 659–663 (2009)
35. S. Richardson, P.J. Green, On Bayesian analysis of mixtures with an unknown number of components (with discussion). J. Royal Stat. Soc. B **59**(4), 731–792 (1997)
36. C. Robert, *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation* (Springer, Berlin, 2007)
37. C.P. Robert, T. Ryden, D.M. Titterington, Bayesian inference in hidden Markov models through the reversible jump Markov chain Monte Carlo method. J. Roy. Stat. Soc. B **62**(1), 57–75 (2000)
38. L. Rokach, O. Maimon, Clustering methods, in *Data Mining and Knowledge Discovery Handbook* (Springer, Berlin, 2005), pp. 321–352
39. T. Rydén, T. Teräsvirta, S. Åsbrink, Stylized facts of daily return series and the hidden Markov model. J. Appl. Econ. **13**(3), 217–244 (1998)
40. S. Sagiroglu, D. Sinanc, Big data: a review, in *2013 International Conference on Collaboration Technologies and Systems (CTS)* (IEEE, Piscataway, 2013), pp. 42–47
41. C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004* (IEEE, Piscataway, 2004). https://doi.org/10.1109/ICPR.2004.1334462
42. J. Sohn, N.S. Kim, W. Sung, A statistical model-based voice activity detection. IEEE Signal Process. Lett. **6**(1), 1–3 (1999)
43. K. Soomro, A.R. Zamir, M. Shah, Ucf101: a dataset of 101 human actions classes from videos in the wild (2012). http://arxiv.org/abs/1212.0402
44. R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, 2018)
45. V. Tiwari, MFCC and its applications in speaker recognition. Int. J. Emerg. Technol. **1**(1), 19–22 (2010)
46. R. Xu, D. Wunsch, *Clustering*, vol. 10. (Wiley, Hoboken, 2008)
47. Y. Yang, S. Newsam, Bag-of-visual-words and spatial extensions for land-use classification, in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2010), pp. 270–279
48. D. Yu, L. Deng, *Automatic Speech Recognition* (Springer, Berlin, 2016)
49. Y. Zhang, R. Jin, Z.H. Zhou, Understanding bag-of-words model: a statistical framework. Int. J. Mach. Learn. Cybern. **1**(1–4), 43–52 (2010)

Online Learning of Inverted Beta-Liouville HMMs for Anomaly Detection in Crowd Scenes

Rim Nasfi and Nizar Bouguila

1 Introduction

Data categorization is rapidly becoming one of the most important parts of data analysis, particularly with the exponential growth of data under all sorts of formats. Thereby, it is crucial to study and discover hidden patterns in order to extract valuable information promoting accurate and solid decision making.

When modeling data, it is a notable fact that Gaussian mixture models (GMMs) are not always the perfect solution for all data types. Through HMM deployment, most existing related works have not considered the characteristics of data sets. In fact, most of the work present in the literature relies on the use of Gaussian distributions. Although HMMs were mainly developed for discrete and Gaussian data [39], diversity of applications in contexts and domains such as activity recognition, image categorization, and dynamic forecasting, increased the necessity of modifying the underlying HMM model so that it efficiently suits those new data types [37, 42].

Thanks to the proliferation of carried research on these distributions and their mathematical simplicity, most finite mixture models mainly consider Gaussian as their basic distributions. Nevertheless, it is undeniable that the least appropriate way of modeling non-Gaussian data is to use Gaussian distributions [11]. For example, inverted Dirichlet or generalized inverted Dirichlet [5, 18, 20] can often outperform the Gaussian mixture model for modeling positive vectors in many applications such as image categorization, human action video recognition, etc. Recently, numerous works have been achieved in order to model positive vectors based on inverted Dirichlet mixture models [5, 37]. However, the inverted Dirichlet distribution has

R. Nasfi (✉) · N. Bouguila
Concordia Institute for Information Systems Engineering, Montreal, QC, Canada
e-mail: r_nasfi@encs.concordia.ca; nizar.bouguila@concordia.ca

N. Bouguila et al. (eds.), *Hidden Markov Models and Applications*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-99142-5_7

a very restraining covariance structure that significantly limited its flexibility. In our work, we propose to model positive vectors based on a finite mixture model with inverted Beta-Liouville (IBL) distributions embedded into the framework of HMMs as emission probabilities.

Inverted Beta-Liouville mixture models have recently arisen as an efficient way to model positive vectors [10]. Thanks to its general covariance structure and its smaller number of parameters compared to the inverted Dirichlet and generalized inverted Dirichlet [6, 34], IBL has proven its effectiveness when dealing with positive vectors modeling [28]. Originally derived from the Liouville distributions family [26]. As earlier mentioned, one of the main advantages of the IBL is its general covariance structure that can either be positive or negative. It is noteworthy to mention that the discussed distribution has not been extensively investigated and that only a handful of works have adopted it, giving more room to further exploitation of this surprisingly underrated distribution. Even more effectively, this choice is mainly motivated by the fact that the IBL distribution contains inverted Dirichlet distribution as a special case and therefore can provide more flexibility compared to previously investigated distributions [37]. Also, compared with Gaussian that can only approximate symmetric distributions, IBL allows both symmetric and asymmetric distributions.

The work presented in this manuscript can be viewed intellectually at two different levels. First, it allows the application, for the first time to the best of our knowledge, of IBL-based HMMs to effectively handle positive vectors; second, it proposes to undertake online-based learning of parameters by applying an online EM procedure for HMMs.

The remainder of this chapter is organized as follows: In Sect. 2, we present some of the work related to online and incremental learning, and we discuss the choice of application in this paper. Section 3 presents HMMs, their formulation, and the online EM derivations. Section 4 explains the choice of the IBL mixture models and details derivations and parameters estimation. Then, in Sect. 5 we present our applied model as well as results and interpretations. Finally, we conclude with some insights and future work perspectives.

2 Related Work

The performance of hidden Markov models (HMMs) is often acclaimed through their massive use in several complex real-world applications namely image categorization [21], action recognition [22], occupancy estimation in smart buildings [37], and unusual events detection [20]. HMMs are highly capable of representing probability distributions corresponding to these complex real-world phenomena when they are fed an adequate number of states as well as a sufficiently rich set of data. Nevertheless, the mentioned applications tend to often drain HMMs' performance particularly when results need to be inferred from very long sets of data such as videos in an action recognition context. In fact, in the context of

HMMs, analyzing large sets of training data is costly, laborious, and long sustained. Thus, there is often not enough analyzed data to be representative of the underlying distribution, causing the HMM to incorporate some uncertainty.

In such cases, it is suitable to update the model's parameters online [44]. Online learning of new data sequences permits the adaptation of the HMM parameters while new data becomes available. This way of feeding data to the model is also called an incremental method. It is actually common for a model to be fed additional data after its training. This allows for a more adaptation of HMMs as a result of newly acquired data. Therefore, incremental learning is an undeniable asset to refine HMMs' behavior toward any novelties encountered in the environment and thus reducing their level of uncertainty by maintaining a high level of performance.

When applying incremental learning for HMM parameters estimation, there are commonly standard techniques used that mostly involve batch learning. Those techniques can either rely on specialized EM techniques [16] such as Baum–Welch (BW) algorithm [4] or on numerical optimization techniques such as the Gradient Descent algorithm [33], where regardless of the used technique, parameters are estimated after numerous training repetitions prior to maximizing an objective function over certain independent validation data. In most cases, when applying a batch learning technique, a fixed-length sequence $O = o_1, o_2, .., o_T$ of T training observations, o_i is hypothetically available during the whole learning process. If we suppose that O is assembled into a block D of training data, each training iteration involves observing all sub-sequences in D prior to updating HMM parameters. When a new block of data comes through, the previously trained HMM cannot accommodate the second batch without accumulating and storing all the training data in memory. It will eventually train again for the beginning making use of all the cumulative data involving both batches. This procedure is deemed to be necessary in order to prevent any sort of corruption of the previously acquired knowledge, and that could compromise the HMM performance. Notwithstanding, there are clearly some significant costs relating to processing time and storage requirements when using batch learning methods. Time and memory complexity would grow linearly with the length and number of training observation sequences and quadratically with the number of HMM states.

As a viable alternative, numerous online learning techniques have been proposed in the literature; this includes techniques based on EM [13, 36] where numerical optimization and recursive estimations are performed, and EM variants such as BOEM (Block Online EM) [32]. These methods assume the observation is a stream of data and are particularly used in situations where training symbols are organized into a block of one or more sub-sequences. Their parameters are re-estimated upon observing each new sub-sequence of symbols. Some of the aforementioned techniques are tailored to update HMM parameters at a symbol level, also perceived as recursive or sequential estimation techniques. Symbol-wise updated techniques are designed for situations in which training symbols are received one at a time where parameters are then re-estimated upon observing each new symbol. Across the full range of contexts, HMMs parameters are updated from new training data, beyond any requirement for access to the formerly learned training data and most

plausibly preventing corruption of any previously acquired knowledge [14]. In this manner, the main takeaway of the stated techniques is essential to allow sustaining a high level of performance while preserving the memory requirements, given the fact that storing data from previous training phases is completely unnecessary. Besides, bearing in mind that training is only performed on the new training sequences, and not all accumulated data, online learning also provides lower time complexity when learning new data. In this work, we aim to study the effectiveness of online-based HMMs compared to standard-learning-based HMMs when used along with a remarkably interesting distribution, that is the Inverted Beta-Liouville, as emission probabilities.

Further to the raised interest in online learning as a technical concern, the studies carried out in this work revolve around analyzing human-related visual data. We choose to bring a special focus on disclosing information from looking at videos with humans doing certain activities and analyzing, in particular, security surveillance to predict certain anomalies. Indeed, it would be of great help to assist in detecting either normal or abnormal events or behaviors and use this as a starting point to make decisions such as in the contexts of smart cities where there is a growing need to improve security. In fact, this can be achieved by quickly and accurately identifying criminal activities in a real-time fashion [15]. Similarly, in an entertainment environment, activity recognition can notably improve users' experience by automatically recognizing different player's actions during a game of tennis or a soccer game for example [31, 41], with the goal of understanding the action of each player and how they interact with each other.

What is challenging in performing this type of analysis, is that crowded scenes and dynamic environments are bound to a degraded performance as soon as the crowd becomes too dense [19]. In fact, the number of independent objects moving at the same time and the occlusions it involves degrades the performance of detection. Additionally, the dynamic background is an important restriction when it comes to tracking movements.

As far as HMMs are concerned, modeling normal scenes and determining whether an unseen video sequence deviates from normality is an achievable task, which serves perfectly the anomaly detection aim. In the work of Bettini et al. in [7], the features used are histograms that can be seen as positive vectors once extracted. The likelihood criterion for anomaly detection is somehow efficient despite the simple adaptive threshold adopted by the classifier. The work is obviously relying on different processes leading together to detection results, which constitutes a clear limitation to the improvement of the global approach and the use of a standardized, unique model, capable of providing a more compact representation of the data and thus a more accurate anomaly detection. In a related context, the author in [47] exploits the notion of profiling an online anomaly sampling to model dynamic scenes in a way that optimizes the intrusion detection rate by refraining from using any manual labeling of the training data set. The method relies on a Dynamic Bayesian Network (DBN) to model each behavior pattern. Further, an online Likelihood Ratio Test (LRT) method is used to detect abnormal behavior, while normal behavior is recognized when sufficient visual evidence is available. The mentioned

procedure lacks accuracy since some events can sometimes be undetected by the model due to missing visual evidence or ambiguities between event classes. This can be avoided by taking the temporal information into consideration by developing a Baum–Welch EM algorithm to the mixture of DBNs to learn the behavior model directly rather than taking a phased approach such as the one adopted. Andrea et al. [1–3] used HMMs with Gaussian mixtures to characterize the normal behavior of a crowd by learning normal motion patterns from the optical flow of image blocks. The method relied mainly on Principal Component Analysis (PCA) to build feature prototypes, along with spectral clustering to find the optimal number of models to group video segments containing similar motion patterns. An HMM was trained for each model and used for event recognition and anomaly detection.

3 Hidden Markov Models

Hidden Markov Models are described according to Ghahramani [24], as an ubiquitous tool to model time series data. They have been used for decades in speech recognition systems as well as artificial intelligence and pattern recognition applications. These models are a generalization of mixture models [25]. In fact, the probability density functions overall observable states defined by an HMM are considered as a mixture of densities defined by each state.

HMMs allow us to represent probability distributions over sequences of observations, with the assumption that observations are discrete. An observation at time t is denoted by the variable O_T.

Hidden Markov Models are governed by two main properties. First, it assumes that the observation at time t is generated by some process whose state h_t is hidden from the observer. Second, it assumes that the state of this hidden process satisfies the Markov property; that is, given the value of h_{t-1}; the current state h_t is independent of all the states prior to the time $t-1$.

A hidden Markov model is characterized by a set of parameters that will be specified later in this paper. The task of the learning algorithm is to find the best set of state transitions and emission probabilities between the states of the model. Therefore, an output sequence or a set of these sequences is given. To illustrate our model, we are first listing various HMM notations and enumerating the upcoming used work script.

3.1 Notations and Offline EM for HMMs

We consider a HMM with continuous emissions and K states. We put $y = \{y_0, y_1, \ldots, y_T\}$ the sequence of observed data with $y_t \in \mathbb{R}^L$. The observation for the l-th feature at time t, which is represented by the l-th component of y_t, is denoted by y_{lt}.

Let $x = \{x_0, x_1, \ldots, x_T\}$ be the sequence of hidden data. The transition matrix of the Markov chain associated with this sequence is denoted as $B = \{b_{ij} = P(x_t = j|x_{t-1} = i)\}$ and π is the initial state probability. Thus the complete-data likelihood can be expressed as

$$p(x, y|\Lambda) = \pi_{x_0} c_{x_0}(y_0) \prod_{t=1}^{T} b_{x_{t-1},x_t} c_{x_t}(y_t) \tag{1}$$

where Λ is the set of model parameters, π_{x_0} is the initial state (x_0) probability, and $c_{x_t}(y_t)$ is the emission probability given state x_t.

The M-step aims to maximize the data log-likelihood. By denoting Z as hidden variables and X as the data, we can express the data likelihood $\mathscr{L}(\theta|X) = p(X|\theta)$ by

$$\begin{aligned} E(X, \theta) - R(Z) &= \sum_{Z} p(Z|X) \log(p(X, Z)) - \sum_{Z} p(Z|X) \log(p(Z|X)) \\ &= \sum_{Z} p(Z|X) \log(p(X|\theta)) \\ &= \log(p(X|\theta)) \sum_{Z} p(Z|X) \log(p(X|\theta)) \\ &= \log(p(X|\theta)) = \mathscr{L}(\theta|X) \end{aligned} \tag{2}$$

with θ representing all the HMM parameters, $E(X, \theta)$ is the value of the complete-data log-likelihood with the maximized parameters θ, and $R(Z)$ is the log-likelihood of the hidden data given the observations.

The expected complete-data log-likelihood is

$$E\left(X, \theta, \theta^{old}\right) = \sum_{Z} p(Z|X, \theta^{old}) \log(p(X, Z|\theta)) \tag{3}$$

In the following, we take the case of a unique observation sequence, X, then the complete-data likelihood is expanded as

$$p(X, Z|\theta) = p(h_0) \prod_{t=0}^{T-1} p(h_{t+1}|h_t) \prod_{t=0}^{T} p(m_t|h_t) p(x_t|h_t, m_t) \tag{4}$$

When considering an HMM, as defined earlier in this section, where the final time T may be unbounded in the online case, offline learning consists of adjusting the model parameters to maximize the likelihood of a given training sequence $y_{0 \longrightarrow T}$. This procedure results in the following update equations that can be reviewed in

detail in a previous work [37]:

$$\hat{b}_{ij}^{(n+1)} = \frac{\sum_{t=1}^{T} P(x_{t-1} = i, x_t = j | y_{0 \longrightarrow T}, \hat{\theta}_n)}{\sum_{t=1}^{T} P(x_{t-1} = i | y_{0 \longrightarrow T}, \hat{\theta}_n)} \tag{5}$$

$$\hat{c}_{jk}^{(n+1)} = \frac{\sum_{t=1}^{T} P(x_t = j, y_t = k | y_{0 \longrightarrow T}, \hat{\theta}_n)}{\sum_{t=1}^{T} P(x_t = j | y_{0 \longrightarrow T}, \hat{\theta}_n)} \tag{6}$$

where $k = 1, \ldots, K$ and the probabilities on the right-hand side are conditioned on the training sequence $y_{0 \longrightarrow T}$ and on the current parameters' estimate $\hat{\theta}_n \equiv (\{\hat{b}_{ij}^{(n)}\}, \{\hat{c}_{jk}^{(n)}\})$. Computation of these quantities can be done efficiently using the forward–backward procedure, although this will imply storing the whole training sequence.

3.2 *Online EM for HMMs*

Online learning has proven to be an effective way to improve learning, mainly in large-scale settings [9, 36]. In this work, we build upon the work presented by Mongillo et al. in [36] and Cappé in [13], to put forward an online and incremental EM algorithm for HMMs. For the matter, a recall of Cappés' online EM is desired. The latter uses a stochastic approximation approach in the scope of sufficient statistics in order to achieve a limiting EM recursion. This EM recursion is nothing but a batch-based EM algorithm with infinite data. All the parameter updates are handled in a recursive manner. This procedure is built around a forward-only smoothing recursion, in which the expected sufficient statistics needed for parameter updates are computed recursively. This can be achievable thanks to an expectation–maximization algorithm that updates and improves lower bounds on the likelihood after each observation.

In this phase, we focus on calculating the likelihood of an observation sequence of a given length to classify it. After determining the sequence category, we use the corresponding data to train a specific HMM and use its parameters to update the previously trained HMM corresponding to the said category.

The adopted method consists of applying the online EM developed in [36], which we expand to handle positive vector modeling thanks to the adoption of IBL mixtures as emission probabilities.

We here derive a version of the EM procedure that does not require the storage of the inputs by reproducing the EM update (Eqs. 5 and 6) in terms of sufficient statistics updated recursively

3.2.1 Sufficient Statistics for Parameter Estimation

The required sufficient statistics are

$$\phi_{ijk}(T;\theta) = \frac{1}{T}\sum_{t=1}^{T}\delta(y_t - k)\cdot P(x_{t-1}=i, x_t = j | y_{0\longrightarrow T}, \theta) \tag{7}$$

with $1 \leq i, j \leq K$ and $1 \leq k \leq M$, where $\delta(\cdot)$ is the Kronecker delta: 1 when its argument is 0 and 0 otherwise. The prefactor $\frac{1}{T}$ ensures that $\phi_{ijk}(T;\theta)$ do not diverge for an infinitely long training sequence $T \longrightarrow \infty$.

The update equations can thus be written as follows:

$$\hat{b}_{ij}^{(n+1)} = \frac{\sum_k \phi_{ijk}(T;\hat{\theta}_n)}{\sum_{jk}\phi_{ijk}(T;\hat{\theta}_n)} \tag{8}$$

$$\hat{c}_{jk}^{(n+1)} = \frac{\sum_i \phi_{ijk}(T;\hat{\theta}_n)}{\sum_{ik}\phi_{ijk}(T;\hat{\theta}_n)} \tag{9}$$

3.2.2 Recurrence Relations

$$\phi_{ijk}^{\gamma}(T) = \frac{1}{T}\sum_{t=1}^{T}\delta(y_t - k)\cdot P(x_{t-1}=i, x_t = j, x_T = \gamma | y_{0\longrightarrow T}) \tag{10}$$

where we drop the explicit independence on the model parameters θ assumed to be constant and hence $\sum_\gamma \phi_{ijk}(T) = \phi_{ijk}(T)$. We can then write

$$\begin{aligned} P(x_{t-1}=i, x_t = j, x_{T-1}=\zeta, x_T = \gamma, y_{0\longrightarrow T}) &= P(y_T|x_T=\gamma) \\ \times\, P(x_T = \gamma | x_{T-1}=\zeta) P(x_{t-1}=i, x_t = j, x_T = \zeta, y_{0\longrightarrow T-1}) & \end{aligned} \tag{11}$$

where we used the product rule and the dependency conditions. Dividing both sides by $P(y_{0\longrightarrow T-1})$ and summing over ζ we get

$$\begin{aligned} &P(x_{t-1}=i, x_t = j, x_T = \gamma | y_{0\longrightarrow T}) \\ &\quad = \sum_\zeta \eta_{\zeta\gamma}(y_T)\cdot P(x_{t-1}=i, x_t = j, x_{T-1}=\zeta | y_{0\longrightarrow T-1}) \end{aligned} \tag{12}$$

with

$$\eta_{\zeta\gamma}(y_T) \equiv \frac{P(y_T|x_T=\gamma)P(x_T=\gamma|x_{T-1}=\zeta)}{P(y_T|y_{0\longrightarrow T-1})} \tag{13}$$

Equation 13 inserted into Eq. 10 provides the following recurrence relation for the ϕ_{ijk}^{γ}:

$$\phi_{ijk}^{\gamma}(T) = \frac{1}{T} \cdot \delta(y_T - k) \cdot \eta_{ij}(y_T) \cdot P(x_{T-1} = j | y_{0 \longrightarrow T-1}) + \frac{1}{T} \sum_{t=1}^{T-1} \delta(y_t - k) \cdot \sum_{\zeta} \eta_{\zeta\gamma}(y_T) \cdot P(x_{t-1} = i, x_t = j, x_{T-1} = \zeta | y_{0 \longrightarrow T-1}) \tag{14}$$

by changing the order of summation we can write the second term on the right-hand side of the equation as

$$\frac{1}{T} \sum_{t=1}^{T-1} \delta(y_t - k) \cdot \sum_{\zeta=1} \eta_{ij}(y_T) \cdot P(x_{t-1} = i, x_{T-1} = \zeta | y_{0 \longrightarrow T-1}) = \left(1 - \frac{1}{T}\right) \sum_{\zeta} \eta_{\zeta\gamma}(y_T) \cdot \phi_{ijk}^{\zeta}(T-1) \tag{15}$$

Finally by inserting Eq. (15) into Eq. (14) and changing terms order we obtain

$$\phi_{ijk}^{\gamma}(T) = \sum_{\zeta} \eta_{\zeta\gamma}(y_T) \times \phi_{ijk}^{\zeta}(T-1) + \frac{1}{T}[\delta(y_T - k) \cdot g_{ij}(\zeta, \gamma) \cdot \omega_{\zeta}(T-1) - \phi_{ijk}^{\zeta}(T-1)] \tag{16}$$

with $g_{ij}(\zeta, \gamma) \equiv \delta(i-\zeta) \cdot \delta(j-\gamma)$, and $\omega_{\zeta}(T-1) \equiv P(x_{T-1} = \zeta | y_{0 \longrightarrow T-1})$, which can be computed recursively, and $\eta_{\zeta\gamma}(y_T)$ is expressed in terms of the model's parameters as

$$\eta_{\zeta\gamma}(y_T) = \frac{b_{\zeta\gamma} c_{\gamma, y_T}}{\sum_{m,k} b_{m,k} c_{k, y_T} \omega_m(T-1)} \tag{17}$$

with c_{γ, y_T} is the probability of emitting an output y_T in state γ , that is $c_{\gamma, y_T} \equiv \sum_k c_{\gamma k} \cdot \delta(y_T - k)$

4 Inverted Beta-Liouville Mixture Model

We suppose a D-dimension vector $\mathbf{X} = (X_1, \ldots, X_D)$ is drawn from an inverted Beta-Liouville distribution [23], then we have

$$p(\mathbf{X}|\alpha_d, \ldots, \alpha_d, \alpha, \beta, \lambda) = \frac{\Gamma\left(\sum_{d=1}^{D} \alpha_d\right) \Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \prod_{d=1}^{D} \frac{X_d^{\alpha_l - 1}}{\Gamma(\alpha_d)} \times \lambda^{\beta} \left(\sum_{d=1}^{D} X_d\right)^{\alpha - \sum_{d=1}^{D} \alpha_d} \left(\lambda + \sum_{d=1}^{D} X_d\right)^{-(\alpha+\beta)} \tag{18}$$

where $X_d > 0$ for $d = 1, \ldots, D$, $\alpha > 0, \beta > 0$ and $\lambda > 0$. In fact, the IBL distribution can be viewed as a generalized form of the inverted Dirichlet distribution that involves multiple symmetric and asymmetric modes. The mean, variance, and covariance of the IBL distribution are given by

$$E(X_d) = \frac{\lambda\alpha}{\beta - 1} \frac{\alpha_d}{\sum_{d=1}^{D} \alpha_d} \tag{19}$$

$$Var(X_d) = \frac{\lambda^2\alpha(\alpha + 1)}{(\beta - 1)(\beta - 2)} \frac{\alpha_d(\alpha + 1)}{\sum_{d=1}^{D} \alpha_d (\sum_{d=1}^{D} \alpha_d + 1)} - \frac{\lambda^2\alpha^2}{(\beta - 1)^2} \frac{{\alpha_d}^4}{(\sum_{d=1}^{D} \alpha_d)^4} \tag{20}$$

$$Cov(X_m, X_n) = \frac{\alpha_m \alpha_n}{\sum_{d=1}^{D}} \left[\frac{\lambda^2\alpha(\alpha + 1)}{(\beta - 1)(\beta - 2)(\sum_{d=1}^{D} \alpha_d + 1)} - \frac{\lambda^2\alpha^2}{(\beta - 1)^2(\sum_{d=1}^{D} \alpha_d)} \right] \tag{21}$$

If a set of data contains N vectors: $\mathscr{X} = \{\mathbf{X_1}, \ldots, \mathbf{X_N}\}$, where each $\mathbf{X_i} = (X_{i1}, \ldots, X_{iD})$ is drawn from the IBL mixture model with M components and is defined as follows:

$$p(\mathbf{X_i}|\boldsymbol{\pi}, \Theta) = \sum_{j=1}^{M} \pi_j \, p(\mathbf{X_i}|\theta_j) \tag{22}$$

where $\Theta = (\theta_1, \ldots, \theta_M)$, $p(\mathbf{X_i}|\theta)$ denotes the IBL distribution in Eq. (18) associated with the jth component with parameters $\theta_j = (\alpha_{j1}, \ldots, \alpha_{jD}, \alpha_j, \beta_j, \lambda_j)$, and
$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_M)$ are the mixing coefficients where $0 \leq \pi_j \leq 1$ and $\sum_{j=1}^{M} = 1$.

Maximum Likelihood Estimation

In order to learn the models' parameters, we choose a learning approach based on Maximum Likelihood (ML). The values of different parameters are obtained by maximizing the log-likelihood function such as

$$\widetilde{\Theta} = \underset{\Theta}{\operatorname{argmax}} \log p(\mathscr{X}|\boldsymbol{\pi}, \Theta) \tag{23}$$

where the log-likelihood function is given by

$$\begin{aligned} \mathscr{L}(\mathscr{X}|\boldsymbol{\pi}, \Theta) &= \log p(\mathscr{X}|\boldsymbol{\pi}, \Theta) = \log \prod_{i=1}^{N} p(\mathbf{X_i}|\boldsymbol{\pi}, \Theta) \\ &= \sum_{i=1}^{N} \log \left(\sum_{j=1}^{M} \pi_j \, p(\mathbf{X_i}|\theta_j) \right) \end{aligned} \tag{24}$$

We define latent variables as indicators for a set of observed data. Let $\mathscr{Z} = \{\mathbf{Z_1}, \ldots, \mathbf{Z_N}\}$, each $\mathbf{Z_i} = (Z_{i1}, \ldots, Z_{iM})$ corresponds to an observed data vector $\mathbf{X_i}$, where $Z_{ij} \in \{0, 1\}$ and $\sum_{j=1}^{M} Z_{ij} = 1$, and $Z_{ij} = 1$ if $\mathbf{X_i}$ belongs to component j, and 0 otherwise. The log-likelihood of the complete data can thus be expressed as follows:

$$\Phi(\mathscr{X}, \mathscr{Z}|\boldsymbol{\pi}, \Theta) = \sum_{i=1}^{N} \sum_{j=1}^{M} Z_{ij} \left\{ \log \pi_j + \log p(\mathbf{X_i}|\theta_j) \right\} \tag{25}$$

Next, the conditional expectation of the complete-data log-likelihood is maximized in the M-step of the EM algorithm, which is given by

$$\Omega(\mathscr{X}|\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{M} \langle Z_{ij} \rangle \left\{ \log \pi_j + \log p(\mathbf{X_i}|\theta_j) \right\} \tag{26}$$

with the posterior probability $\langle Z_{ij} \rangle$ being the expected value of the indicator variable and is given by

$$\langle Z_{ij} \rangle = \frac{\pi_j \, p(\mathbf{X_i}|\theta_j)}{\sum_{k=1}^{M} \pi_k \, p(\mathbf{X_i}|\theta_k)} \tag{27}$$

We maximize the conditional expectation of the complete-data log-likelihood by computing the first derivatives with respect to all parameters

$$\frac{\partial \Omega(\mathscr{X}|\Theta)}{\partial \alpha_j} = \sum_{i=1}^{N} \langle Z_{ij} \rangle \left[\log \sum_{d=1}^{D} X_{id} - \log(\lambda_j + \sum_{d=1}^{D} X_{id}) \right] + [\Psi(\alpha_j + \beta_j) - \Psi(\alpha_j)] \sum_{i=1}^{N} \langle Z_{ij} \rangle \tag{28}$$

$$\frac{\partial \Omega(\mathscr{X}|\Theta)}{\partial \beta_j} = \sum_{i=1}^{N} \langle Z_{ij} \rangle \left[\log \lambda_j - \log(\lambda_j + \sum_{d=1}^{D} X_{id}) \right] + [\Psi(\alpha_j + \beta_j) - \Psi(\beta_j)] \sum_{i=1}^{N} \langle Z_{ij} \rangle \tag{29}$$

$$\frac{\partial \Omega(\mathscr{X}|\Theta)}{\partial \alpha_{jd}} = \sum_{i=1}^{N} \langle Z_{ij} \rangle \left[\log X_{id} - \log \sum_{d=1}^{D} X_{id} \right] + [\Psi(\sum_{d=1}^{D} \alpha_{jd} - \Psi(\alpha_{jd})] \sum_{i=1}^{N} \langle Z_{ij} \rangle \tag{30}$$

$$\frac{\partial \Omega(\mathscr{X}|\Theta)}{\partial \lambda_j} = \sum_{i=1}^{N} \langle Z_{ij} \rangle \left[\frac{\beta_j}{\lambda_j} - \frac{\alpha_j + \beta_j}{\lambda_j + \sum_{d=1}^{D} X_{id}} \right] \tag{31}$$

with $\Psi(.)$ being the digamma function. It is obvious that a closed-form solution for θ_j does not exist. Thus, to estimate these parameters, we use the Newton–Raphson method [38] such as

$$\theta_j^{(t+1)} = \theta_j^{(t)} - H(\theta_j^{(t)})^{-1} \frac{\partial \Omega(\mathscr{X}|\boldsymbol{\pi}^{(t)}, \Theta^{(t)})}{\partial \theta_j^{(t)}} \tag{32}$$

where $H(\theta_j^{(t)})^{-1}$ represents the inverse Hessian matrix for parameter θ_j and is described in detail in [28].

4.1 Online Update for the Sufficient Statistics and Model Parameters

To set up an online EM, we start with an initial guess for the model parameters $\hat{\theta}(0)$, the initial state probabilities, $\omega_\zeta \equiv P(x_0 = \zeta)$, and the sufficient statistics, $\hat{\phi}_{ijk}^{\gamma}(0)$.

After removing the contribution of the sufficient statistics such as performed in [36], state estimates, $\omega_\zeta(T)$, which represent the probability of being in the state ζ at time T are then expressed such as

$$\hat{\omega}_\zeta(T) = \sum_m \eta_{m\zeta}(y_T; \hat{\theta}(T-1)) \cdot \hat{\omega}_m(T-1) \tag{33}$$

Finally, the parameters are re-estimated according to the following equations:

$$\hat{b}_{ij}(T) = \frac{\sum_k \sum_\zeta \phi_{ijk}^\zeta(T)}{\sum_{j,k} \sum_\zeta \phi_{ijk}^\zeta(T)} \tag{34}$$

$$\hat{c}_{jk}(T) = \frac{\sum_i \sum_\zeta \phi_{ijk}^\zeta(T)}{\sum_{i,k} \sum_\zeta \phi_{ijk}^\zeta(T)} \tag{35}$$

5 Experiments and Results

In this section, extensive experiments are conducted and we have implemented several real-world topical yet challenging applications using the online HMM with IBL emission probabilities. We are mainly comparing our new approach to its classical online Gaussian-based HMMs competitors and other new adaptations that we executed for the sake of comparison and testing, e.g., inverted Dirichlet-based online HMM (Online ID-HMM) and Dirichlet-based online HMM (Online Dir-HMM). It is noteworthy that the learning of the mentioned adaptations has been based on the same methodology described in the previous section to learn the IBL mixture-based HMM. Real-world applications on two video data sets, an anomaly in a crowd context and direction-related anomaly detection in an airport, are tested to validate the performance of our model.

Recognition of human action in videos gained a great deal of attention thanks to the multitude of applications in many domains such as human–computer interfaces, video surveillance [40, 49], and activity biometry [17]. Applications involve but are not limited to, detecting violence, hostile behavior, and sexual harassment [45], not to mention life-threatening events such as pedestrians accidents, criminality [43].

It is worthwhile to mention that dealing with crowded scenes analysis often involves a sizable amount of individuals acquiring irregular directions in an exceedingly vast region hence the complexity of the task. Anomalies or abnormal events can be intuitively defined as any occurrence of a deviation from the conventional crowd behavior in an exceedingly vast video. Moreover, an anomaly could eventually be a pattern that does not follow expected traditional behavior in a given context.

Hidden Markov Models are indeed an appropriate tool to tackle this problem since they are particularly suitable when working with dynamic data such as videos, and attempting to unveil unknown natures of anomalies.

5.1 Anomaly Detection in a Crowd of Pedestrians

The main goal of this experiment is to detect any anomalies in the surveillance video of the publicly available UCSD Ped1 and Ped2 data set [29]. Both data sets are formed from video sequences of pedestrians on a walkway and divided into a training set, with normal frames only, and a testing set composed of both normal and abnormal frames. These two data sets only differ in the camera viewpoint from which footage has been captured. We still are able to benefit from ground truth, provided for all test sequences. Sample frames from the training set with different crowd densities and anomalies are presented in Figs. 1 and 2. In the following we proceed to the feature extraction in a procedure we describe briefly (see [19] for further details).

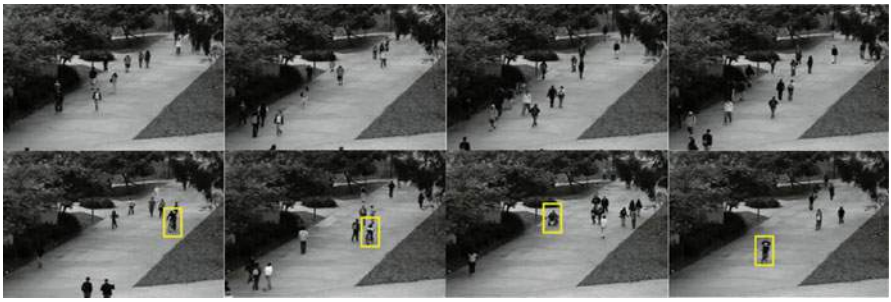


Fig. 1 Frames from the Ped1 normal (upper row) and abnormal activities (bottom row) with anomalies highlighted

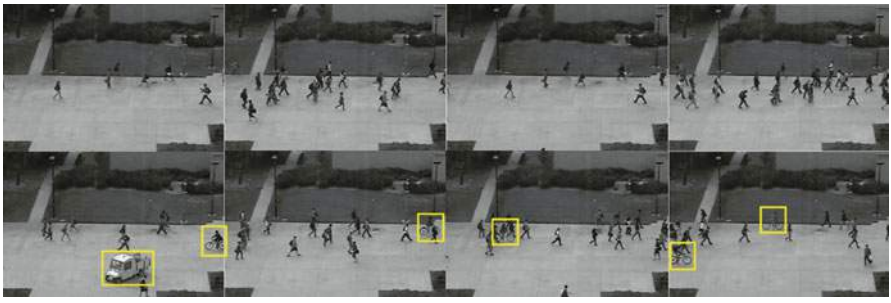


Fig. 2 Frames from the Ped2 normal (upper row) and abnormal activities (bottom row) with anomalies highlighted

The pre-processing involves a gray-level re-sampling of the frames to a size of 160×240 pixels, with a filter-based Gaussian noise reduction where the filter size is $[3, 3]$ and $\sigma = 1.1$. Next, we perform some dimensional sampling steps in order to meet HMMs use requirements such as lowering the length of our histograms (12 in this application), small volumes extracted from sequences, called *cuboids*, are repeated several times in a sequence in order to avoid model overfitting. The dimensional sampling adopted here consists in dividing sequences into cuboids each of them subdivided into 8 subregions, 2 along each direction. Pixels' contribution within a subregion is weighted by its magnitude and is computed in the same fashion as in [19]. We model each cuboid by a series of 8 normalized histograms through which a dynamic mechanism embedded in each cuboid is illustrated. An HMM is trained for each cuboid location taking into account all the available observations.

We set a threshold to compare each computed likelihood from the testing videos in order to fulfill the classification task. This threshold is tied to the location of cuboids and is set using the minimum likelihood value of training samples at each location multiplied by a factor k chosen depending on the frequency of anomalous sequences and can either be $k = 1, k < 1$, or $k > 1$ [19].

Eventually, when dealing with applications such as anomaly detection, we wish, as far as practicable, to achieve the optimal Equal Error Rate (EER). However, the latter is not the only point of performance on which we should rely when assessing our results. The overall performance can thus be studied by computing the Area Under the Curve (AUC).

We choose to set our model to a number of states $K = 2$ and a number of mixtures per state $M = 3$. It is better to keep those two values low as they drastically contribute to the simplicity of computing. We also carry offline and online trials for the sake of comparison. Results will be detailed later in this section. The number of states K and mixture components M is set using K-means [27] clustering of the training data, with the number of clusters varying from 2 to 20.

We train each HMM with a set of training features for each of the classes 10 times. Then we keep track of the scored results as an average across the training times. Results and comparison with different used models in the same experimental context can be observed in Table 1.

Table 1 Average recognition rates for different used HMMs in the context of video anomaly detection UCSD, ped1 and ped2 data sets

Method	Ped1	Ped2
GMM-HMM	72.03	73.19
ID-HMM	75.28	77.51
GID-HMM	89.99	87.27
IBL-HMM	**90.09**	**90.41**
Online GMM-HMM	88.60	84.53
Online ID-HMM	91.13	91.72
Online GID-HMM	89.03	84.33
Online IBL-HMM	**95.10**	**92.69**

Bold values are to reference models implemented as the main contribution of this manuscript

The results show an apparent improvement each time we chose to integrate the online EM into the HMM framework. This is related to the gradual adjustment of the parameters that allow for better fitting of the data by the proposed model. Nonetheless, it is noteworthy to mention that Online IBL-HMM performed significantly better than its offline peer, plus even better than the inverted Dirichlet-based HMM and the generalized inverted Dirichlet-based HMM as well. The online setup combined with an appropriate choice of distribution contributed to this decent amelioration.

5.2 *Anomaly Detection: Airport Security Line-Up*

This application permits identifying people going in the wrong direction in an airport security line-up. The videos are treated as sequences extracted from the anomalous Behavior data set [48]. The latter has been gathered from a surveillance camera hung up to the ceiling and filming vertically downwards. One part of the data set is clear from any anomalies and hence used for the training step, while the other is used for testing purposes. Figure 3 shows some frames from the data set.

Anomalies displayed in this data set are of a larger scale compared to the previous application, we then choose to increase the cuboid size to prevent as much as possible false positive cuboids. Here we choose 80×80 pixels. We use AUC-ROC curve [12] as a performance assessment measure. What is interesting, is that in this binary classification context, a model has to predict whether the frame is an anomaly or not. The AUC curve measures the models' performance depending on various thresholds. The highest AUC score will help us determine the best model. The AUC-ROC curve is plotted with True Positive Rate (TPR) and False Positive Rate (FPR). We thought it would also be interesting to allow some interest in evaluating the Equal Error Rate (EER) as a performance assessment. EER is an optimized value where a false positive intersects with a false negative. The better a model is the lower its EER score. Results are displayed in Fig. 4.

$$TPR = \frac{TruePositive}{TruePositive + FalseNegative} \tag{36}$$

Fig. 3 Frames from Anomalous Behavior airport wrong direction with highlighted anomalies

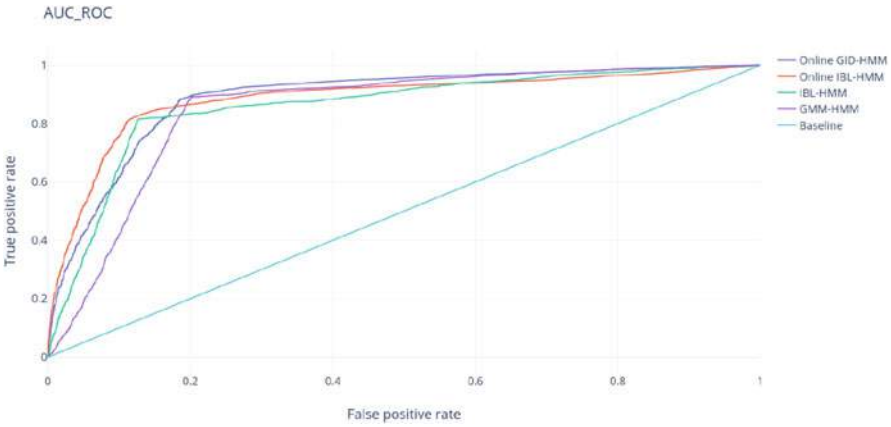


Fig. 4 AUC-ROC curve comparison of the proposed online IBL-HMM with other methods for anomalous behavior data set

Table 2 Average recognition rates for different used HMMs in the context of video anomaly detection Anomalous Behavior data set both online and offline

Method	Online	Offline
GMM-HMM	86.13	79.02
ID-HMM	89.64	80.11
GID-HMM	91.17	86.98
IBL-HMM	**94.83**	**92.06**

Bold values are to reference models implemented as the main contribution of this manuscript

$$FPR = \frac{FalsePositive}{TrueNegative + FalsePositive} \tag{37}$$

Performances of different tested methods displayed in Table 2 show the significant role played by the online learning method in improving the detection performance of anomalous events.

5.3 Abnormal Crowd Behavior: Escape Scene

This experiment aims to capture abnormal crowd behavior in three different scenes in the video sequence of unusual crowd events captured synthetically by the University of Minnesota (UMN) [46]. The data set is composed of videos of 11 different scenarios of an escape event in 3 different indoor and outdoor scenes: Lawn, Indoor, and Plaza. Each video is composed of an initial part of normal behavior followed by sequences of abnormal behavior where people run from the center of the scene to simulate an escaping event. All footage is recorded at a frame rate of 30 frames per second at a resolution of 640 $\times$ 480 using a static camera.

Figure 5 shows sample frames of these scenes. Here, the process for identifying the likely patterns is performed in a similar way as in [35], where we use the bag of words [30] method to identify the events and normal videos for training LDA [8]. For computational simplicity, the resolution of the particle grid is kept at 25% of the number of pixels. We partition our frames into blocs of C clips. Then, from each clip C_j, W visual words are extracted. We randomly pick visual words of size $5 \times 5 \times 10$ and code a book of size S using K-means clustering. In this case, we extract $W = 30$ visual words from a block of 10 frames. Thus a final codebook contains $C = 10$ clips. To evaluate our model, 50 different frames of each scene are selected.

Table 3 shows the average accuracy comparison of several tested methods namely online-based and offline-based HMMs implemented for the sake of this particular comparison. We specifically want to focus on the role played by online HMMs compared to offline models but in detecting escape scenes, we also want to focus on the role played by the IBL as a distribution to improve the average recognition accuracy of anomalous scenes. Overall, the proposed method achieves the best accuracy with an average of 89.12%, which is higher than the average accuracy of 83.53% where we did not use the online-based model. We also observe that both online and offline IBL-HMM perform better compared to other methods.

Fig. 5 Frames from the UMN data set with normal (upper row) and abnormal escape scenes (bottom row) from three different indoor and outdoor scenes

Table 3 Average recognition rates for different used HMMs in the context of a crowd escape scene detection on the UMN data set, both online and offline

Method	Online	Offline
GMM-HMM	71.13	69.80
ID-HMM	76.08	73.42
GID-HMM	83.40	78.55
IBL-HMM	**89.12**	**83.53**

Bold values are to reference models implemented as the main contribution of this manuscript

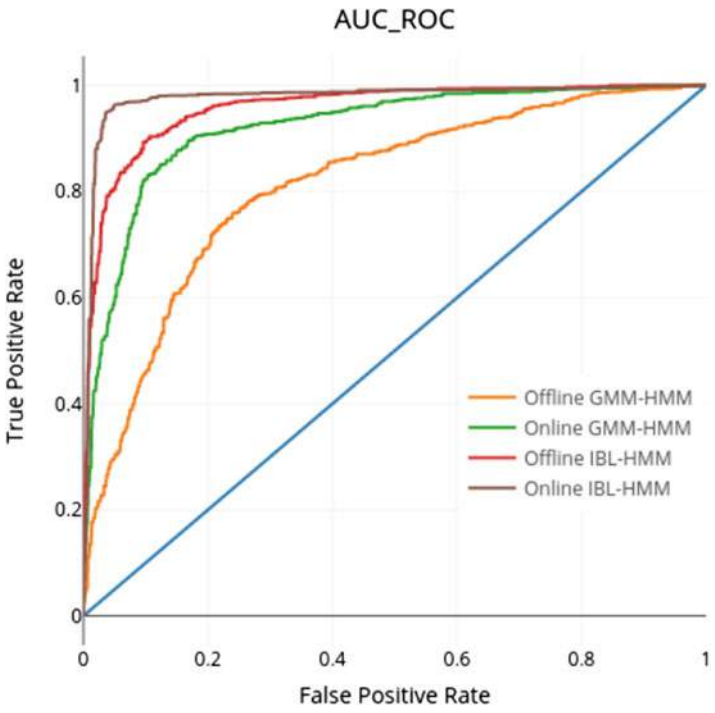


Fig. 6 AUC-ROC curve for each of the tested models on the UMN data set

For further performance evaluation, we have presented the ROC curves in Fig. 6 for the different used models and can thus observe that our method achieves a better ratio and the number of false positives is significantly lower.

One of the main takeaways is the crucial role that online learning plays in reducing false positive detection of anomalous behavior especially in binary contexts where only two scenarios such as "normal" or "escape scene" are possible. Clearly, in the mentioned situations we aim for the least false positive detection rate possible to avoid false alerts and thus reduce unnecessary alarming situations.

6 Conclusion

There is a multitude of techniques that researchers are adopting to address the challenge of abundant and massive data modeling. Online learning methods are one of the most powerful tools to handle big streams of data such as videos in a real-time context. Using HMMs is also a suitable way to deal with dynamic data such as videos, but the biggest challenge remains in finding the most powerful distribution to faithfully model specific types of data such as positive vectors.

Further, the interest in adopting IBL mixtures for modeling our data arose from the limitations encountered when other distributions such as Gaussian mixtures and inverted Dirichlet were adopted. In fact, IBL mixtures provided a smaller number of parameters compared to the generalized inverted Dirichlet, not to mention that it showed its effectiveness when dealing with positive vectors modeling in contrast to the rest of the tested distributions. In this paper, we proposed a model in which all the aforementioned problems are addressed simultaneously in the case of human activities modeling and anomalies detection. The developed approach applies online learning of parameters within the HMM framework. Experimental results involving challenging real-life applications such as anomaly detection in a human crowd context showed that the proposed approach is highly promising. We have demonstrated that the proposed method is highly effective at discriminating between scenes of normal and abnormal behavior, and that our approach operates in real-time. Future works are intended to be done in the near future extending this work to different flexible distributions and considering a hybrid Generative-Discriminative model using Support Vector Machines kernels to improve classification capabilities and to further reduce error rates.

Acknowledgments The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. E.L. Andrade, O.J. Blunsden, R.B. Fisher, Performance analysis of event detection models in crowded scenes, in *2006 IET International Conference on Visual Information Engineering* (2006), pp. 427–432
2. E.L. Andrade, S. Blunsden, R.B. Fisher, Hidden Markov models for optical flow analysis in crowds, in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1 (IEEE, Piscataway, 2006), pp. 460–463
3. E.L. Andrade, S. Blunsden, R.B. Fisher, Modelling crowd scenes for event detection, in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1 (IEEE, Piscataway, 2006), pp. 175–178
4. L.E. Baum, T. Petrie, G. Soules, N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann. Math. Stat. **41**(1), 164–171 (1970)
5. T. Bdiri, N. Bouguila, Positive vectors clustering using inverted dirichlet finite mixture models. Exp. Syst. Appl. **39**(2), 1869–1882 (2012)
6. T. Bdiri, N. Bouguila, D. Ziou, Variational Bayesian inference for infinite generalized inverted dirichlet mixtures with feature selection and its application to clustering. Appl. Intell. **44**(3), 507–525 (2016)
7. M. Bertini, A. Del Bimbo, L. Seidenari, Multi-scale and real-time non-parametric approach for anomaly detection and localization. Comput. Vision Image Understand. **116**(3), 320–329 (2012)
8. D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
9. L. Bottou, et al., Online learning and stochastic approximations. On-Line Learn. Neural Netw. **17**(9), 142 (1998)

10. S. Bourouis, R. Alroobaea, S. Rubaiee, M. Andejany, F.M. Almansour, N. Bouguila, Markov chain Monte carlo-based bayesian inference for learning finite and infinite inverted beta-liouville mixture models. IEEE Access **9**, 71170–71183 (2021)
11. S. Boutemedjet, D. Ziou, N. Bouguila, Model-based subspace clustering of non-Gaussian data. Neurocomputing **73**(10–12), 1730–1739 (2010)
12. A.P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit. **30**(7), 1145–1159 (1997)
13. O. Cappé, Online em algorithm for hidden Markov models. J. Comput. Graph. Stat. **20**(3), 728–749 (2011)
14. L. Carnevali, F. Santoni, E. Vicario, Learning marked Markov modulated poisson processes for online predictive analysis of attack scenarios, in *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)* (IEEE, Piscataway, 2019), pp. 195–205
15. S. Chackravarthy, S. Schmitt, L. Yang, Intelligent crime anomaly detection in smart cities using deep learning, in *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)* (IEEE, Piscataway, 2018), pp. 399–404
16. A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm. J. Roy. Stat. Soc. B **39**(1), 1–22 (1977)
17. A. Drosou, D. Ioannidis, K. Moustakas, D. Tzovaras, Spatiotemporal analysis of human activities for biometric authentication. Comput. Vision Image Understand. **116**(3), 411–421 (2012)
18. E. Epaillard, N. Bouguila, Hidden Markov models based on generalized dirichlet mixtures for proportional data modeling, in *IAPR Workshop on Artificial Neural Networks in Pattern Recognition* (Springer, Berlin, 2014), pp. 71–82
19. E. Epaillard, 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 Recognit. **55**, 125–136 (2016)
20. E. Epaillard, N. Bouguila, Variational Bayesian learning of generalized dirichlet-based hidden Markov models applied to unusual events detection. IEEE Trans. Neural Netw. Learn. Syst. **30**(4), 1034–1047 (2018)
21. E. Epaillard, N. Bouguila, D. Ziou, Classifying textures with only 10 visual-words using hidden Markov models with dirichlet mixtures, in *International Conference on Adaptive and Intelligent Systems* (Springer, Berlin, 2014), pp. 20–28
22. W. Fan, R. Wang, N. Bouguila, Simultaneous positive sequential vectors modeling and unsupervised feature selection via continuous hidden markov models. Pattern Recognit. **119**, 108073 (2021)
23. K.-T. Fang, S. Kotz, K.W. Ng, *Symmetric Multivariate and Related Distributions* (Chapman and Hall/CRC, Boca Raton, 2018)
24. Z. Ghahramani, An introduction to hidden Markov models and Bayesian networks, in *Hidden Markov Models: Applications in Computer Vision* (World Scientific, Singapore, 2001)
25. Z. Ghahramani, M.I. Jordan, Factorial hidden Markov models. Mach. Learn. **29**(2), 245–273 (1997)
26. R.D. Gupta, D.St.P. Richards, Multivariate liouville distributions, iii. J. Multivariate Anal. **43**(1), 29–57 (1992)
27. J.A. Hartigan, M.A. Wong, Algorithm as 136: a k-means clustering algorithm. J. Roy. Stat. Soc. C **28**(1), 100–108 (1979)
28. C. Hu, W. Fan, J.-X. Du, N. Bouguila, A novel statistical approach for clustering positive data based on finite inverted beta-liouville mixture models. Neurocomputing **333**, 110–123 (2019)
29. F. Jiang, Y. Wu, A.K. Katsaggelos, Abnormal event detection from surveillance video by dynamic hierarchical clustering, in *2007 IEEE International Conference on Image Processing*, vol. 5 (IEEE, Piscataway, 2007), pp. V–145
30. T. Joachims, Text categorization with support vector machines: learning with many relevant features, in *European Conference on Machine Learning* (Springer, Berlin, 1998), pp. 137–142
31. S.-R. Ke, H.L.U. Thuc, Y.-J. Lee, J.-N. Hwang, J.-H. Yoo, K.-H. Choi, A review on video-based human activity recognition. Computers **2**(2), 88–131 (2013)

32. S. Le Corff, G. Fort, Online expectation maximization based algorithms for inference in hidden Markov models. Electron. J. Stat. **7**, 763–792 (2013)
33. S.E. Levinson, L.R. Rabiner, M.M. Sondhi, An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. Bell Syst. Tech. J. **62**(4), 1035–1074 (1983)
34. M.A. Mashrgy, T. Bdiri, N. Bouguila, Robust simultaneous positive data clustering and unsupervised feature selection using generalized inverted dirichlet mixture models. Knowl. Based Syst. **59**, 182–195 (2014)
35. R. Mehran, A. Oyama, M. Shah, Abnormal crowd behavior detection using social force model, in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Piscataway, 2009), pp. 935–942
36. G. Mongillo, S. Deneve, Online learning with hidden Markov models. Neural Comput. **20**(7), 1706–1716 (2008)
37. R. Nasfi, M. Amayri, N. Bouguila, A novel approach for modeling positive vectors with inverted dirichlet-based hidden Markov models. Knowl.-Based Syst. **192**, 105335 (2020)
38. P.E. Nikravesh, *Computer-Aided Analysis of Mechanical Systems* (Prentice-Hall, Hoboken, 1988)
39. L.R. Rabiner, B.H. Juang, An introduction to hidden Markov models. IEEE ASSP Mag. **3**, 4–6 (1986)
40. P.C. Ribeiro, R. Audigier, Q.C. Pham, Rimoc, a feature to discriminate unstructured motions: application to violence detection for video-surveillance. Comput. Vision Image Understand. **144**, 121–143 (2016)
41. R. Sanford, S. Gorji, L.G. Hafemann, B. Pourbabaee, M. Javan, Group activity detection from trajectory and video data in soccer, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (2020), pp. 898–899
42. T. Sebastian, V. Jeyaseelan, L. Jeyaseelan, S. Anandan, S. George, S.I. Bangdiwala, Decoding and modelling of time series count data using poisson hidden Markov model and Markov ordinal logistic regression models. Stat. Methods Med. Res. **28**, 1552–1563 (2018)
43. R.S. Sidhu, M. Sharad, Smart surveillance system for detecting interpersonal crime, in *2016 International Conference on Communication and Signal Processing (ICCSP)* (IEEE, Piscataway, 2016), pp. 2003–2007
44. J.C. Stiller, G. Radons, Online estimation of hidden Markov models. IEEE Signal Process. Lett. **6**(8), 213–215 (1999)
45. M.D. Tanzil Shahria, F.T. Progga, S. Ahmed, A. Arisha, Application of neural networks for detection of sexual harassment in workspace, in *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)* (IEEE, Piscataway, 2021), pp. 1–4
46. Unusual crowd activity dataset of university of minnesota. http://mha.cs.umn.edu/movies/crowd-activity-all.avi
47. T. Xiang, S. Gong, Video behavior profiling for anomaly detection. IEEE Trans. Pattern Anal. Mach. Intell. **30**(5), 893–908 (2008)
48. A. Zaharescu, R. Wildes, Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing, in *European Conference on Computer Vision* (Springer, Berlin, 2010), pp. 563–576
49. N. Zamzami, N. Bouguila, Deriving probabilistic SVM kernels from exponential family approximations to multivariate distributions for count data, in *Mixture Models and Applications* (Springer, Berlin, 2020), pp. 125–153

A Novel Continuous Hidden Markov Model for Modeling Positive Sequential Data

Wenjuan Hou, Wentao Fan, Manar Amayri, and Nizar Bouguila

1 Introduction

Nowadays, sequential data modeling has become a critical research topic in different fields, ranging from gesture recognition [1], human genome sequences modeling [2], text clustering [3] to abnormal behaviors detection [4]. One of the most effective approaches for modeling sequential data is the Hidden Markov model (HMM) [5, 6], which is formulated by assuming that each observed data instance in a hidden state is generated from a probability distribution (often known as the emission distribution). When observations are continuous, we have the *continuous* HMM with a continuous probability distribution as the emission density.

For continuous HMM, the Gaussian distribution or the Gaussian mixture model (GMM) has normally been applied as the emission density due to their well-defined properties [7, 8]. This choice, however, the Gaussian distribution or the GMM, is not suitable in situations where we have non-Gaussian data [9–14]. According to

W. Hou
Instrumental Analysis Center, Huaqiao University, Xiamen, China
e-mail: houwenjuan@hqu.edu.cn

W. Fan
Department of Computer Science and Technology, Huaqiao University, Xiamen, China
e-mail: fwt@hqu.edu.cn

M. Amayri
G-SCOP Lab, Grenoble Institute of Technology, Grenoble, France
e-mail: manar.amayri@grenoble-inp.fr

N. Bouguila (✉)
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca

N. Bouguila et al. (eds.), *Hidden Markov Models and Applications*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-99142-5_8

various studies, other distributions can also be adopted as the emission densities for HMMs and are able to provide superior performance than Gaussian-based HMMs for modeling sequential data [15–18]. Particularly, the mixture of inverted Beta-Liouville (IBL) distributions [19] has shown promising performance in modeling positive data that are naturally involved in a variety of real-world applications [19–23]. Therefore, the aim of this work is to propose an effective approach for modeling positive sequential data by formulating a continuous HMM with the IBL mixture model as the emission density.

The contributions of our work can be summarized as follows. Firstly, we propose a novel continuous HMM for modeling positive sequential observations, in which the emission distribution of each hidden state is distributed according to an IBL mixture model that has shown better capability for modeling positive data than other popular distributions (e.g., the Gaussian distribution). Secondly, the proposed IBL-based HMM is learned by theoretically developing a convergence-guaranteed algorithm based on variational Bayes (VB) [24, 25]. The VB inference is a deterministic learning algorithm for approximating probability densities through optimization and has been successfully applied in various Bayesian models. Lastly, we demonstrate the advantages of our model by conducting experiments on real-world positive sequential data sets.

The remaining part of this chapter can be listed as follows. In Sect. 2, we propose the continuous HMM with IBL mixtures. In Sect. 3, we develop a learning algorithm based on VB inference to estimate the parameters of our model. In Sect. 4, we provide experimental results of our model on two real-world sequential data sets. Finally, conclusion is given in Sect. 5.

2 The HMM with Inverted Beta-Liouville Mixture Models

2.1 The Formulation of IBL-HMM

If a D-dimensional vector $\mathbf{X} = (X_1, \ldots, X_D)$ in R_+^D is distributed according to an inverted Beta-Liouville (IBL) distribution [19], then the probability dentistry function is defined by:

$$\mathrm{IBL}(\mathbf{X}|\boldsymbol{\lambda}, \alpha, \beta) = \frac{\Gamma(\alpha+\beta)\Gamma(\sum_{d=1}^{D}\lambda_d)}{\Gamma(\alpha)\Gamma(\beta)} \prod_{d=1}^{D} \frac{X_d^{\lambda_d - 1}}{\Gamma(\lambda_d)} \frac{(\sum_{d=1}^{D} X_d)^{\alpha - \sum_{d=1}^{D}\lambda_d}}{(1+\sum_{d=1}^{D} X_d)^{\alpha+\beta}}, \tag{1}$$

where $\{\boldsymbol{\lambda}, \alpha, \beta\}$ is the set of parameters of the IBL distribution.

Now we can formulate a continuous HMM that deploys a mixture of k IBL distributions as its emission density. Then, we can define the IBL-based HMM (denoted by IBL-HMM) based on a set of parameters $\Theta = \{\boldsymbol{\pi}, \mathbf{a}, \mathbf{c}, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \boldsymbol{\beta}\}$, where $\boldsymbol{\pi} = \{\pi_i\}_{i=1}^{N}$ denotes the vector of initial-state probabilities, $\mathbf{a} = \{a_{ij}\}_{i=1,j=1}^{N,N}$

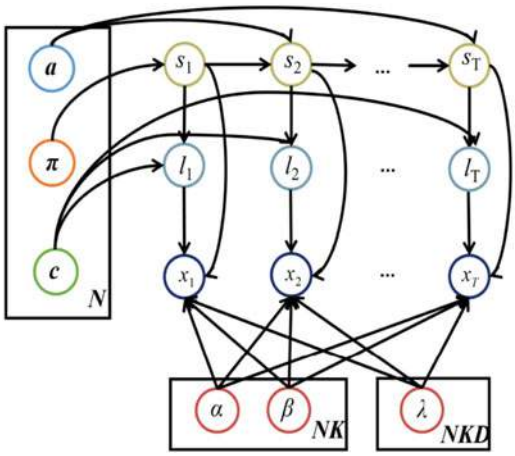


Fig. 1 The graphical model that represents the proposed IBL-HMM

denotes the state transition matrix, $\mathbf{c} = \{c_{ik}\}_{i=1,k=1}^{N,K}$ is the matrix of mixing coefficients with c_{ik} indicates the mixing coefficient of component k under the state i; $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_{ik}\}_{i=1,k=1}^{N,K}$, $\boldsymbol{\alpha} = \{\alpha_{ik}\}_{i=1,k=1}^{N,K}$, and $\boldsymbol{\beta} = \{\beta_{ik}\}_{i=1,k=1}^{N,K}$ are the parameters of the IBL distributions, where $\boldsymbol{\lambda}_{ik}$, α_{ik}, and β_{ik} represent the parameters of the kth IBL distribution in state i.

Given a sequence of T observations $X = \{\mathbf{x}_t\}_{t=1}^{T}$, where $\mathbf{x}_t = \{x_{td}\}_{d=1}^{D}$ represents the feature vector at time t, we can define the complete-data likelihood for the IBL-HMM as

$$p(X, S, L|\Theta) = \pi_{s_1}\left[\prod_{t=1}^{T-1} a_{s_t s_{t+1}}\right]\left[\prod_{t=1}^{T} c_{s_t l_t} \mathrm{IBL}(\mathbf{x}_t|\boldsymbol{\lambda}_{s_t l_t}, \alpha_{s_t l_t}, \beta_{s_t l_t})\right], \tag{2}$$

where the latent variable s_t is the state indicator, and the latent variable l_t is the indicator of the mixture component. Then, we can represent the likelihood function of model parameters Θ as

$$p(X|\Theta) = \sum_{S,L} \pi_{s_1}\left[\prod_{t=1}^{T-1} a_{s_t s_{t+1}}\right]\left[\prod_{t=1}^{T} c_{s_t l_t} \mathrm{IBL}(\mathbf{x}_t|\boldsymbol{\lambda}_{s_t l_t}, \alpha_{s_t l_t}, \beta_{s_t l_t})\right]. \tag{3}$$

Figure 1 shows the graphical model that represents the proposed IBL-HMM.

2.2 The Prior Distributions

As we formulate the IBL-HMM through the Bayesian framework, we need to assign prior distributions for all random variables. For parameters $\boldsymbol{\pi}$, $\mathbf{a}$, and $\mathbf{c}$, we adopt

Dirichlet distributions Dir(·) as their priors as in [15, 16, 26]

$$p(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi}|\theta_1^{\lambda}, \ldots, \theta_N^{\lambda}), \tag{4}$$

$$p(\mathbf{a}) = \prod_{i=1}^{N} \mathrm{Dir}(a_{i1}, \ldots, a_{iN}|\theta_{i1}^{a}, \ldots, \theta_{iN}^{a}), \tag{5}$$

$$p(\mathbf{c}) = \prod_{i=1}^{N} \mathrm{Dir}(c_{i1}, \ldots, c_{iK}|\theta_{i1}^{c}, \ldots, \theta_{iK}^{c}). \tag{6}$$

For positive parameters $\boldsymbol{\lambda}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$, following [19], we adopt Gamma distributions $\mathcal{G}(\cdot)$ as their priors

$$p(\boldsymbol{\lambda}) = \prod_{i=1}^{N}\prod_{k=1}^{K}\prod_{d=1}^{D} \mathcal{G}(\boldsymbol{\lambda}|\mathbf{m}, \mathbf{b}) = \prod_{i=1}^{N}\prod_{k=1}^{K}\prod_{d=1}^{D} \frac{b_{ikd}^{m_{ikd}}}{\Gamma(m_{ikd})}\lambda^{m_{ikd}-1}e^{-b_{ikd}\lambda_{ikd}}, \tag{7}$$

$$p(\boldsymbol{\alpha}) = \prod_{i=1}^{N}\prod_{k=1}^{K} \mathcal{G}(\boldsymbol{\alpha}|\mathbf{u}, \mathbf{v}) = \prod_{i=1}^{N}\prod_{k=1}^{K} \frac{v_{ik}^{u_{ik}}}{\Gamma(u_{ik})}\alpha_{ik}^{u_{ik}-1}e^{-v_{ik}\alpha_{ik}}, \tag{8}$$

$$p(\boldsymbol{\beta}) = \prod_{i=1}^{N}\prod_{k=1}^{K} \mathcal{G}(\boldsymbol{\beta}|\mathbf{u}', \mathbf{v}') = \prod_{i=1}^{N}\prod_{k=1}^{K} \frac{v_{ik}^{\prime u'_{ik}}}{\Gamma(u'_{ik})}\beta_{ik}^{u'_{ik}-1}e^{-v'_{ik}\beta_{ik}}. \tag{9}$$

3 Model Fitting by Variational Bayes

In this part, we systematically develop an effective learning approach that is tailored for learning the proposed IBL-HMM based on variational Bayes (VB). The central idea of our VB model learning approach is to discover a suitable approximation $q(S, L, \Theta)$ to the true posterior $p(S, L, \Theta|X)$, where $\{S, L, \Theta\}$ denotes the set of latent and unknown variables in IBL-HMM as described previously. To obtain a tractable inference procedure, we apply the mean-field theory [27] as

$$q(S, L, \Theta) = q(S, L)q(\Theta) = q(S, L)q(\mathbf{a})q(\boldsymbol{\pi})q(\mathbf{c})q(\boldsymbol{\lambda})q(\boldsymbol{\alpha})q(\boldsymbol{\beta}). \tag{10}$$

Based on VB inference, we can find the approximations $q(S, L)$ and $q(\Theta)$ (also known as variational posteriors) by maximizing the objective function, which is the evidence lower bound (ELBO) and is defined by:

$$\mathrm{ELBO}(q) = \int q(S, L, \Theta) \ln \frac{p(X, S, L, \Theta)}{q(S, L, \Theta)} \mathrm{d}S\mathrm{d}L\mathrm{d}\Theta$$

$$\begin{aligned}=&\mathrm{ELBO}(q(\boldsymbol{\pi}))+\mathrm{ELBO}(q(\mathbf{a}))+\mathrm{ELBO}(q(\mathbf{c}))+\mathrm{ELBO}(q(S,L))\\&+\mathrm{ELBO}(q(\boldsymbol{\lambda}))+\mathrm{ELBO}(q(\boldsymbol{\alpha}))+\mathrm{ELBO}(q(\boldsymbol{\beta}))+Constant.\end{aligned} \tag{11}$$

3.1 The Optimization of $q(\mathbf{a})$, $q(\boldsymbol{\pi})$, and $q(\mathbf{c})$

The variational posteriors of the initial state probability matrix $q(\boldsymbol{\pi})$, the state transition matrix $q(\mathbf{a})$, and the mixing coefficient matrix $q(\mathbf{c})$ can be optimized by maximizing the ELBO in (17) as

$$q(\mathbf{a})=\prod_{i=1}^{N}\mathrm{Dir}(a_{i1},\ldots,a_{iN}|\Lambda_{i1}^{a},\ldots,\Lambda_{iN}^{a}). \tag{12}$$

$$q(\boldsymbol{\pi})=\mathrm{Dir}(\pi_1,\ldots,\pi_N|\Lambda_1^{\pi},\ldots,\Lambda_N^{\pi}), \tag{13}$$

$$q(\mathbf{c})=\prod_{i=1}^{N}\mathrm{Dir}(c_{i1},\ldots,c_{iK}|\Lambda_{i1}^{c},\ldots,\Lambda_{iK}^{c}), \tag{14}$$

where the involved hyperparameters can be obtained by:

$$\Lambda_{ij}^{a}=\sum_{t=1}^{T-1}\omega_{ijt}^{a}+\theta_{ij}^{a}, \tag{15}$$

$$\Lambda_{i}^{\pi}=\omega_{i}^{\pi}+\theta_{i}^{\pi}, \tag{16}$$

$$\Lambda_{ik}^{c}=\sum_{t=1}^{T}\omega_{ikt}^{c}+\theta_{ik}^{c}, \tag{17}$$

with ω_{ijt}^{a}, ω_{i}^{λ}, and ω_{ikt}^{c} that are defined by:

$$\omega_{ijt}^{a}=q(s_t=i,s_{t+1}=j), \tag{18}$$

$$\omega_{i}^{\pi}=q(s_1=i), \tag{19}$$

$$\omega_{ikt}^{c}=q(s_t=i,l_t=k). \tag{20}$$

It is noteworthy that the values of ω_{ijt}^{q}, ω_{i}^{π}, and ω_{ikt}^{c} can be easily obtained through the classic forward–backward algorithm according to [28].

3.2 The Optimization of $q(\boldsymbol{\lambda})$, $q(\boldsymbol{\alpha})$, and $q(\boldsymbol{\beta})$

The variational posteriors $q(\boldsymbol{\lambda})$,$q(\boldsymbol{\alpha})$, and $q(\boldsymbol{\beta})$ can be optimized by maximizing the ELBO with respect to the corresponding parameter as

$$q(\boldsymbol{\lambda}) = \prod_{i=1}^{N}\prod_{k=1}^{K}\prod_{d=1}^{D}\mathcal{G}(\lambda_{ikd}|m_{ikd}^{*}, b_{ikd}^{*}), \tag{21}$$

$$q(\boldsymbol{\alpha}) = \prod_{i=1}^{N}\prod_{k=1}^{K}\mathcal{G}(\alpha_{ik}|u_{ik}^{*}, v_{ik}^{*}), \tag{22}$$

$$q(\boldsymbol{\beta}) = \prod_{i=1}^{N}\prod_{k=1}^{K}\mathcal{G}(\beta_{ik}|u_{ik}^{\prime *}, v_{ik}^{\prime *}), \tag{23}$$

where we have

$$\begin{aligned} m_{ikd}^{*} =& m_{ikd} + \sum_{t=1}^{T}\omega_{ikt}^{c}\bar{\lambda}_{ikd}\Bigg[\psi(\sum_{d=1}^{D}\bar{\lambda}_{ikd}) + \psi'(\sum_{d=1}^{D}\bar{\lambda}_{ikd})\sum_{j\neq d}^{D}(\langle\ln\lambda_{ikj}\rangle \\ & - \ln\bar{\lambda}_{ikj})\bar{\lambda}_{ikj} - \psi(\bar{\lambda}_{ikd})\Bigg], \end{aligned} \tag{24}$$

$$b_{ikd}^{*} = b_{ikd} - \sum_{t=1}^{T}\omega_{ikt}^{c}\Bigg[\ln x_{td} - \ln(\sum_{d=1}^{D}x_{td})\Bigg], \tag{25}$$

$$\begin{aligned} u_{ik}^{*} =& u_{ik} + \sum_{t=1}^{T}\omega_{ikt}^{c}\big[\bar{\beta}_{ik}\psi'(\bar{\alpha}_{ik}+\bar{\beta}_{ik})(\langle\ln\beta_{ik}\rangle - \ln\bar{\beta}_{ik}) + \psi(\bar{\alpha}_{ik}+\bar{\beta}_{ik}) \\ & - \psi(\bar{\alpha}_{ik})\big]\bar{\alpha}_{ik}, \end{aligned} \tag{26}$$

$$v_{ik}^{*} = v_{ik} - \sum_{t=1}^{T}\omega_{ikt}^{c}\Bigg[\ln(\sum_{d=1}^{D}x_{td}) - \ln(1+\sum_{d=1}^{D}x_{td})\Bigg], \tag{27}$$

$$u_{ik}'^{*} = u_{ik}' + \sum_{t=1}^{T} \omega_{ikt}^{c} \big[\bar{\alpha}_{ik}\psi'(\bar{\alpha}_{ik} + \bar{\beta}_{ik})(\langle \ln \alpha_{ik} \rangle - \ln \bar{\alpha}_{ik}) + \psi(\bar{\alpha}_{ik} + \bar{\beta}_{ik}) - \psi(\bar{\beta}_{ik})\big]\bar{\beta}_{ik}, \tag{28}$$

$$v_{ik}'^{*} = v_{ik}' + \sum_{t=1}^{T} \omega_{ikt}^{c} \ln(1 + \sum_{d=1}^{D} x_{td}), \tag{29}$$

where the expected values in above equations are given by:

$$\bar{\lambda}_{ikd} = \frac{m_{ikd}^{*}}{b_{ikd}^{*}}, \qquad \bar{\alpha}_{ik} = \frac{u_{ik}^{*}}{v_{ik}^{*}}, \qquad \bar{\beta}_{ik} = \frac{u_{ik}'^{*}}{v_{ik}'^{*}}, \tag{30}$$

$$\langle \ln \lambda_{ikd} \rangle = \psi(m_{ikd}^{*}) - \ln b_{ikd}^{*}, \tag{31}$$

$$\langle \ln \alpha_{ik} \rangle = \psi(u_{ik}^{*}) - \ln v_{ik}^{*}, \quad \langle \ln \beta_{ik} \rangle = \psi(u_{ik}'^{*}) - \ln v_{ik}'^{*}. \tag{32}$$

3.3 The Optimization of $q(S, L)$

The joint variational posterior $q(S, L)$ is optimized by maximizing the ELBO with respect to the state indicator S and the mixture component indicator L

$$q(S, L) = \frac{1}{Z}\pi_{s_1}^{*} \prod_{t=1}^{T-1} a_{s_t s_{t+1}}^{*} \prod_{t=1}^{T} c_{s_t, l_t}^{*} \mathrm{IBL}^{*}(\mathbf{x}_t | \boldsymbol{\lambda}_{s_t l_t}, \alpha_{s_t l_t}, \beta_{s_t l_t}), \tag{33}$$

where we have

$$\pi_i^{*} = \exp\left[\int q(\boldsymbol{\pi}) \ln \pi_i d\boldsymbol{\pi}\right] = \exp\left[\Psi(\Lambda_i^{\pi}) - \Psi(\sum_{i=1}^{N} \Lambda_i^{\pi})\right], \tag{34}$$

$$a_{ij}^{*} = \exp\left[\int q(\mathbf{a}) \ln a_{ij} d\mathbf{a}\right] = \exp\left[\Psi(\Lambda_{ij}^{a}) - \Psi(\sum_{j=1}^{N} \Lambda_{ij}^{a})\right], \tag{35}$$

$$c_{ik}^{*} = \exp\left[\int q(\mathbf{c}) \ln c_{ik} d\mathbf{c}\right] = \exp\left[\Psi(\Lambda_{ik}^{c}) - \Psi(\sum_{k=1}^{K} \Lambda_{ik}^{c})\right]. \tag{36}$$

$$\begin{aligned}\mathrm{IBL}^*(\mathbf{x}_t|\boldsymbol{\lambda}_{s_t l_t}, \alpha_{s_t l_t}, \beta_{s_t l_t}) = \exp\Bigg[&\Big\langle \ln \frac{\Gamma\big(\sum_{d=1}^{D} \lambda_{ikd}\big)}{\prod_{d=1}^{D} \Gamma(\lambda_{ikd})}\Big\rangle + \Big\langle \ln \frac{\Gamma(\alpha_{ik}+\beta_{ik})}{\Gamma(\alpha_{ik})\Gamma(\beta_{ik})}\Big\rangle \\ &+ \left(\bar{\alpha}_{ik} - \sum_{d=1}^{D} \bar{\lambda}_{ikd}\right) \ln\left(\sum_{d=1}^{D} x_{td}\right) - \left(\bar{\alpha}_{ik} + \bar{\beta}_{ik}\right) \ln\left(1 + \sum_{d=1}^{D} x_{td}\right) \\ &+ \sum_{d=1}^{D} \left(\bar{\lambda}_{ikd} - 1\right) \ln x_{td}\Bigg], \end{aligned} \tag{37}$$

where the normalizing constant Z in (33) can be obtained by:

$$Z = q(X|\Theta^*) = \sum_{S,L} \pi^*_{s_1} \prod_{t=1}^{T-1} a^*_{s_t s_{t+1}} \prod_{t=1}^{T} c^*_{s_t, l_t} \mathrm{IBL}^*(\mathbf{x}_t|\boldsymbol{\lambda}_{s_t l_t}, \alpha_{s_t l_t}, \beta_{s_t l_t}). \tag{38}$$

It is noteworthy that (38) can be considered as an approximation to the likelihood of the optimized model with parameters Θ^*, as we compare it with (3).

As the $ELBO(q)$ in (11) is convex with respect to each of the variational posterior, the proposed VB inference algorithm for leaning IBL-HMM is guaranteed to converge [27]. Moreover, it is straightforward to inspect the convergence status by checking if the variation in ELBO(q) has fallen below some predefined threshold (e.g., less than 10^{-4}). The VB-based algorithm for learning the IBL-HMM is provided in Algorithm 2.

Algorithm 2 The VB inference of IBL-HMM

1: Initialize hyperparameters
2: Initialize ω^{π}, ω^{a}, $and\omega^{c}$ from their prior distributions with (4), (5), and (6)
3: Compute Λ^{π}, Λ^{a}, and Λ^{c} with (15), (16), and (17)
4: Initialize $\boldsymbol{\pi}$, $\mathbf{a}$, and $\mathbf{c}$ with (34)∼(36)
5: **repeat**
6: Compute the responsibilities ω^{λ}, ω^{a}, and ω^{c} using $\boldsymbol{\lambda}$, $\mathbf{a}$, and $\mathbf{c}$ with (18), (19), and (20)
7: Optimize variational posteriors $q(\boldsymbol{\lambda})$, $q(\boldsymbol{\alpha})$, and $q(\boldsymbol{\beta})$ with (21) ∼ (23)
8: Update Λ^{π}, Λ^{a}, and Λ^{c} using ω^{π}, ω^{a}, and ω^{c} with (15), (16), and (17)
9: Update $\boldsymbol{\lambda}$, $\mathbf{a}$, and $\mathbf{c}$ using Λ^{π}, Λ^{a}, and Λ^{c} with (34)∼(36)
10: Compute the approximated likelihood Z using (38)
11: **until** Convergence is reached

4 Experimental Results

In order to test the effectiveness of the proposed IBL-HMM and the developed VB model learning method, we conducted two experiments on two real-world positive sequential data sets. We initialized the hyperparameters as follows: $m_{ikd} = 0.5$,

Table 1 The detailed information of the tested data sets

Data sets	No. of observations	No. of features	No. of classes
AR	75128	8	4
EEG	14980	14	2

$b_{ikd} = 0.01$, $u_{ik} = 0.1$, and $v_{ik} = 0.05$. The initial value of the number of mixture components K in the IBL mixture model was set to 10. The number of hidden states for all experiments was set to 2. These values were chosen based on cross-validation.

4.1 Data Sets and Experimental Settings

In our experiments, two publicly available data sets from the UCI Machine Learning Repository[1] were adopted for testing the performance of the proposed IBL-HMM, including the activity recognition with healthy older people using a batteryless wearable sensor data set (denoted by the AR data set), and the EEG eye state data set (denoted by the EEG data set).

The AR data set contains sequential motion data from 14 healthy older people aged 66 to 86 years old using a batteryless, wearable sensor on top of their clothing for the recognition of activities in clinical environments. It includes 75218 data sequences that can be divided into 4 different activities (sit on bed, sit on chair, lying, and ambulating), where each sequence contains 8 features (e.g., Acceleration reading in G for frontal axis, Received signal strength indicator, etc.).

The EEG data set contains data sequences that were obtained from one continuous EEG measurement with the Emotiv EEG Neuro headset. The duration of the measurement was 117 seconds. The eye state (open or closed) was detected via a camera during the EEG measurement and added later manually to the file after analyzing the video frames. This data set includes 14980 sequences with 14 features. Table 1 summarizes the information of these two real-life sequential data sets.

In our experiment, these two data sets were normalized and then modeled by the proposed IBL-HMM. In order to demonstrate the advantages of our model, we compared it with other well-defined HMMs that employ different mixture models as their emission densities: the HMM with Gaussian mixture models, which is learned by VB inference (GMM-HMM-VB) [8], the HMM with Dirichlet mixture model, which is learned by EM algorithm (DMM-HMM-EM) [16], the HMM with inverted Dirichlet mixture model, which is learned by EM algorithm (IDMM-HMM-EM) [17], and the HMM with inverted Dirichlet mixture model, which is learned by VB inference (IDMM-HMM-VB) [26]. For the tested models, we adopted the same

[1] https://archive.ics.uci.edu.

Table 2 The average recognition performance over 10 runs by different approaches

Methods	The AR data set
GMM-HMM-VB [8]	0.801 ± 0.011
DMM-HMM-EM [16]	0.827 ± 0.009
IDMM-HMM-EM [17]	0.839 ± 0.015
IDMM-HMM-VB [26]	0.855 ± 0.013
IBL-HMM	0.881 ± 0.010

Table 3 The average recognition performance over 10 runs by different approaches

Methods	The EEG data set
GMM-HMM-VB [8]	0.839 ± 0.012
DMM-HMM-EM [16]	0.853 ± 0.017
IDMM-HMM-EM [17]	0.882 ± 0.015
IDMM-HMM-VB [26]	0.903 ± 0.021
IBL-HMM	0.932 ± 0.016

parameter values as in their original papers. All tested models were implemented on the same data sets as described in our experiments.

Tables 2 and 3 demonstrate the recognition performance by different models on the two real data sets. As we can see from these two tables, the proposed IBL-HMM with VB inference is able to outperform other HMM-based approaches with higher recognition accuracies for all data sets, which verified the merits of applying IBL-based HMMs for modeling activities and EEG data. We may also notice that HMM based on Gaussian mixture models (GMM-HMM-VB) has obtained the lowest recognition performance for both data sets, which verifies that the HMM with GMM emission densities is not a good choice for modeling positive sequential data.

5 Conclusion

In this chapter, we proposed a novel continuous HMM for modeling positive sequential observations, in which the emission distribution of each hidden state is distributed according to an IBL mixture model that has shown better capability for modeling positive data than other popular distributions (e.g., the Gaussian distribution). The proposed IBL-HMM was learned by theoretically developing a convergence-guaranteed algorithm based on VB inference. We demonstrated the advantages of our model by conducting experiments on real-world positive sequential data sets.

Acknowledgments The completion of this work was supported by the National Natural Science Foundation of China (61876068), the Natural Sciences and Engineering Research Council of Canada (NSERC), and the framework of the EquipEx program AmiQual4Home ANR-11-EQPX-00 and the cross disciplinary program Eco-SESA.

References

1. L. Pigou, A. V. Den Oord, S. Dieleman, M. Van Herreweghe, J. Dambre, Beyond temporal pooling: recurrence and temporal convolutions for gesture recognition in video. Int. J. Comput. Vision **126**, 430–439 (2018)
2. C.J.R. Illingworth, S. Roy, M.A. Beale, H.J. Tutill, R. Williams, J. Breuer, On the effective depth of viral sequence data. Virus Evol. **3**(2), vex030 (2017)
3. Z. Qiu, H. Shen, User clustering in a dynamic social network topic model for short text streams. Inf. Sci. **414**, 102–116 (2017)
4. A.B. Mabrouk, E. Zagrouba, Abnormal behavior recognition for intelligent video surveillance systems. Exp. Syst. Appl. **91**, 480–491 (2018)
5. L. Rabiner, B. Juang, An introduction to hidden Markov models. IEEE ASSP Mag. **3**(1), 4–16 (1986)
6. L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 267–296 (1989)
7. S. Volant, C. Berard, M. Martinmagniette, S. Robin, Hidden markov models with mixtures as emission distributions. Stat. Comput. **24**(4), 493–504 (2014)
8. S. Ji, B. Krishnapuram, L. Carin, Variational bayes for continuous hidden Markov models and its application to active learning. IEEE Trans. Pattern Anal. Mach. Intell. **28**(4), 522–532 (2006)
9. W. Fan, N. Bouguila, D. Ziou, Variational learning for finite Dirichlet mixture models and applications. IEEE Trans. Neural Netw. Learn. Syst. **23**(5), 762–774 (2012)
10. W. Fan, N. Bouguila, Online learning of a dirichlet process mixture of beta-liouville distributions via variational inference. IEEE Trans. Neural Netw. Learn. Syst. **24**(11), 1850–1862 (2013)
11. W. Fan, H. Sallay, N. Bouguila, Online learning of hierarchical pitman-yor process mixture of generalized dirichlet distributions with feature selection. IEEE Trans. Neural Netw. Learn. Syst. **28**(9), 2048–2061 (2017)
12. W. Fan, N. Bouguila, J. Du, X. Liu, Axially symmetric data clustering through Dirichlet process mixture models of Watson distributions. IEEE Trans. Neural Netw. Learn. Syst. **30**(6), 1683–1694 (2019)
13. J. Taghia, Z. Ma, A. Leijon, Bayesian estimation of the von mises-fisher mixture model with variational inference. IEEE Trans. Pattern Anal. Mach. Intell. **36**(9), 1701–1715 (2014)
14. T.M. Nguyen, Q.M.J. Wu, H. Zhang, Asymmetric mixture model with simultaneous feature selection and model detection. IEEE Trans. Neural Netw. Learn. Syst. **26**(2), 400–408 (2015)
15. S.P. Chatzis, D.I. Kosmopoulos, A variational bayesian methodology for hidden Markov models utilizing student's-t mixtures. Pattern Recognit. **44**(2), 295–306 (2011)
16. E. Epaillard, N. Bouguila, Variational bayesian learning of generalized dirichlet-based hidden Markov models applied to unusual events detection. IEEE Trans. Neural Netw. **30**(4), 1034–1047 (2019)
17. R. Nasfi, M. Amayri, N. Bouguila, A novel approach for modeling positive vectors with inverted dirichlet-based hidden markov models. Knowl. Based Syst. **192**, 105335 (2020)
18. W. Fan, L. Yang, N. Bouguila, Y. Chen, Sequentially spherical data modeling with hidden Markov models and its application to FMRI data analysis. Knowl. Based Syst. **206**, 106341 (2020)
19. W. Fan, N. Bouguila, Modeling and clustering positive vectors via nonparametric mixture models of liouville distributions. IEEE Trans. Neural Netw. Learn. Syst. **31**(9), 3193–3203 (2020)
20. T. Bdiri, N. Bouguila, Positive vectors clustering using inverted dirichlet finite mixture models. Exp. Syst. Appl. **39**(2), 1869–1882 (2012)
21. T. Bdiri, N. Bouguila, Bayesian learning of inverted dirichlet mixtures for SVM kernels generation. Neural Comput. Appl. **23**(5), 1443–1458 (2013)

22. M.A. Mashrgy, T. Bdiri, N. Bouguila, Robust simultaneous positive data clustering and unsupervised feature selection using generalized inverted Dirichlet mixture models. Knowl. Based Syst. **59**, 182–195 (2014)
23. T. Bdiri, N. Bouguila, D. Ziou, Variational Bayesian inference for infinite generalized inverted Dirichlet mixtures with feature selection and its application to clustering. Appl. Intell. **44**(3), 507–525 (2016)
24. M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, L.K. Saul, An introduction to variational methods for graphical models. Mach. Learn. **37**(2), 183–233 (1999)
25. D.M. Blei, A. Kucukelbir, J. Mcauliffe, Variational inference: a review for statisticians. J. Am. Stat. Assoc. **112**(518), 859–877 (2017)
26. R. Wang, W. Fan, Positive sequential data modeling using continuous hidden markov models based on inverted dirichlet mixtures. IEEE Access **7**, 172341–172349 (2019)
27. C.M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006)
28. L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**, 257–286 (1989)

Multivariate Beta-Based Hidden Markov Models Applied to Human Activity Recognition

Narges Manouchehri, Oumayma Dalhoumi, Manar Amayri, and Nizar Bouguila

1 Introduction

Human Activity Recognition (HAR) has become a vibrant research area and several studies have been conducted so far. Analyzing activities is complicated but valuable in our real life and it has been mainly used for important applications such as automated surveillance systems [1, 2], remote monitoring, healthcare [3–6], analyzing environments such as smart buildings [7–21]. Scientists have leveraged various approaches to obtain activity-related information and a noticeable amount of systems have been developed. In addition, there is a tremendous improvement in using smart technologies, and diverse electronic devices were introduced to our daily lives. This has led to generating a huge amount of data based on two mainstream systems: vision and sensor-based platforms [22–26]. Collecting vision-based data is relatively easy and such type of data has been initially used for activity recognition over the past decades. Working with these data may provide good results; however, we face some critical issues such as lack of privacy. On the other hand, due to the popularity of using low-priced sensors, capturing human behaviors and logging of daily data became so practical and common. A significant amount of research was conducted in the light of such a convenient solution. To collect sensor-based data, there are a variety of techniques such as deploying object-

N. Manouchehri · O. Dalhoumi · N. Bouguila (✉)
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: narges.manouchehri@mail.concordia.ca; o_dalhou@encs.concordia.ca; nizar.bouguila@concordia.ca

M. Amayri
Grenoble Institute of Technology, G-SCOP Lab, Grenoble, France
e-mail: manar.amayri@grenoble-inp.fr

N. Bouguila et al. (eds.), *Hidden Markov Models and Applications*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-99142-5_9

tagged, ambient, and wearable sensors [27–33]. Though using a sensing platform such as smartwatches, smartphones, or tagged objects may not always be feasible. For instance, we may simply forget to wear the device or miss-use an object. Thus, collecting data from the sensors that need human interaction is more challenging. This motivated us to focus on ambient sensors platforms [34–39] as they are independent of users during the procedure of data collection.

Analyzing human activity naturally carries some issues such as complexity of association of activities to multiple users, extraction of distinguishable features for a specific activity, differentiating similar activities (for instance walking and running), unexpected events that affect the activity (for example falling down during walking), generating an unknown class of activity as a result of an accidental event, diverse pattern and styles for a single activity even for a specific individual, high complexity of some activities (such as cooking that needs several actions), environmental noise, and difficulties of data annotation.

Thanks to the rapid progress of computational power and admirable development in numerous analytical methods, many machine learning techniques have been broadly applied to extract meaningful patterns and infer human activities in the past couple of decades [40]. For instance, deep learning methods are among the most attractive ones [41–43] and have shown outperformance compared to previous supervised machine learning techniques in several fields. However, these models need large-scale labeled data for training and evaluation. Annotation is a remarkable barrier as it is expensive and time-consuming. These characteristics encouraged us to focus on Hidden Markov Models (HMMs), which are known as powerful generative models specifically for temporal data. HMMs have demonstrated great potential in different fields and applications such as speech processing, anomaly detection, healthcare, facial identification, stock market and financial analysis and human activity recognition [44–55]. Considering several works in this domain, we realize that the typical choice of emission probability distribution follows Gaussian Mixture Models (GMM) [56–61]. In recent years, some researches have shown that this assumption of Gaussianity could not be generalized while working on various kinds of data. Thus, other distributions such as Dirichlet, generalized Dirichlet, and inverted Dirichlet distribution [62–65] have been applied as capable alternatives. This motivated us to conduct our research on multivariate Beta mixture models (MBMM), which has a high potential and flexibility to model the symmetric, asymmetric, and skewed data [66, 67]. Thus, we improved structure of HMM assuming that emission probabilities are raised from MBMM.

To estimate the parameter of our models including HMM and MBMM parameters, we apply two well-known methods: maximum likelihood (ML) approach and variational inference framework with adoption of Expectation–Maximization framework. Each of these methods has its own pros and cons. In ML, we determine the model parameters in such a way that they maximize the likelihood. However, this technique may lead to overfitting and convergence to a local maximum. To tackle such issues, full Bayesian inferring methods have been introduced to approximate

the likelihood. Despite achieving more precise results, these approaches are computationally expensive. In recent years, variational inference has been proposed, which is faster than fully Bayesian approaches and provides better results compared to ML [68].

The paper is organized as follows: in Sect. 2, we discuss HMM. Sections 3 and 4 are devoted to parameter estimation with ML and variational inference, respectively. In Sect. 5, we present the results of evaluating our proposed model in human activity recognition. We conclude in Sect. 6.

2 Hidden Markov Model

Hidden Markov Model (HMM) is generally applied in predicting hidden states using sequential data and changing systems such as weather patterns, speech, text, etc. It is specifically useful when we aim to compute the probability for a sequence of events that may not be directly visible in the world. To explain HMM, we express first Markov chain. Let us assume to have a sequence of events or states. Further to Markov property, a principal assumption in establishing first-order Markov Model is that future event or state depends only on the current event or state and not on any other previous states. To express mathematically, the probability of an event at a specific point of time t only depends on the event at time step $t-1$. This characteristic is one of the strengths of Markov Model. For instance, let us imagine that we would like to predict tomorrow's weather. Thus, we need to examine only today's weather and the previous day's data have no impact on our current prediction. In HMM the state of the system will be unknown or hidden; however, our system will emit a visible symbol at every particular time step t. These observable symbols are the only information that we have access to. To describe HMM, we explain following parameters:

- Transition probability: This is the probability of changing one state at time step t to another state or same state at time step $t+1$. A principle property is that all the transition probabilities given the current state sum up to 1.
- Initial Probability: At time step 0, the initial state of HMM that the system will start from is denoted as π. All probabilities sum up to 1.
- Emission Probability or observation likelihoods: These are the parameters expressing the probability of an observation generated from a specific state.

In this work, we use following notations to describe HMM:

- We assume to have an ordered observation sequence $\mathcal{X} = \{X_1, \ldots, X_T\}$ generated by hidden states $H = \{h_1, \ldots, h_T\}$ $h_j \in [1, K]$ such that K is the number of the states.
- Transition matrix: $B = \{b_{jj'} = P(h_t = j' \mid h_{t-1} = j)\}$. This shows the probabilities of transition between the states:

- Emission matrix: $C = \{C_{ij} = P(m_t = j \mid h_t = i)\}$ for $j \in [1, M]$ such that M is the number of mixture components associated with the state j.
- π_j: Initial probability to begin the observation sequence from the state j.

Thus, HMM is denoted by $\lambda = \{B, C, \varphi, \pi\}$ such that φ is the set of mixture parameters. In our work, we will apply multivariate Beta mixture model that has one shape parameter. In HMM, we need to tackle three problems:

1. Evaluation problem: Given the model parameter λ, and a sequential dataset represented by $\mathcal{X}$, we need to find the likelihood of $p(\mathcal{X} \mid \lambda)$.
2. Learning problem: In HMM as an unsupervised learning method, number of visible symbol is known and number of hidden states is unknown. In learning process, we try to find the best set of state transitions and emission probabilities through Expectation Maximization (EM) algorithm. This process is called Forward–Backward or Baum–Welch algorithm.
3. Decoding problem: After having the estimations for transition and emission probabilities, we can then use model parameters to predict hidden states that generated the observable sequence. This decoding process is known as Viterbi Algorithm.

Emission probability distributions in HMM are commonly assumed to follow Gaussian mixtures [56, 60, 69–74]. In this work, we construct HMM using multivariate Beta mixture model as emission probabilities. Our motivation behind this choice is flexibility of multivariate Beta distribution (MB) and its potential to model different-shaped data [75, 76]. To describe it, we assume first to have a D-dimensional vector $\mathbf{X} = (x_1, \ldots, x_d)$ drawn from a MB distribution with following probability density function where $0 < x_d < 1$ and $\Gamma(.)$ indicates the Gamma function:

$$\mathcal{MB}(\mathbf{X} \mid \boldsymbol{\alpha}) = \frac{\Gamma(\mid \alpha \mid) \prod_{d=1}^{D} x_d^{\alpha_d - 1}}{\prod_{d=0}^{D} \Gamma(\alpha_d) \prod_{d=1}^{D} (1 - x_d)^{(\alpha_D + 1)}} \left[1 + \sum_{d=1}^{D} \frac{x_d}{(1 - x_d)}\right]^{-|\alpha|} \tag{1}$$

$\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_D)$ is the shape parameter where $\alpha_d > 0$ for $d = 0, \ldots, D$ and $\mid \alpha_j \mid = \sum_{d=0}^{D} \alpha_d$. Figure 1 illustrates some examples of this distribution with various parameters.

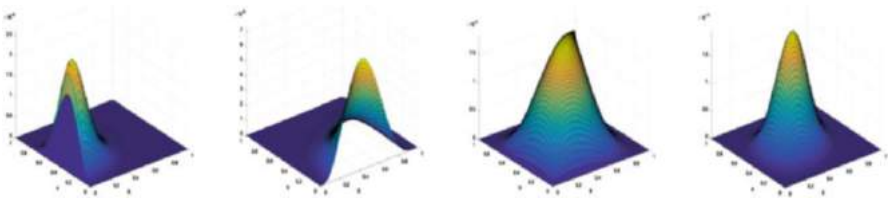


Fig. 1 Multivariate Beta distribution

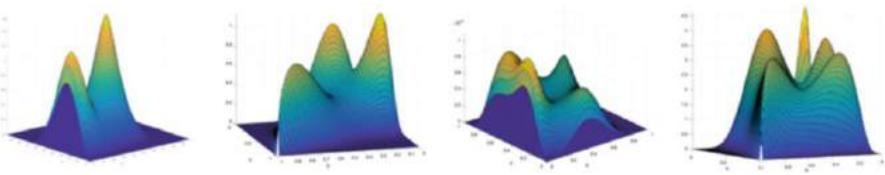


Fig. 2 Multivariate Beta mixture model with multiple components

Figure 2 illustrates some examples of finite multivariate Beta mixture model [77–79] with multiple components.

By changing the emission probability distribution, some modifications in the expectation maximization (EM) estimation process happen and we set following notations for the quantities:

$$\gamma^t_{h_t,m_t} \triangleq p(h_t, m_t \mid x_0, \ldots, x_T) \tag{2}$$

which represent the estimates of the states and mixture components and,

$$\xi^t_{ht,ht+1} \triangleq p(h_t, h_{t+1} \mid x_0, \ldots, x_T) \tag{3}$$

for representing the local states sequence given the whole observation set. In expectation step (E-step), we estimate $\gamma^t_{h_t,m_t}$ and $\xi^t_{h_t,h_r+1}$ for all $t \in [1, T]$. These quantities are obtained with the initial parameters at E-step and the result of the maximization step (M-step) subsequently. We compute them with a similar forward–backward procedure as for HMM with mixtures of Gaussians. In M-step, we aim to maximize the data log-likelihood by maximizing its lower bound. We represent Z and X as the hidden variables and data, respectively. The data likelihood $\mathcal{L}(\theta \mid \mathbf{X}) = p(\mathbf{X} \mid \theta)$ is expressed by

$$\begin{aligned} E(\mathbf{X}, \theta) - R(Z) &= \sum_Z p(Z \mid \mathbf{X}) \log(p(\mathbf{X}, Z)) - \sum_Z p(Z \mid \mathbf{X}) \log(p(Z \mid \mathbf{X})) \\ &= \sum_Z p(Z \mid \mathbf{X}) \log(p(\mathbf{X} \mid \theta)) = \log(p(\mathbf{X} \mid \theta)) \sum_Z p(Z \mid \mathbf{X}) \\ &= \log(p(\mathbf{X} \mid \theta)) = \mathcal{L}(\theta \mid \mathbf{X}) \end{aligned} \tag{4}$$

where θ represents all the HMM parameters. $E(\mathbf{X}, \theta)$ is the value of the complete-data log-likelihood with the maximized parameters θ. $R(Z)$ is the log-likelihood of hidden data given the observations.

The expected complete-data log-likelihood is defined as follows:

$$E(\mathbf{X}, \theta, \theta^{\text{old}}) = \sum_{Z} p(Z \mid X, \theta^{\text{old}}) \log(p(\mathbf{X}, Z \mid \theta)) \tag{5}$$

In this part, we explain first the case of a unique observation $\mathbf{X}$ and then we extend it to the whole dataset $\mathbf{X}$.

The complete-data likelihood for $\mathbf{X}$ can be expressed by

$$p(\mathbf{X}, Z \mid \theta) = p(h_0) \prod_{t=0}^{T-1} p(h_{t+1} \mid h_t) \prod_{t=0}^{T} p(m_t \mid h_t) p(x_t \mid h_t, m_t) \tag{6}$$

The different terms of the expression are identified as follows:

$$p(\mathbf{X}, Z \mid \theta) = \pi_{h_0} \prod_{t=0}^{T-1} B_{h_t, h_{t+1}} \prod_{t=0}^{T} C_{h_t, m_t} \mathcal{MB}(x_t \mid h_t, m_t) \tag{7}$$

As we assume that MB distribution is emission probability, we substitute it in Eq. (7) and after applying logarithm to the expression, we get the complete-data log-likelihood:

$$\begin{aligned} \log(p(\mathbf{X}, Z \mid \theta)) &= \log(\pi_{h_0}) + \sum_{t=0}^{T-1} \log(B_{h_t, h_{t+1}}) + \sum_{t=0}^{T} \log(C_{h_t, m_t}) + \\ &+ \sum_{t=0}^{T} \log\left(\Gamma\left(\sum_{d=0}^{D} \alpha_d\right)\right) - \log\left(\prod_{d=0}^{D} \Gamma(\alpha_d)\right) + \sum_{d=1}^{D} \left((\alpha_d - 1) \log x_d\right) \\ &- \sum_{d=1}^{D} \left((\alpha_d + 1) \log(1 - x_d)\right) - \left(\sum_{d=0}^{D} \alpha_d\right) \log\left[1 + \sum_{d=1}^{D} \frac{x_d}{1 - x_d}\right] \end{aligned} \tag{8}$$

The expected complete-data log-likelihood can then be written:

$$\begin{aligned} E(X, \theta, \theta^{\text{old}}) &= \sum_{k=1}^{K} \sum_{m=1}^{M} \gamma_{k,m}^{0} \log(\pi_k) + \sum_{t=0}^{T} \sum_{k=1}^{K} \sum_{m=1}^{M} \gamma_{k,m}^{t} \log(C_{k,m}) \\ &+ \sum_{t=0}^{T-1} \sum_{i=1}^{K} \sum_{j=1}^{K} \xi_{i,j}^{t} \log(B_{i,j}) + \log \mathcal{MB}(\mathbf{X} \mid \boldsymbol{\alpha}) \end{aligned} \tag{9}$$

$$\log \mathcal{MB}(\mathbf{X} \mid \boldsymbol{\alpha}) = \sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{k=1}^{K}\sum_{m=1}^{M}\gamma_{k,m}^{t}\Bigg[\log\Bigg(\Gamma\Bigg(\sum_{d=0}^{D}\alpha_{k,m,d}\Bigg)\Bigg)$$
$$-\log\Bigg(\prod_{d=0}^{D}\Gamma\Big(\alpha_{k,m,d}\Big)\Bigg)+\sum_{d=1}^{D}\Big((\alpha_{k,m,d}-1)\log x_d\Big) \tag{10}$$
$$-\sum_{d=1}^{D}\Big((\alpha_{k,m,d}+1)\log(1-x_d)\Big)-\Bigg(\sum_{d=0}^{D}\alpha_{k,m,d}\Bigg)\log\Bigg[1+\sum_{d=1}^{D}\frac{x_d}{1-x_d}\Bigg]\Bigg]$$

For the dataset with more than one sequential observations, a sum over $n \in [1, N]$, $N \geq 1$ has to be added in front of Eq. (9). The sum over time goes from 0 to T_n. Length of the observations is n and x_d changes to x_{nd}.

3 HMM Parameters Estimation with Maximum Likelihood

The maximization of expectation of complete-data log-likelihood with respect to π, B, and C results in following updated equations:

$$\pi_k^{new} \propto \sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_{k,m}^{0,n} \tag{11}$$

$$B_{k,k'}^{new} \propto \sum_{n=1}^{N}\sum_{t=0}^{T_n-1}\xi_{k,k'}^{t,n} \tag{12}$$

$$C_{k,m}^{new} \propto \sum_{n=1}^{N}\sum_{t=0}^{T_n}\gamma_{k,m}^{t,n} \tag{13}$$

where $k, k' = \{1, \ldots, K\}$, and $m = \{1, \ldots, M\}$.

To estimate the parameter of $\log \mathcal{MB}(\mathbf{X} \mid \boldsymbol{\alpha})$ in Eq. (10), we use EM algorithm [77]. To tackle this problem, we need to find a solution to the following equation:

$$\frac{\partial \log \mathcal{MB}\Big(\mathbf{X} \mid \boldsymbol{\alpha}\Big)}{\partial \alpha_k, m, d} = \gamma_{k,m}^{t}\Psi_0\Bigg(\sum_{d=0}^{D}\alpha_{k,m,d}\Bigg) - \gamma_{k,m}^{t}\Psi_0(\alpha_{k,m,d}) \tag{14}$$
$$-\log\Bigg[1+\sum_{d=1}^{D}\frac{x_{ld}}{1-x_{ld}}\Bigg]\Bigg] = 0$$

where $\Psi_0(.)$ is the digamma function.

As there is no closed-form solution to estimate our parameters, we apply Newton–Raphson method as an iterative technique to maximize $\log \mathcal{MB}(\boldsymbol{\alpha})$.

The global estimation equation is given by

$$\theta^{\text{new}} = \theta^{\text{old}} - H(\theta^{\text{old}})^{-1} \frac{\partial \mathcal{L}(X \mid \theta^{\text{old}})}{\partial \theta^{\text{old}}} \tag{15}$$

H is the Hessian matrix associated with $\log \mathcal{MB}(\mathbf{X} \mid \boldsymbol{\alpha})$ and G is the first derivatives vector defined by

$$G = \left(\frac{\partial \log \mathcal{MB}(\mathbf{X} \mid \boldsymbol{\alpha})}{\partial \alpha_{k,m,d}}, \ldots, \frac{\partial \log \mathcal{MA}(\mathbf{X} \mid \boldsymbol{\alpha})}{\partial \alpha_{k,m,D}} \right)^T \tag{16}$$

The Hessian of $\log \mathcal{MB}(\mathbf{X} \mid \boldsymbol{\alpha})$ is calculated with the second derivatives:

$$\frac{\partial^2 \log \mathcal{MB}(\mathbf{X} \mid \boldsymbol{\alpha})}{\partial^2 \alpha_{k,m,D}} = \gamma_{k,m}^t \Psi_1 \left(\sum_{d=0}^{D} \alpha_{k,m,D} \right) - \gamma_{k,m}^t \Psi_1(\alpha_{k,m,D}) \tag{17}$$

$$\frac{\partial^2 \log \mathcal{MB}(\mathbf{X} \mid \boldsymbol{\alpha})}{\partial \alpha_{k,m,d_1} \partial \alpha_{k,m,d_2}} = \gamma_{k,m}^t \Psi_1 \left(\sum_{d=0}^{D} \alpha_{k,m} \right) \tag{18}$$

where $\Psi_1(.)$ is the trigamma function.

$$H = \bar{\gamma} \times \begin{pmatrix} \Psi_1\left(\sum_{d=0}^{D} \alpha_d\right) - \Psi_1(\alpha_1) & \Psi_1\left(\sum_{d=0}^{D} \alpha_d\right) & \cdots & \Psi_1\left(\sum_{d=0}^{D} \alpha_d\right) \\ \Psi_1\left(\sum_{d=0}^{D} \alpha_d\right) & \Psi_1\left(\sum_{d=0}^{D} \alpha_d\right) - \Psi_1(\alpha_2) & \cdots & \Psi_1\left(\sum_{d=0}^{D} \alpha_d\right) \\ \vdots & \ddots & & \vdots \\ \Psi_1\left(\sum_{d=0}^{D} \alpha_d\right) & \cdots & & \Psi_1\left(\sum_{d=0}^{D} \alpha_d\right) - \Psi_1(\alpha_D) \end{pmatrix} \tag{19}$$

Thus, H can be written as follows:

$$H = D + \delta A A^T \tag{20}$$

where D is a diagonal matrix,

$$D = \text{diag}[-\bar{\gamma} \Psi_1(\alpha_D)] \tag{21}$$

and,

$$\delta = \bar{\gamma}\Psi_1\left(\sum_{d=0}^{D}\alpha_d\right) \tag{22}$$

$$A^T = (a_0, \ldots, a_D), d = 0, \ldots, D \tag{23}$$

Thus, we have

$$H^{-1} = D^{-1} + \delta^* A^{*^T} A^* \tag{24}$$

where D^{-1} is computed as follows:

$$A^* = \frac{-1}{\bar{\gamma}}\left(\frac{1}{\Psi_1(\alpha_1)}, \ldots, \frac{1}{\Psi_1(\alpha_D)}\right) \tag{25}$$

$$\delta^* = \bar{\gamma}\Psi_1\left(\sum_{d=1}^{D}\alpha_d\right)\left[\Psi_1\left(\sum_{d=1}^{D}\alpha_d\right)\sum_{d=1}^{D}\frac{1}{\Psi_1(\alpha_d)} - 1\right] \tag{26}$$

$\bar{\gamma} = \sum_{d=1}^{D}\sum_{t=1}^{T}\gamma^{d,t}$ is the cumulative sum to the state estimates of the observation sequence. After having H^{-1} and G, we update the parameters of the MB mixture model. We monitor data likelihood and whenever there is no or minor change and less than a threshold, we achieve convergence. As the data log-likelihood is maximized with its lower bound, convergence of this bound can help us to stop iterations. This lower bound is given by $E(X, \theta, \theta^{\text{old}}) - R(Z)$ in Eq. (5). $R(Z)$ is derived using Bayes rule:

$$p(Z \mid X) = p(h_0)p(m_0 \mid h_0)\prod_{t=1}^{T} p(h_t \mid h_{t-1})p(m_t \mid h_t) \tag{27}$$

$$= p(h_0)\frac{p(m_0, h_0)}{p(h_0)}\prod_{t=1}^{T}\frac{p(h_t, h_{t-1}p(m_t, h_t))}{p(h_{t-1})p(h_t)}$$

We denote $\eta_t \triangleq p(h_t \mid \mathbf{x})$ and $R(Z)$ is

$$R(Z) = \sum_{k=1}^{K}\left[\eta_k^0 \log(\eta_k^0) + \eta_k^T \log(\eta_k^T) - 2\sum_{t=0}^{T}\eta_k^t \log(\eta_k^t)\right]$$

$$+ \sum_{t=0}^{T}\sum_{m=1}^{M}\sum_{k=1}^{K}\gamma_{k,m}^t \log(\gamma_{k,m}^t) + \sum_{t=0}^{T-1}\sum_{k=1}^{K}\sum_{k'=0}^{K}\xi_{i,j}^t \log(\xi_{i,j}^t) \tag{28}$$

4 HMM Parameters Estimation with Variational Inference

In this section, we will discuss variational approach in which we consider all the parameters of HMM (such as transition matrix, the parameters of the emission distributions, mixing matrix, initial state, vector coefficients) as random variables. So, we assume a prior distribution for each of them. The likelihood of a sequence of observations $\mathbf{X}$ given the model parameters is defined as follows such that S is the set of hidden states and L the set of mixtures' components:

$$p(\mathbf{X} \mid A, C, \pi, \alpha) = \sum_{S} \sum_{L} \pi_{s_1} \left[\prod_{t=2}^{T} a_{s_{t-1}, s_t} \right] \left[\prod_{t=1}^{T} c_{s_t, m_t} p(x_t \mid \alpha_{s_t, m_t}) \right] \tag{29}$$

Using the complete-data likelihood, we have

$$p(\mathbf{X}) = \int d\pi dA dC d\alpha \sum_{S,L} p(A, C, \pi, \alpha) p(\mathbf{X}, S, L \mid A, C, \pi, \alpha) \tag{30}$$

As this quantity is intractable, we apply a lower bound by introducing an approximating distribution $q(A, C, \pi, \alpha, S, L)$ of $p(A, C, \pi, \alpha, S, L \mid \mathbf{X})$, which is the true posterior. Thus,

$$\begin{aligned} \ln(p(\mathbf{X})) &= \ln \left\{ \int dA dC d\pi d\alpha \sum_{S,L} p(A, C, \pi, \alpha) p(\mathbf{X}, S, L \mid A, C, \pi, \alpha) \right\} \\ &\geq \int d\pi dA dC d\alpha \sum_{S,L} q(A, C, \pi, \alpha, S, L) \ln \left\{ \frac{p(A, C, \pi, \alpha) p(\mathbf{X}, S, L \mid A, C, \pi, \alpha)}{q(A, C, \pi, \alpha, S, L)} \right\} \end{aligned} \tag{31}$$

Considering Jensen's inequality and recalling that $KL(q \mid\mid p) \geq 0$, we have $KL(q \mid\mid p) = 0$ when q equals the true posterior where KL is the Kullback–Leibler distance between the true posterior and its approximate distribution. $\mathcal{L}(q)$ could be considered as a lower bound to $\ln p(\mathbf{X})$ such that:

$$\ln(p(\mathbf{X})) = \mathcal{L}(q) - \mathrm{KL}(q(A, C, \pi, \alpha, S, L) \| p(A, C, \pi, \alpha, S, L \mid \mathbf{X})) \tag{32}$$

The true posterior distribution is not computationally tractable. So, we consider a restricted family of distributions with the help of mean field theory and we can write q in a factorized form: $q(A, C, \pi, \alpha, S, L) = q(A)q(C)q(\pi)q(\alpha)q(S, L)$. Thus, the lower bound can be defined by

$$\ln(p(\mathbf{X})) \geq \sum_{S,L} \int dA dC d\pi d\alpha q(\pi) q(A) q(C) q(\alpha) q(S, L) \{\ln(p(\pi)) + \ln(p(A))$$

$$+\ln(p(C))+\ln(p(\alpha))+\ln(\pi_{s_1})+\sum_{t=2}^{T}\ln(a_{s_{t-1},s_t})+\sum_{t=1}^{T}\ln(c_{s_t,m_t})$$
$$+\sum_{t=1}^{T}\ln(p(x_t\mid\alpha_{s_t,m_t}))-\ln(q(S,L))-\ln(q(\pi))-\ln(q(A))-\ln(q(C)) \tag{33}$$
$$-\ln(q(\alpha))\}=F(q(\pi))+F(q(C))+F(q(A))+F(q(\alpha))+F(q(S,L))$$

We need to define the priors of all HMM parameters. A natural choice for the prior of parameters A, C, and π is the Dirichlet distribution as all the coefficients of these matrices and vector are strictly positive, less than 1, with each row summing up to one.

$$p(\pi)=\mathcal{D}(\pi\mid\phi^{\pi})=\mathcal{D}(\pi_1,\ldots,\pi_K\mid\phi_1^{\pi},\ldots,\phi_K^{\pi})$$
$$p(A)=\prod_{i=1}^{K}\mathcal{D}(a_{i_1},\ldots,a_{i_K}\mid\phi_{i_1}^{A},\ldots,\phi_{i_K}^{A})$$
$$p(C)=\prod_{i=1}^{M}\mathcal{D}(c_{i_1},\ldots,c_{i_M}\mid\phi_{i_1}^{C},\ldots,\phi_{i_M}^{C}) \tag{34}$$

For α as the shape parameter of multivariate Beta distribution, we define a conjugate prior that can be expressed as follows:

$$p(\alpha)=f(\nu,\mu)[\frac{\Gamma(\sum_{l=1}^{D}\alpha_l)}{\prod_{l=1}^{D}\Gamma(\alpha_l)}]^{\nu}\prod_{l=1}^{D}e^{-\mu_l(\alpha_l-1)} \tag{35}$$

where $f(\nu,\mu)$ is a normalization coefficient and (ν,μ) are hyperparameters. As evaluation of the normalization coefficient is difficult, this prior is intractable and we approximate it with the Gamma distribution $\mathcal{G}$ expressed as follows:

$$p(\alpha_{ijl})=\mathcal{G}(\alpha_{ijl}\mid u_{ijl},v_{ijl})=\frac{v_{ijl}^{u_{ijl}}}{\Gamma(u_{ijl})}\alpha_{ijl}^{u_{ijl}-1}e^{-v_{ijl}\alpha_{ijl}} \tag{36}$$

where $l\in[1,D], i\in[1,K]$ and $j\in[1,M]$ and u and v are strictly positive hyperparameters.

$$p(\{\alpha_{ij}\}_{i,j=1}^{K,M})=\prod_{l=1}^{D}\prod_{i=1}^{K}\prod_{j=1}^{M}\frac{v_{ijl}^{u_{ijl}}}{\Gamma(u_{ijl})}\alpha_{ijl}^{u_{ijl}-1}e^{-v_{ijl}\alpha_{ijl}} \tag{37}$$

The optimization of $q(A), q(C)$, and $q(\pi)$ are independent from the emission distributions and common to other continuous HMM.

$$F(q(A)) = \int dA q(A) \ln \left[\frac{\prod_{i=1}^{K} \prod_{j=1}^{K} a_{ij}^{w_{ij}^{A}-1}}{q(A)} \right] \tag{38}$$

$$w_{ij}^{A} = \sum_{t=2}^{T} \gamma_{ijt}^{A} + \phi_{ij}^{A} \tag{39}$$

$$\gamma_{ijt}^{A} \triangleq q(s_{t-1} = i, s_t = j) \tag{40}$$

$$q(A) = \prod_{i=1}^{K} \mathcal{D}(a_{i1}, \ldots, a_{iK} \mid w_{i1}^{A}, \ldots, w_{iK}^{A}) \tag{41}$$

$$q(\pi) = \mathcal{D}(\pi_1, \ldots, pi_K \mid w_1^{\pi}, \ldots, w_K^{\pi}) \tag{42}$$

$$w_i^{\pi} = \gamma_i^{\pi} + \phi_i^{\pi} \tag{43}$$

$$\gamma_i^{\pi} \triangleq q(s_1 = i) \tag{44}$$

$$q(C) = \prod_{i=1}^{K} \mathcal{D}(c_{i1}, \ldots, c_{iM} \mid w_{i1}^{C}, \ldots, w_{iM}^{C}) \tag{45}$$

$$w_{ij}^{C} = \sum_{t=1}^{T} \gamma_{ijt}^{C} + \phi_{ij}^{C} \tag{46}$$

$$\gamma_{ijt}^{C} \triangleq q(s_t = i, m_t = j) \tag{47}$$

For optimizing $F(q(\alpha))$, we have

$$F(q(\alpha)) = \int d\alpha q(\alpha) \ln \left\{ \frac{\prod_{i=1}^{K} \prod_{j=1}^{M} p(\alpha_{ij}) \prod_{t=1}^{T} p(x_t \mid \alpha_{ij})^{\gamma^{c}{}_{ijt}}}{q(\alpha)} \right\} \tag{48}$$

The log-evidence maximization is given by

$$q(\alpha) = \prod_{i=1}^{K}\prod_{j=1}^{M} q(\alpha_{ij}), \quad q(\alpha_{ij}) = \prod_{l=1}^{D} \mathcal{G}(\alpha_{ijl} \mid u^{\star}_{ijl}, v^{\star}_{ijl}) \tag{49}$$

$$u^{\star}_{ijl} = u_{ijl} + \mathcal{U}_{ijl}, \quad v^{\star}_{ijl} = v_{ijl} - \mathcal{V}_{ijl} \tag{50}$$

$$\begin{aligned}\mathcal{U}_{ijl} = \sum_{p=1}^{P} \langle Z_{pij}\rangle \bar{\alpha}_{ijl} \Bigg[\Psi\Bigg(\sum_{d=1}^{D} \bar{\alpha}_{ijd}\Bigg) - \Psi(\bar{\alpha}_{ijl}) \\ + \sum_{d=1, d\neq l}^{D} \Psi'\Bigg(\sum_{d=1}^{D} \bar{\alpha}_{ijd}\Bigg) \bar{\alpha}_{ijd}(\langle \ln(\alpha_{ijd})\rangle - \ln(\bar{\alpha}_{ijd}))\Bigg]\end{aligned} \tag{51}$$

$$\mathcal{V}_{ijl} = \sum_{p=1}^{P} \langle Z_{pjd}\rangle \Bigg[\ln x_{pl} - \ln(1 - x_{pl}) - \ln\Bigg[1 + \sum_{l=1}^{D} \frac{x_{il}}{(1 - x_{pl})}\Bigg]\Bigg] \tag{52}$$

The value of $Z_{pij} = 1$ if X_{pt} belongs to state i and mixture component j and zero, otherwise. Thus, $\langle Z_{pij}\rangle = \sum_{t=1}^{T} \gamma^{C}_{pijt} = p(s = i, m = j \mid X)$ and we compute responsibilities through a simple forward–backward procedure [80]. In the E-step, the parameters that were estimated in previous step are kept fixed and $q(S, L)$ is estimated. We rewrite $\mathcal{L}(q)$ [81] as follows:

$$\mathcal{L}(q) = F(q(S, L)) - \mathrm{KL}(q(A, C, \pi, \alpha) \mid p(A, C, \pi, \alpha)) \tag{53}$$

where

$$\begin{aligned}F(q(S, L)) = \sum_{S} q(S) \int q(\pi) \ln(\pi_{s_1}) d\pi + \sum_{S} q(S) \int q(A) \sum_{t=2}^{T} \ln(a_{s_{t-1}, s_t}) dA \\ + \sum_{S,L} q(S, L) \int q(C) \sum_{t=1}^{T} \ln(c_{s_t, m_t}) dC \\ + \sum_{S,L} q(S, L) \int q(\alpha) \sum_{t=1}^{T} \ln(f(x_t \mid \alpha_{s_t, m_t})) d\alpha - \sum_{S,L} q(S, L) \ln(q(S, L))\end{aligned} \tag{54}$$

and the second term is fixed in this E-step. Thus, we have

$$\pi^{\star}_i \triangleq \exp[\langle \ln(\pi_i)\rangle_{q(\pi)}] = \exp[\Psi(w^{\pi}_i) - \Psi\Bigg(\sum_{i} w^{\pi}_i\Bigg)]$$

$$a^{\star}_{jj'} \triangleq \exp[\langle \ln(a_{jj'})\rangle_{q(A)}] = \exp[\Psi(w^{A}_{jj'}) - \Psi\left(\sum_{j'} w^{A}_{jj'}\right)]$$

$$c^{\star}_{ij} \triangleq \exp[\langle \ln(c_{ij})\rangle_{q(C)}] = \exp[\Psi(w^{C}_{ij}) - \Psi\left(\sum_{j} w^{C}_{ij}\right)] \tag{55}$$

Ψ and $\langle . \rangle$ indicate the Digamma function and expectation with respect to the quantity indicated as a subscript, respectively. Then, we optimize the following quantity:

$$\ln(p^{\star}(X_t \mid \alpha_{s_t,l_t})) = \int q(\alpha) \ln(p(X_t \mid \alpha_{s_t,l_t}))d\alpha \tag{56}$$

where

$$p\left(X_t \mid \alpha_{s_t,l_t}\right) = \left[\frac{\Gamma\left(\sum_{l=0}^{D} \alpha_{ijl}\right) \prod_{l=1}^{D} x_{tl}^{\alpha_{jl}-1}}{\prod_{d=0}^{D} \Gamma\left(\alpha_{\alpha_{ijl}}\right) \prod_{l=1}^{D} (1 - x_d)\,(\alpha_D) + 1}\right.$$

$$\left. \times \left[1 + \sum_{l=1}^{D} \frac{x_{tl}}{(1 - x_{tl})}\right]^{-\left(\sum_{l=0}^{D} \alpha_{ijl}\right)}\right]^{\gamma^{C}_{ijt}} \tag{57}$$

Then, we substitute Eq. (57) in Eq. (56). We discussed in detail the variational inference for the case of multivariate Beta mixture models [67] and similar to our previous works, we have

$$\bar{\alpha}_{ijl} = \frac{u_{ijl}}{v_{ijl}}, \quad \langle \ln(\alpha_{ijd})\rangle = \Psi(u_{ijd}) - \ln(v_{ijd}) \tag{58}$$

The optimized $q(S, L)$ is defined by

$$q(S, L) = \frac{1}{W} \pi^{\star}_{s_1} \prod_{t=2}^{T} a^{\star}_{s_{t-1},s_t} \prod_{t=1}^{T} c^{\star}_{s_t,l_t} p^{\star}(X_t \mid \theta_{s_t,l_t}) \tag{59}$$

where W as the normalizing constant is

$$W = \sum_{S,L} \pi^{\star}_{s_1} \prod_{t=2}^{T} a^{\star}_{s_{t-1},s_t} \prod_{t=1}^{T} c^{\star}_{s_t,l_t} p^{\star}(X_t \mid \theta_{s_t,l_t}) \tag{60}$$

W is the likelihood of the optimized model $(A^{\star}, C^{\star}, \pi^{\star}, \alpha^{\star}, S, L)$ and we can compute it via a forward–backward algorithm [80, 81]. The variational Bayesian learning of the MB-based HMM is presented in Algorithm 1.

Algorithm 1 Variational learning of MB-based distributions

1. $\phi^A = \text{ones}(1, K) \times 1/K$, $\phi^C = \text{ones}(1, M) \times 1/M$, $\phi^\pi = \text{ones}(1, K) \times 1/M$
2. $v_{ijl} = 1$, $u_{ijl} = \alpha_{init_{ijl}}$, for all i, j, l
3. Draw the initial responsibilities γ^A, γ^C, and γ^π from prior distributions with Eq. (34).
4. Compute w^A, w^C, and w^π with Eqs. (39), (43) and (46).
5. Initialize A, C, and π with coefficients computed with Eq. (55)
6. hlik $k^{\text{old}} = 10^6$; hlik $^{\text{new}} = 10^5$; iter $= 0$
7. while | hlik $^{\text{old}}$ − hlik $^{\text{new}}$ |$\geq$ tol & iter $\leq$ maxIter do
8. Compute data likelihood dlik using X, u, v, and α_{init} with Eqs. (1) and (58).
9. Compute responsibilities γ^A, γ^C and γ^π with forward–backward procedure using dlik, A, C, and π. Eqs. (40), (44) and (47).
10. Update u and v with Eqs. (50) to (52).
11. Update w^A, w^C, and w^π using responsibilities γ^A, γ^C, and γ^π with Eqs. (39), (43) and (46). Update A, C, and π using w^A, w^C, and w^π with Eq. (55).
12. stopping criteria Compute hlik $^{\text{new}}$ with Eq. (60) and forward–backward procedure
13. iter $+ = 1$

5 Experimental Results

To evaluate our proposed methodology, we selected datasets including ambient sensors-based samples. We used a dataset called opportunity. In this publicly available resource [82, 83], information was collected with three types of sensors including external and wearable sensors. Figure 3 shows the setup and some of the sensors fixed in points of interest or attached to volunteer users. We will test our proposed algorithm just on a part of data collected by ambient sensors. This system was able to recognize activities of different levels such as:

- Modes of locomotion sit, stand, lie, walk, idle (no activity).
- 5 high-level activity classes.
- 17 mid-level gesture classes (e.g., open/close door/fridge/dishwasher, drink).
- Low-level action classes relating 13 actions to 23 objects.

The ambient sensors include 13 switches and 8 3D acceleration sensors in kitchen appliances and furniture. Data were collected from 4 volunteers with 6 runs of experiment for each one of individuals (5 activities of daily life and one drill run, which is a scripted sequence of activities). In our test, we focused on four actions: standing, walking, lying, and sitting of four individuals in drill mode. Figure 4 illustrates the frequencies of activities for each individual. While modeling these four datasets, we faced some issues. The first challenge was having missing data points and we replaced them with the median of each feature in all 4 datasets. As shown in Fig. 4, all datasets are unbalanced, which results in biased inference. So, we applied Synthetic Minority Over-sampling Technique (SMOTE) to have balanced datasets. Considering the various ranges of features, we normalized data with Min-Max scaling method as follows:

Fig. 3 Setup, wearable, object, and ambient sensors

$$\mathbf{X} = \frac{\mathbf{X} - \mathbf{X}_{min}}{\mathbf{X}_{max} - \mathbf{X}_{min}} \tag{61}$$

We compared the performance of three models: MB-based HMM with ML (MB-HMM-ML) and variational (MB-HMM-VR) approach as well as Gaussian

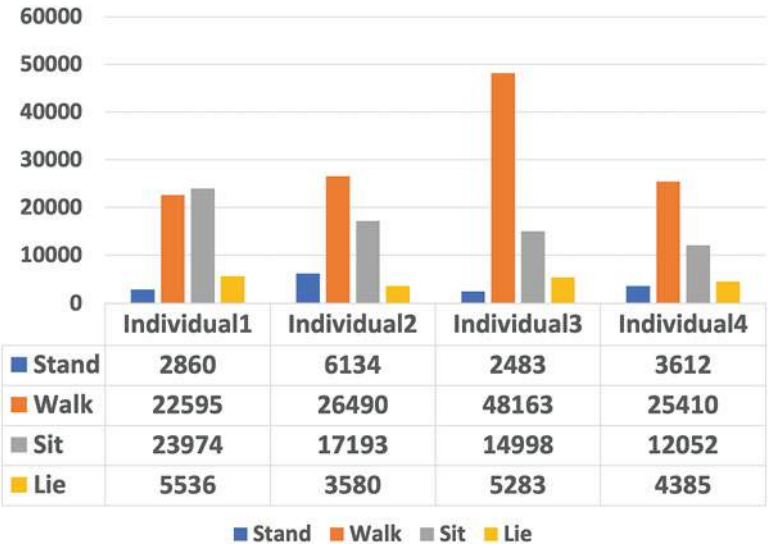


Fig. 4 Frequencies of activities for four individuals

mixture model HMM (CHH-HMM). To evaluate the performance of our proposed model, we applied following metrics. TP, TN, FP, and FN represent the total number of true positives, true negatives, false positives, and false negatives, respectively.

$$\begin{aligned} Accuracy &= \frac{TP + TN}{\text{Total number of observations}} \\ Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \\ F1 - score &= \frac{2 \times precision \times recall}{precision + recall} \end{aligned} \tag{62}$$

The results of evaluation are presented in Figs. 5 and 6. As it is shown, MB-HMM-VR has the highest F1-score in all four cases with 87.39%, 88.53%, 91.64%, and 86.34%. In each dataset, the results of MB-MHH-VR outperform the ones of other two models. This promising outcome indicates the capability of our proposed model.

In Fig. 6, the size of data in each cluster (after SMOTE over-sampling) is shown, and it is multiplied by 4 as we consider equal samples for each cluster and the total observation in each dataset is demonstrated also. In the case of individual 3, we have

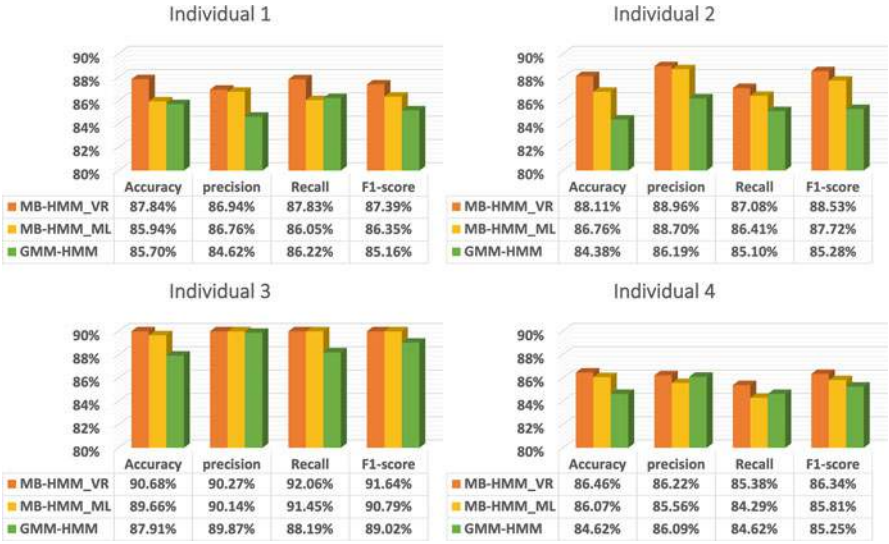


Fig. 5 Model performance evaluation

	Number of observation	Model	Accuracy	precision	Recall	F1-score
Individual 1	23974 * 4 = 95896	MB-HMM-VR	87.84%	86.94%	87.83%	87.39%
		MB-HMM-ML	85.94%	86.76%	86.05%	86.35%
		GMM-HMM	85.70%	84.62%	86.22%	85.16%

	Number of observation	Model	Accuracy	precision	Recall	F1-score
Individual 2	26490 * 4 = 105960	MB-HMM-VR	88.11%	88.96%	87.08%	88.53%
		MB-HMM-ML	86.76%	88.70%	86.41%	87.72%
		GMM-HMM	84.38%	86.19%	85.10%	85.28%

	Number of observation	Model	Accuracy	precision	Recall	F1-score
Individual 3	48163 * 4 = 192652	MB-HMM-VR	90.68%	90.27%	92.06%	91.64%
		MB-HMM-ML	89.66%	90.14%	91.45%	90.79%
		GMM-HMM	87.91%	89.87%	88.19%	89.02%

	Number of observation	Model	Accuracy	precision	Recall	F1-score
Individual 4	25410 * 4 = 101640	MB-HMM-VR	86.46%	86.22%	85.38%	86.34%
		MB-HMM-ML	86.07%	85.56%	84.29%	85.81%
		GMM-HMM	84.62%	86.09%	84.62%	85.25%

Fig. 6 Model performance evaluation

the best results with 91.64, 90.79, and 89.02 percent of F1-score for MB-HMM-VR, MB-HMM-ML, and GMM-HMM, respectively. This output could be a result of its size (approximately doubled compared to the other three datasets).

6 Conclusion

In this work, we proposed multivariate Beta-based hidden Markov models as a new extension of the HMMs and applied them to human activity recognition. Our motivation was that the assumption of Gaussianity could not be valid for all types of data. Other distributions such as multivariate Beta distribution have demonstrated considerable flexibility to model data in various real-life applications. Multivariate Beta distribution could be symmetric, asymmetric, or various skewed forms. Here, we assumed that emission probability distributions are raised from multivariate Beta mixture models. We believe that such modifications in the structure of GMM-based HMM, which has been typically used, may carry some robustness. To find the parameter of our MB-based HMM model, we applied two different learning methods, maximum likelihood, and variational inference approach. Then, we tested our model on four datasets related to human activity recognition. Our main goal was studying and detecting the activities of an individual based on analyzing ambient sensor-based data. The results of our test indicate that our proposed model outperforms the conventionally used model such as GMM-based HMM. In future work, we can study the activities of several individuals and test other distributions as emission probabilities.

Acknowledgments The completion of this research was made possible thanks to Ericsson—Global Artificial Intelligence Accelerator in Montreal, Mitacs and the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. C. Dhiman, D.K. Vishwakarma, A review of state-of-the-art techniques for abnormal human activity recognition. Eng. Appl. Artif. Intell. **77**, 21–45 (2019)
2. A. Keshavarzian, S. Sharifian, S. Seyedin, Modified deep residual network architecture deployed on serverless framework of IoT platform based on human activity recognition application. Fut. Gener. Comput. Syst. **101**, 14–28 (2019)
3. M.A. Gul, M.H. Yousaf, S. Nawaz, Z.U. Rehman, H. Kim, Patient monitoring by abnormal human activity recognition based on CNN architecture. Electronics **9**(12), 1993 (2020)
4. X. Zhou, W. Liang, I. Kevin, K. Wang, H. Wang, L.T. Yang, Q. Jin, Deep-learning-enhanced human activity recognition for internet of healthcare things. IEEE Internet Things J. **7**(7), 6429–6438 (2020)
5. Y. Wang, S. Cang, H. Yu, A survey on wearable sensor modality centred human activity recognition in health care. Exp. Syst. Appl. **137**, 167–190 (2019)

6. A. Subasi, M. Radhwan, R. Kurdi, K. Khateeb, IoT based mobile healthcare system for human activity recognition, in *2018 15th Learning and Technology Conference (L&T)* (IEEE, New York, 2018), pp. 29–34
7. S.A. Khowaja, B.N. Yahya, S.-L. Lee, CAPHAR: context-aware personalized human activity recognition using associative learning in smart environments. Human-Centric Comput. Inf. Sci. **10**(1), 1–35 (2020)
8. D. Triboan, L. Chen, F. Chen, Fuzzy-based fine-grained human activity recognition within smart environments, in *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)* (IEEE, New York, 2019), pp. 94–101
9. O.M. Igwe, Y. Wang, G.C. Giakos, J. Fu, Human activity recognition in smart environments employing margin setting algorithm. J. Amb. Intell. Human. Comput. 1–13 (2020). https://doi.org/10.1007/s12652-020-02229-y
10. H. Alemdar, C. Ersoy, Multi-resident activity tracking and recognition in smart environments. J. Amb. Intell. Human. Comput. **8**(4), 513–529 (2017)
11. A. Ghosh, A. Chakraborty, J. Kumbhakar, M. Saha, S. Saha, Humansense: a framework for collective human activity identification using heterogeneous sensor grid in multi-inhabitant smart environments. Pers. Ubiquit. Comput. 1–20 (2020). https://doi.org/10.1007/s00779-020-01432-0
12. A.G. Salguero, M. Espinilla, Ontology-based feature generation to improve accuracy of activity recognition in smart environments. Comput. Electr. Eng. **68**, 1–13 (2018)
13. N. Irvine, C. Nugent, S. Zhang, H. Wang, W.W. Ng, Neural network ensembles for sensor-based human activity recognition within smart environments. Sensors **20**(1), 216 (2020)
14. J. Suto, S. Oniga, Efficiency investigation from shallow to deep neural network techniques in human activity recognition. Cognit. Syst. Res. **54**, 37–49 (2019)
15. N. Zehra, S.H. Azeem, M. Farhan, Human activity recognition through ensemble learning of multiple convolutional neural networks, in *2021 55th Annual Conference on Information Sciences and Systems (CISS)* (IEEE, New York, 2021), pp. 1–5
16. B. Fu, N. Damer, F. Kirchbuchner, A. Kuijper, Sensing technology for human activity recognition: A comprehensive survey. IEEE Access **8**, 83791–83820 (2020)
17. A. Gupta, K. Gupta, K. Gupta, K. Gupta, A survey on human activity recognition and classification, in *2020 International Conference on Communication and Signal Processing (ICCSP)* (IEEE, New York, 2020), pp. 0915–0919
18. D. Goel, R. Pradhan, A comparative study of various human activity recognition approaches, in *IOP Conference Series: Materials Science and Engineering*, volume 1131 (IOP Publishing, Bristol, 2021), pp. 012004
19. K. Banjarey, S.P. Sahu, D.K. Dewangan, A survey on human activity recognition using sensors and deep learning methods, in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)* (IEEE, New York, 2021), pp. 1610–1617
20. A. Jalal, M.A.K. Quaid, A.S. Hasan, Wearable sensor-based human behavior understanding and recognition in daily life for smart environments, in *2018 International Conference on Frontiers of Information Technology (FIT)* (IEEE, New York, 2018), pp. 105–110
21. F. Samie, L. Bauer, J. Henkel, Hierarchical classification for constrained IoT devices: a case study on human activity recognition. IEEE Internet Things J. **7**(9), 8287–8295 (2020)
22. L.M. Dang, K. Min, H. Wang, M.J. Piran, C.H. Lee, H. Moon, Sensor-based and vision-based human activity recognition: a comprehensive survey. Patt. Recogn. **108**, 107561 (2020)
23. H.-B. Zhang, Y.-X. Zhang, B. Zhong, Q. Lei, L. Yang, J.-X. Du, D.-S. Chen, A comprehensive survey of vision-based human action recognition methods. Sensors **19**(5), 1005 (2019)
24. M.A.R. Ahad, Vision and sensor-based human activity recognition: challenges ahead, in *Advancements in Instrumentation and Control in Applied System Applications* (IGI Global, Pennsylvania, 2020), pp. 17–35
25. K. Kim, A. Jalal, M. Mahmood, Vision-based human activity recognition system using depth silhouettes: a smart home system for monitoring the residents. J. Electr. Eng. Technol. **14**(6), 2567–2573 (2019)

26. A.D. Antar, M. Ahmed, M.A.R. Ahad, Challenges in sensor-based human activity recognition and a comparative analysis of benchmark datasets: a review, in *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)* (IEEE, New York, 2019), pp. 134–139
27. A. Prati, C. Shan, K.I.-K. Wang, Sensors, vision and networks: from video surveillance to activity recognition and health monitoring. J. Amb. Intell. Smart Environ. **11**(1), 5–22 (2019)
28. A. Ghosh, A. Chakraborty, D. Chakraborty, M. Saha, S. Saha, Ultrasense: a non-intrusive approach for human activity identification using heterogeneous ultrasonic sensor grid for smart home environment. J. Amb. Intell. Human. Comput. 1–22 (2019). https://doi.org/10.1007/s12652-019-01260-y
29. S. Kalimuthu, T. Perumal, R. Yaakob, E. Marlisah, L. Babangida, Human activity recognition based on smart home environment and their applications, challenges, in *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)* (IEEE, New York, 2021), pp. 815–819
30. A.R. Sanabria, T.W. Kelsey, S. Dobson, J. Ye, Representation learning for minority and subtle activities in a smart home environment. J. Amb. Intell. Smart Environ. **11**(6), 495–513 (2019)
31. S.F. Tahir, L.G. Fahad, K. Kifayat, Key feature identification for recognition of activities performed by a smart-home resident. J. Amb. Intell. Human. Comput. **11**(5), 2105–2115 (2020)
32. F. Cicirelli, G. Fortino, A. Giordano, A. Guerrieri, G. Spezzano, A. Vinci, On the design of smart homes: a framework for activity recognition in home environment. J. Med. Syst. **40**(9), 1–17 (2016)
33. M. Gil-Martín, R. San-Segundo, F. Fernández-Martínez, R. de Córdoba, Human activity recognition adapted to the type of movement. Comput. Electr. Eng. **88**, 106822 (2020)
34. W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas, et al., Towards environment independent device free human activity recognition, in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking* (2018), pp. 289–304
35. S.M. Kwon, S. Yang, J. Liu, X. Yang, W. Saleh, S. Patel, C. Mathews, Y. Chen, Hands-free human activity recognition using millimeter-wave sensors, in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)* (IEEE, New York, 2019), pp. 1–2
36. H. Zou, Y. Zhou, R. Arghandeh, C.J. Spanos, Multiple kernel semi-representation learning with its application to device-free human activity recognition. IEEE Internet Things J. **6**(5), 7670–7680 (2019)
37. H. Xue, W. Jiang, C. Miao, F. Ma, S. Wang, Y. Yuan, S. Yao, A. Zhang, L. Su, DeepMV: multi-view deep learning for device-free human activity recognition. Proc. ACM Interact. Mob. Wear. Ubiquit. Technol. **4**(1), 1–26 (2020)
38. H. Yuan, X. Yang, A. He, Z. Li, Z. Zhang, Z. Tian, Features extraction and analysis for device-free human activity recognition based on channel statement information in b5g wireless communications. EURASIP J. Wirel. Commun. Network. **2020**(1), 1–10 (2020)
39. H. Raeis, M. Kazemi, S. Shirmohammadi, Human activity recognition with device-free sensors for well-being assessment in smart homes. IEEE Instrument. Meas. Mag. **24**(6), 46–57 (2021)
40. S.R. Ramamurthy, N. Roy, Recent trends in machine learning for human activity recognition a survey. Wiley Interdiscipl. Rev. Data Min. Knowl. Discov. **8**(4), e1254 (2018)
41. J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, Deep learning for sensor-based activity recognition: a survey. Patt. Recogn. Lett. **119**, 3–11 (2019)
42. K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, Y. Liu, Deep learning for sensor-based human activity recognition: overview, challenges, and opportunities. ACM Comput. Surv. (CSUR) **54**(4), 1–40 (2021)
43. H.F. Nweke, Y.W. Teh, M.A. Al-Garadi, U.R. Alo, Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: state of the art and research challenges. Expert Syst. Appl. **105**, 233–261 (2018)
44. B. Mor, S. Garhwal, A. Kumar, A systematic review of hidden Markov models and their applications. Arch. Comput. Methods Eng. **28**(3), 1428–1448 (2021)

45. A. Kouadri, M. Hajji, M.-F. Harkat, K. Abodayeh, M. Mansouri, H. Nounou, M. Nounou, Hidden Markov model based principal component analysis for intelligent fault diagnosis of wind energy converter systems. Renew. Energy **150**, 598–606 (2020)
46. G. Manogaran, V. Vijayakumar, R. Varatharajan, P.M. Kumar, R. Sundarasekar, C.-H. Hsu, Machine learning based big data processing framework for cancer diagnosis using hidden Markov model and gm clustering. Wirel. Pers. Commun. **102**(3), 2099–2116 (2018)
47. M. Maseri, M. Mamat, Malay language speech recognition for preschool children using hidden Markov model (HMM) system training, in *Computational Science and Technology* (Springer, New york, 2019), pp. 205–214
48. J.C. Liu, L. Zhang, X. Chen, J.W. Niu, Facial landmark automatic identification from three dimensional (3d) data by using hidden Markov model (HMM). Int. J. Ind. Ergonom. **57**, 10–22 (2017)
49. N. Nguyen, Hidden Markov model for stock trading. Int. J. Finan. Stud. **6**(2), 36 (2018)
50. M. Zhang, X. Jiang, Z. Fang, Y. Zeng, K. Xu, High-order hidden Markov model for trend prediction in financial time series. Phys. A: Stat. Mech. Appl. **517**, 1–12 (2019)
51. P. Asghari, E. Soleimani, E. Nazerfard, Online human activity recognition employing hierarchical hidden Markov models. J. Amb. Intell. Human. Comput. **11**(3), 1141–1152 (2020)
52. Y. Sung-Hyun, K. Thapa, M.H. Kabir, L. Hee-Chan, Log-Viterbi algorithm applied on second-order hidden Markov model for human activity recognition. Int. J. Distrib. Sensor Netw. **14**(4), 1550147718772541 (2018)
53. G. Hu, X. Qiu, L. Meng, Human activity recognition based on hidden Markov models using computational RFID, in *2017 4th International Conference on Systems and Informatics (ICSAI)* (IEEE, New York, 2017), pp. 813–818
54. M.Z. Uddin, Human activity recognition using segmented body part and body joint features with hidden Markov models. Multimedia Tools Appl. **76**(11), 13585–13614 (2017)
55. N.F. Monroy, M. Altuve, Joint exploitation of hemodynamic and electrocardiographic signals by hidden Markov models for heartbeat detection, in *Latin American Conference on Biomedical Engineering* (Springer, New York, 2019), pp. 208–217
56. J. Zhao, S. Basole, M. Stamp, Malware classification with GMM-HMM models (2021). Preprint. arXiv:2103.02753
57. F. Zhang, S. Han, H. Gao, T. Wang, A Gaussian mixture based hidden Markov model for motion recognition with 3d vision device. Comput. Electr. Eng. **83**, 106603 (2020)
58. F. Tian, Q. Zhou, C. Yang, Gaussian mixture model-hidden Markov model based nonlinear equalizer for optical fiber transmission. Optics Express **28**(7), 9728–9737 (2020)
59. Y. Li, B. Hu, T. Niu, S. Gao, J. Yan, K. Xie, Z. Ren, GMM-HMM-based medium-and long-term multi-wind farm correlated power output time series generation method. IEEE Access **9**, 90255–90267 (2021)
60. X. Cheng, B. Huang, J. Zong, Device-free human activity recognition based on GMM-HMM using channel state information. *IEEE Access* **9**, 76592–76601 (2021)
61. C.L.P. Lim, W.L. Woo, S.S. Dlay, B. Gao, Heartrate-dependent heartwave biometric identification with thresholding-based GMM–HMM methodology. IEEE Trans. Ind. Inf. **15**(1), 45–53 (2018)
62. R. Nasfi, M. Amayri, N. Bouguila, A novel approach for modeling positive vectors with inverted Dirichlet-based hidden Markov models. Knowledge-Based Syst. **192**, 105335 (2020)
63. E. Epaillard, N. Bouguila, Data-free metrics for Dirichlet and generalized Dirichlet mixture-based HMMS–a practical study. Patt. Recogn. **85**, 207–219 (2019)
64. E. Epaillard, N. Bouguila, Variational Bayesian learning of generalized Dirichlet-based hidden Markov models applied to unusual events detection. IEEE Trans. Neural Netw. Learn. Syst. **30**(4), 1034–1047 (2018)
65. N. Bouguila, E. Epaillard, Data-free metrics for Dirichlet and generalized Dirichlet mixture-based HMMS-a practical study. Patt. Recogn. **85**, 207–219 (2018)
66. N. Manouchehri, N. Bouguila, W. Fan, Nonparametric variational learning of multivariate beta mixture models in medical applications. Int. J. Imag. Syst. Technol. **31**(1), 128–140 (2021)

67. N. Manouchehri, M. Kalra, N. Bouguila, Online variational inference on finite multivariate beta mixture models for medical applications. IET Image Process. **15**, 1869–1882 (2021)
68. D.M. Blei, A. Kucukelbir, J.D. McAuliffe, Variational inference: a review for statisticians. J. Am. Stat. Assoc. **112**(518), 859–877 (2017)
69. J. Tejedor, J. Macias-Guarasa, H.F. Martins, S. Martin-Lopez, M. Gonzalez-Herraez, A Gaussian mixture model-hidden Markov model (GMM-HMM)-based fiber optic surveillance system for pipeline integrity threat detection, in *Optical Fiber Sensors* (Optical Society of America, Washington, 2018), p. WF36
70. L. Huang, H. Huang, Y. Liu, A fault diagnosis approach for rolling bearing based on wavelet packet decomposition and GMM-HMM. Int. J. Acoust. Vib. **24**(2), 199 (2019)
71. J. Tejedor, J. Macias-Guarasa, H.F. Martins, S. Martin-Lopez, M. Gonzalez-Herraez, A contextual GMM-HMM smart fiber optic surveillance system for pipeline integrity threat detection. J. Lightw. Technol. **37**(18), 4514–4522 (2019)
72. P. Peng, Z. He, L. Wang, Automatic classification of microseismic signals based on MFCC and GMM-HMM in underground mines. Shock Vibr. **2019** (2019). Article ID 5803184
73. A. Xu, Key information recognition algorithm for mobile network video propagation based on discrete GMM-HMM, in *IOP Conference Series: Materials Science and Engineering*, vol. 750 (IOP Publishing, Bristol, 2020), pp. 012224
74. J. Li, N. Gattu, S. Ghosh, Fauto: an efficient GMM-HMM FPGA implementation for behavior estimation in autonomous systems, in *2020 International Joint Conference on Neural Networks (IJCNN)* (IEEE, New York, 2020), pp. 1–8
75. I. Olkin, R. Liu, A bivariate beta distribution. Stat. Probab. Lett. **62**(4), 407–412 (2003)
76. I. Olkin, T.A. Trikalinos, Constructions for a bivariate beta distribution. Stat. Probab. Lett. **96**, 54–60 (2015)
77. N. Manouchehri, N. Bouguila, A probabilistic approach based on a finite mixture model of multivariate beta distributions, in *ICEIS (1)* (2019), pp. 373–380
78. N. Manouchehri, H. Nguyen, N. Bouguila, Component splitting-based approach for multivariate beta mixture models learning, in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)* (IEEE, New York, 2019), pp. 1–5
79. N. Manouchehri, N. Bouguila, W. Fan, Batch and online variational learning of hierarchical Dirichlet process mixtures of multivariate beta distributions in medical applications. Patt. Anal. Appl. 1–14 (2021). https://doi.org/10.1007/s10044-021-01023-6
80. L. Rabiner, B. Juang, An introduction to hidden Markov models. IEEE ASSP Mag. **3**(1), 4–16 (1986)
81. S. Ji, B. Krishnapuram, L. Carin, Variational Bayes for continuous hidden Markov models and its application to active learning. IEEE Trans. Patt. Anal. Mach. Intell. **28**(4), 522–532 (2006)
82. D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, et al. Collecting complex activity datasets in highly rich networked sensor environments, in *2010 Seventh International Conference on Networked Sensing Systems (INSS)* (IEEE, New York, 2010), pp. 233–240
83. H. Sagha, S.T. Digumarti, J. del R. Millán, R. Chavarriaga, A. Calatroni, D. Roggen, G. Tröster, Benchmarking classification techniques using the opportunity human activity dataset, in *2011 IEEE International Conference on Systems, Man, and Cybernetics* (IEEE, New York, 2011), pp. 36–40

Multivariate Beta-Based Hierarchical Dirichlet Process Hidden Markov Models in Medical Applications

Narges Manouchehri and Nizar Bouguila

1 Introduction

Hidden Markov model (HMM) is a powerful approach generally applied to model Markov process systems with hidden states. This method is widely used specifically in cases where we would like to capture latent information from observable sequential data. This method has been successfully applied in several domains of science and technology. In this chapter, we will focus on medical application of this strong modeling approach. In medicine, HMM can assist us in monitoring patient's health changes, expressing progressive alterations to patients' situation or treatment process over time. For instance, it could be employed in verification of a disease development, evaluating health condition, inspecting the results, and probability assessment of transitions from a healthy to a disease state. HMM could be effective in prediction and future risk estimation. There are several works devoted to HMMs such as diagnosing Schizophrenia [1], analyzing cardiac function [2–4], eye tracking [5], classification of EEG signals [6], B cell receptor sequence analysis [7], EEG-based sleep stage scoring [8], estimating dynamic functional brain connectivity [9], cancer analysis [10–13], predicting recurrence of cancers [14], genetics [15, 16], speech recognition [17–22], predicting drug response [23], cancer biomarkers detection [24], analyzing chemotherapy outcomes [25], human activity analysis [26–31] such as fall detection and senior activity analysis using motion sensors [32], HIV prediction [33], sentiment analysis [34], medical image processing [35–37], and many other applications [38–46].

N. Manouchehri · N. Bouguila (✉)
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: narges.manouchehri@mail.concordia.ca; nizar.bouguila@concordia.ca

N. Bouguila et al. (eds.), *Hidden Markov Models and Applications*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-99142-5_10

However, in most of the applications, nature of sequential data is recursive. To handle this situation, some extensions to typical HMM such as hierarchical hidden Markov model [47] and hierarchical Dirichlet process hidden Markov model (HDP-HMM) [48–50] have been proposed. In particular, HDP-HMM has considerable flexibility thanks to its nonparametric structure and has been applied in various areas such as speaker diarization, abnormal activity recognition, classifying human actions, motion detection, segmentation, and classification of sequential data [51–53]. This elegant structure is a solution to one of the challenges in HMM that is defining the proper number of states. Also, it lets us learn more complex and multimodal emission distributions in the hierarchical structure of sequences in real-world applications.

Another issue while dealing with HMM is choosing a distribution for emission probabilities. In several works devoted to HMM, Gaussian Mixture Models (GMM) have been commonly used for modeling emission probability distribution [54–59]. However, this assumption could not be generalized, and recent researches indicate that other alternative such as Dirichlet, generalized Dirichlet, and inverted Dirichlet distribution [60–64] could be considered for several types of data. Inspired by these efforts, we were motivated to choose multivariate Beta mixture models (MBMM) that provide considerable flexibility to model symmetric, asymmetric, and skewed data [65, 66]. So, we construct our HDP-HMM model assuming that the emission probabilities follow MBMM. We call our novel HDP-HMM model "multivariate Beta-based hierarchical Dirichlet process hidden Markov models" (MB-HDP-HMM).

To learn our proposed model, a variety of approaches have been investigated. For instance, maximum likelihood approach may result in overfitting and converging toward a local maximum. Another method is fully Bayesian inference that is precise but has a long computational time. To overcome these prohibitive drawbacks, variational Bayesian approaches [67–69] have been proposed and applied to numerous machine learning algorithms. This learning method is faster than fully Bayesian one and more precise compared to the maximum likelihood approach.

Finally, we evaluate our proposed models on a medical application. The main motivation is that our model is unsupervised, which makes it an adequate tool when data labelling is expensive and takes considerable time. Health-related applications are good examples because there are just medical experts who are eligible to label medical data. Moreover, having predictable and explainable results in such a sensitive domain is one of the essential needs. Therefore, decisions-making and inference based on black boxes [70–72] may not be absolutely trustable. Another concerning challenge is our limitation to access a huge amount of data because of the tough confidentiality rules in healthcare. Thus, some platforms, such as deep learning, which provide precise results but need lots of data for learning [73] could not be easily used. Our proposed algorithm could handle datasets of various sizes, and the process is explainable in human terms.

Our contributions in this chapter could be summarized as follows:

1. We propose a modified version of the hierarchical Dirichlet process hidden Markov model in which emission probabilities are raised from multivariate Beta mixture models. This model, which is less costly compared to deep learning, is capable to fit different sizes of datasets and outcomes are explainable.
2. We apply variational inference to learn our proposed algorithm and secure having accurate outcomes within a proper time interval.
3. We measure the performance of our model and compare it with similar alternatives in medical applications.

This chapter is organized as follows: In Sect. 2, we construct our model and describe multivariate Beta-based hidden Markov models and multivariate Beta-based hierarchical Dirichlet process of hidden Markov model. Section 3 is devoted to parameter estimation with variational inference. In Sect. 4, we present the results of evaluating our proposed model in human activity recognition. Finally, we conclude in Sect. 5.

2 Model Specification

To express our proposed model, we start by explaining the basic structure of HMM for a sequence of events or states. Then, we will add the assumption of having multivariate Beta mixture models as emission probabilities. We call this model, multivariate Beta-based hidden Markov model. Then, we discuss the hierarchical Dirichlet process of this modified hidden Markov model, called multivariate Beta-based hierarchical Dirichlet process hidden Markov model.

2.1 Multivariate Beta-Based Hidden Markov Model

Further to the Markovian characteristics of HMM, in the first-order Markov model, the probability of each event t depends just on state $t-1$ that happens immediately before t. In HMM, a system with hidden states emits observable symbols at any specific point of time.

To mathematically formulate HMM, we need the following parameters:

- Transition probability: Indicating the probability of a change in state from t to $t+1$. Sum of all these probabilities given the current state is equal to 1.
- Initial Probability: The initial state that the system starts from it is denoted as π. These probabilities also sum up to 1.
- Emission probability or observation likelihoods: Parameters indicating the probability of a data point being generated from a specific state.

In our work, HMM is expressed by $\lambda = \{B, C, \varphi, \pi\}$ and the following notations:

1. T: Length of the sequence of our interest, M: the number of mixture components in set $L = \{m_, \ldots, m_M\}$, K: the number of the states.
2. A state sequence $\mathcal{S} = \{S_1, \ldots, S_T\}$ drawn from $P(s_t \mid s_{t-1}, \ldots, s_1) = P(s_t \mid s_{t-1})$.
3. Sequential data $\mathcal{X} = \{X_1, \ldots, X_T\}$.
4. Transition probability from state i to i': $A = \{a_{ii'} = P(s_t = i' \mid s_{t-1} = i)\}$.
5. Emission probability of observing j from state i: $B = \{B_{ij} = P(m_t = j \mid s_t = i)\}$ for $j \in [1, M]$.
6. π_j: Initial probability to begin the sequence from the state j.
7. φ is the set of mixture parameters. In this chapter, we apply multivariate Beta mixture model and φ is the shape parameter, $\alpha_{ij} = (\alpha_{1ij}, \ldots, \alpha_{Dij})$, with $i \in [1, K]$ and $j \in [1, M]$.

We can denote the complete likelihood of HMMs as follows:

$$p(\mathbf{X} \mid A, B, \pi, \alpha) = \sum_{S} \sum_{L} \pi_{s_1} \left[\prod_{t=1}^{T-1} a_{s_t, s_{t+1}} \right] \left[\prod_{t=1}^{T} b_{s_t, m_t} p(x_t \mid \alpha_{s_t, m_t}) \right]. \tag{1}$$

Here, we explain the model for one sequence. In case of having more observations, this could be generalized by adding a summation over the whole sequence.

$p(x_t \mid \alpha_{s_t, m_t})$ is the multivariate Beta distribution (MB). To describe it in detail, let us assume to have a D-dimensional observation, $\mathbf{X} = \left(x_1, \ldots, x_D\right)$, where all its elements are greater than zero and less than one.

The probability density function of multivariate Beta distribution [74] is expressed as follows:

$$p(\mathbf{X} \mid \boldsymbol{\alpha}) = \frac{\Gamma(\mid \alpha \mid) \prod_{d=1}^{D} x_d^{\alpha_d - 1}}{\prod_{d=0}^{D} \Gamma(\alpha_d) \prod_{d=1}^{D} (1 - x_d)^{(\alpha_d + 1)}} \left[1 + \sum_{d=1}^{D} \frac{x_d}{(1 - x_d)} \right]^{-|\alpha_j|}. \tag{2}$$

$\boldsymbol{\alpha} = \left(\alpha_0, \ldots, \alpha_D\right)$ is the shape parameter such that $\alpha_d > 0$ for $d = 0, \ldots, D$, $\mid \alpha \mid = \sum_{d=0}^{D} \alpha_d$, and $\Gamma(.)$ represents the Gamma function.

Figures 1 and 2 illustrate some examples of multivariate Beta distributions and multivariate Beta mixture models, respectively. These figures illustrate the flexibility

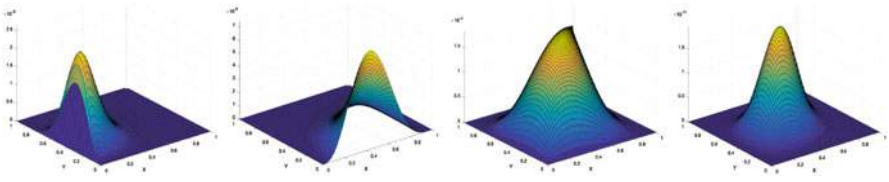


Fig. 1 Multivariate Beta distribution with different shape parameters

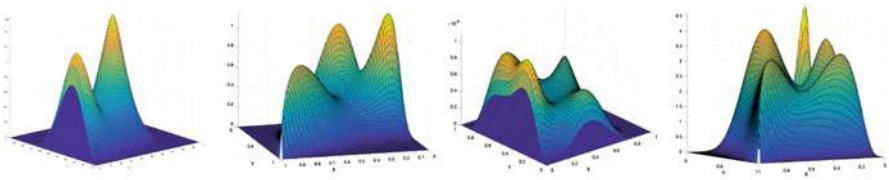


Fig. 2 Multivariate Beta mixture models with 2, 3, 4, and 5 components

of this distribution. So, it has the capability of capturing symmetric and asymmetric shapes of data [65, 66].

Thus, assuming that emission probabilities are raised from MB mixture model, complete log-likelihood of $p(\mathbf{X} \mid B, C, \pi, \alpha)$ could be written as

$$\begin{aligned}\log(p(\mathbf{X}, Z \mid \lambda)) = \log(\pi_{s_1}) + \sum_{t=1}^{T-1} \log(a_{s_t, s_{t+1}}) + \sum_{t=1}^{T} \log(b_{s_t, m_t}) + \\ + \sum_{t=1}^{T} \left[\log\left(\Gamma\left(\sum_{d=0}^{D} \alpha_d \right) \right) - \log\left(\prod_{d=0}^{D} \Gamma(\alpha_d) \right) + \sum_{d=1}^{D} \left((\alpha_d - 1) \log x_d \right) \right. \\ \left. - \sum_{d=1}^{D} \left((\alpha_d + 1) \log(1 - x_d) \right) - \left(\sum_{d=0}^{D} \alpha_d \right) \log\left[1 + \sum_{d=1}^{D} \frac{x_d}{1 - x_d} \right] \right] \end{aligned} \tag{3}$$

2.2 *Multivariate Beta-Based Hierarchical Dirichlet Process of Hidden Markov Model*

To express our hierarchical HMM, we need first to describe Dirichlet process (DP) and stick-breaking construction [75, 76]. The Dirichlet process [77] is an extension of the Dirichlet distribution. It has two inputs, a nonnegative precision scalar, ϵ, and a base distribution G_0. DP is defined over the measurable space $(\Theta, \mathcal{B})$. For a disjoint set of $B = \{B_1, \ldots, B_D\}$ and partition of Θ, the Dirichlet process is defined as follows where $\bigcup_i B_i = \Theta$:

$$\left(G\left(B_1\right), \ldots, G\left(B_D\right)\right) \sim \operatorname{Dir}\left(\epsilon G_0\left(B_1\right), \ldots, \epsilon G_0\left(B_D\right)\right). \tag{4}$$

In terms of dimensionality, DP is infinite ($D \to \infty$). If we draw G from a DP expressed by $G \sim DP(\epsilon G_0)$, we will have

$$G = \sum_{i=1}^{\infty} p_i \delta_{\theta_i}. \tag{5}$$

θ_i indicated the location drawn from G_0 and is related to a measure, p_i.

We can consider θ_i as the emission probability at state i in HMM. To move forward, we need to explain general definition of a stick-breaking process. Let us assume a probability mass function $p = (p_1, \ldots, p_{d+1})$, so we have

$$p_i = V_i \prod_{i'=1}^{i-1} (1 - V_{i'}), \quad p_{d+1} = 1 - \sum_{i=1}^{d} p_i \quad V_i \sim \text{Beta}(v_i, \omega_i). \tag{6}$$

$v_i = (v_1, \ldots, v_d)$ and $\omega_1 = (\omega_d, \ldots, \omega_i)$ are nonnegative, real parameters for $i = 1, \ldots, d$. The value of d could be either finite or infinite, and finite case is similar to a distribution called generalized Dirichlet distribution (GDD) [78, 79]. In infinite case, we may have various ranges of priors by changing v and ω [80]. For HDP-HMM, we construct a draw from DP with the following representation of a stick-breaking process:

$$G = \sum_{i=1}^{\infty} p_i \delta_{\theta_i}, \quad p_i = V_i \prod_{j=1}^{i-1} (1 - V_j), \quad V_i \sim \text{Beta}(1, \gamma), \quad \theta_i \sim G_0. \tag{7}$$

$\gamma = \sum_i \beta_i$ affects a draw from DP. If $\gamma \to 0$, a measure degeneration at a random component with location drawn from G_0 happens. In contrast, if $\gamma \to \infty$, the breaks are very small and G that reaches to convergence to the empirical distribution of the individual draws from G_0, and G_0 is reproduced.

If we focus on a distribution from which we draw the data and show it by $p(x \mid \theta)$ with parameter θ, a DP mixture model is presented by

$$x_i \mid \theta_i \sim p(x \mid \theta_i), \quad \theta_i \mid G \sim G, \quad G \mid \gamma G_0 \sim DP(\gamma G_0). \tag{8}$$

Hidden Markov models could be considered as a special case of mixture models that are dependent on the states. The supports of the mixtures are shared among them with various mixing weights. We represent state-dependent mixture model of HMM as follows where $\theta_i \equiv (b_{i1}, \ldots, b_{iM})$, distribution is MB, and the initial state is selected from π:

$$x_t \mid \theta_{s_t} \sim \mathcal{MB}(\theta_{s_t}), \quad \theta_{s_t} \mid s_{t-1} \sim G_{s_{t-1}}, \quad G_i = \sum_{i'=1}^{D} a_{ii'} \delta_{\theta_{i'}}. \tag{9}$$

If we consider each transition as a DP, it will make a problem specifically if we assume that each row, i, is raised from an infinite transition probability matrix expressed as follows:

$$G_i = \sum_{i'=1}^{\infty} a_{ii'} \delta_{\theta_{ii'}}, \quad a_{ii'} = V_{ii'} \prod_{k=1}^{i'-1} (1 - V_{ik}), \quad V_{ii'} \sim \text{Beta}(1, \gamma), \quad \theta_{ii'} \sim G_0. \tag{10}$$

$a_{ii'}$ presents the i'^{th} component of a_i that is an infinite vector. In case of having a continuous G_0, the probability of transition to a previous state is zero for each $\theta_{ii'}$ because $p(\theta_m = \theta_n) = 0$ for $m \neq n$. Thus, such approaches are not practical to construct Dirichlet process of HMM.

Hierarchical Dirichlet process hidden Markov model is proposed to tackle this issue. In hierarchical Dirichlet process (HDP), the base distribution, G_0, over Θ is itself arised from a DP that relatively assures us that G_0 will be almost discrete. We formulate the process as

$$G_m \sim DP(\beta G_0), \quad G_0 \sim DP(\gamma H). \tag{11}$$

In HDP as a two-level hierarchical structure, the distribution on the data points in Θ is changed from the continuous H to the discrete, but infinite G_0. If we draw for G_m multiple times, the weight on the same set of states will be substantial. This procedure and second level of DP can be expressed as follows with truncation level of K:

$$G_0 = \sum_{i=1}^{K} p_i \delta_{\theta_i}, \quad p_i = V_i \prod_{i'=1}^{i-1} (1 - V_{i'}), \quad V_i \sim \text{Beta}(1, \gamma), \quad \theta_i \sim H \tag{12}$$

$$(G_m(\theta_1), G_m(\theta_2), \ldots, G_m(\theta_K)) \sim \text{Dir}(\beta p_1, \beta p_2, \ldots, \beta p_K).$$

$G(\theta_i)$ indicates a probability measure at location θ_i. To summarize the procedure of two-level hierarchy, we assume to have a DP at top level through which the number of states and their observation parameters are chosen. Then, the mixing weights are considered as prior for second level where the transition probabilities are drawn. As a conjugacy between these two levels does not exist, there is not a truly variational solution [81]. To construct HDP-HMM, we use a prior similar to Eq. (6) that is more general and flexible compared to the stick-breaking process for drawing from the DP, in which we draw simultaneously both of Beta$(1, \alpha)$-distributed random variables and the atoms associated with the resulting weights. As we explained before, Eq. (6) could be considered as a GDD, and its density

function of $\boldsymbol{V} = (V_1, \ldots, V_K)$ is expressed as follows where $v = (v_1, v_2, \ldots)$ and $\omega = (\omega_1, \omega_2, \ldots)$:

$$f(\boldsymbol{V}) = \prod_{i=1}^{K} f(V_i) = \prod_{i=1}^{K} \frac{\Gamma(v_i + \omega_i)}{\Gamma(v_i)\,\Gamma(\omega_i)} V_i^{v_i - 1} (1 - V_i)^{\omega_i - 1}. \tag{13}$$

By changing $\boldsymbol{V}$ to $\boldsymbol{p}$, the density of $\boldsymbol{p}$ is defined by

$$f(\boldsymbol{p}) = \prod_{i=1}^{K} \left(\frac{\Gamma(v_i + \omega_i)}{\Gamma(v_i)\,\Gamma(\omega_i)} p_i^{v_i - 1} \right) p_{K+1}^{\omega_K - 1} (1 - P_1)^{\omega_1 - (v_2 + \omega_2)} \times \cdots \times \tag{14}$$

$$(1 - P_{K-1})^{\omega_{K-1} - (v_{K-1} + \omega_{K-1})}.$$

Mean and variance for each element, p_i, are

$$\mathbb{E}[p_i] = \frac{v_{i'} \prod_{\ell=1}^{i'-1} \omega_\ell}{\prod_{\ell=1}^{i'} (v_\ell + \omega_\ell)}, \quad \mathbb{V}[p_i] = \frac{v_{i'}(v_{i'} + 1) \prod_{\ell=1}^{i'-1} \omega_\ell (\omega_\ell + 1)}{\prod_{\ell=1}^{i'} (v_\ell + \omega_\ell)(v_\ell + \omega_\ell + 1)}. \tag{15}$$

GDD is a special case of typical standard Dirichlet distribution. In GDD case, the construction of $\boldsymbol{p}$ from the infinite process of Eq. (14) is referred by $\boldsymbol{p} \sim GDD(\boldsymbol{v}, \boldsymbol{\omega})$. For a set of N observations that are independent identically distributed (iid) , $X_n \overset{iid}{\sim} \text{Mult}(\boldsymbol{p})$, the posterior of the respective priors presented by $\boldsymbol{v}'$ and ω' is parametrized as follows:

$$v_i' = v_i + \sum_{n=1}^{N} \mathbf{1}(X_n = i), \quad \omega_i' = \sum_{j>i} \sum_{n=1}^{N} \mathbf{1}(X_n = j). \tag{16}$$

$\mathbf{1}(\cdot)$ is an indicator function that will be equal to one if the argument is true and zero, otherwise. This is applied to count the number of times the random variables are equal to values of interest.

3 Variational Learning

To estimate model's parameters, we adopt variational inference. In this method, we introduce an approximating distribution $q(A, B, \pi, \alpha, S, L)$ for the true posterior $p(A, B, \pi, \alpha, S, L \mid \mathbf{X})$. Then, we try to minimize the distance between these two distributions with the help of Kullback–Leibler distance. As marginal distribution is not tractable, we try to find a tractable lower bound in it. Based on Jensen's inequality, as $KL(q \mid\mid p) \geq 0$, $KL(q \mid\mid p) = 0$ when q is equal to true posterior.

$\mathcal{L}(q)$ as a lower bound to $\ln p(\mathbf{X})$ could be found by

$$\ln(p(\mathbf{X})) = \mathcal{L}(q) - \mathrm{KL}(q(A, B, \pi, \alpha, S, L) \| p(A, B, \pi, \alpha, S, L \mid \mathbf{X})). \tag{17}$$

The true posterior distribution is practically intractable and cannot be directly applied in variational inference. Borrowing the idea from *mean field theory*, we consider a restricted family of distributions q and adopt a factorization approach [82, 83]. So, we have

$$q(A, B, \pi, \alpha, S, L) = q(A)q(B)q(\pi)q(\alpha)q(S, L). \tag{18}$$

With the help of iterative expectation maximization (EM), we perform this approximation. Expectation step is as follows [84] such that m_i is the expected number of data points from a component in an iteration with truncation to K dimensions:

$$\langle \ln V_i \rangle = \psi\left(1 + \langle x_i \rangle\right) - \psi\left(1 + \gamma_i + \sum_{i'=i}^{K} \langle x_{i'} \rangle\right) \tag{19}$$

$$\langle \ln\left(1 - V_i\right) \rangle = \psi\left(\gamma_i + \sum_{i'=i+1}^{K} \langle x_{i'} \rangle\right) - \psi\left(1 + \gamma_i + \sum_{i'=i}^{K} \langle x_{i'} \rangle\right) \tag{20}$$

$$\langle \ln p_1 \rangle = \langle \ln V_i \rangle \tag{21}$$

$$\langle \ln p_k \rangle = \langle \ln V_k \rangle + \sum_{i'=1}^{k-1} \langle \ln\left(1 - V_{i'}\right) \rangle \quad 2 \leq k < K \tag{22}$$

$$\langle \ln p_K \rangle = \sum_{i'=1}^{K-1} \langle \ln\left(1 - V_{i'}\right) \rangle\,. \tag{23}$$

ψ represents the digamma function. Then, we optimize the following quantity:

$$\ln(p^{\star}(X_t \mid \alpha_{s_t,m_t})) = \phi_{ijt}^{B} \int q(\alpha) \ln(p(X_t \mid \alpha_{s_t,m_t}))d\alpha \tag{24}$$

$$= \phi_{ijt}^{B} \int q(\alpha) \ln\left(\frac{\Gamma\left(\sum_{d=1}^{D} \alpha_{ijl}\right)}{\prod_{d=1}^{D} \Gamma(\alpha_{ijl})}\right) d\alpha + \phi_{ijt}^{B} \int q(\alpha) \Bigg[\sum_{d=1}^{D} \Big((\alpha_d - 1) \log x_{td}\Big)$$

$$- \sum_{d=1}^{D} \Big((\alpha_d + 1) \log(1 - x_{td})\Big) - \Big(\sum_{d=0}^{D} \alpha_d\Big) \log\left[1 + \sum_{d=1}^{D} \frac{x_{td}}{1 - x_{td}}\right]\Bigg] d\alpha,$$

where $\phi_{ijt}^{B} \triangleq q(s_{t-1} = i, m_t = j)$ and $*$ indicates an optimized parameter. $p(X_t \mid \alpha_{s_t,m_t})$ is the MB distribution. We presented in detail the variational inference of multivariate Beta mixture models in our previous works [66, 85], and similar to them, we have

$$\bar{\alpha}_{ijl} = \frac{u_{ijl}}{v_{ijl}} \tag{25}$$

$$\begin{aligned}\left\langle \frac{\Gamma(\sum_{d=1}^{D}\alpha_{ijl})}{\prod_{d=1}^{D}\Gamma(\alpha_{ijl})} \right\rangle &= \ln \frac{\Gamma\left(\sum_{l=1}^{D}\overline{\alpha}_{jl}\right)}{\prod_{l=1}^{D}\Gamma\left(\overline{\alpha}_{jl}\right)} + \sum_{l=1}^{D}\overline{\alpha}_{jl}\left[\psi\left(\sum_{l=1}^{D}\overline{\alpha}_{jl}\right) - \psi\left(\overline{\alpha}_{jl}\right)\right] \\ &\times \left[\left\langle \ln \alpha_{jl}\right\rangle - \ln \overline{\alpha}_{jl}\right] + \frac{1}{2}\sum_{l=1}^{D}\overline{\alpha}_{jl}^{2}\left[\psi'\left(\sum_{l=1}^{D}\overline{\alpha}_{jl}\right) - \psi'\left(\overline{\alpha}_{jl}\right)\right] \\ &\times \left\langle \left(\ln \alpha_{jl} - \ln \overline{\alpha}_{jl}\right)^2\right\rangle + \frac{1}{2}\sum_{a=1}^{D}\sum_{b=1,a\neq b}^{D}\overline{\alpha}_{ja}\,\overline{\alpha}_{jb}\left[\psi'\left(\sum_{l=1}^{D}\overline{\alpha}_{jl}\right)\right. \\ &\left.\times \left(\left\langle \ln \overline{\alpha}_{ja}\right\rangle - \ln \overline{\alpha}_{ja}\right) \times \left(\left\langle \ln \overline{\alpha}_{jb}\right\rangle - \ln \overline{\alpha}_{jb}\right)\right]\end{aligned} \tag{26}$$

$$\langle \ln(\alpha_{ijd})\rangle = \Psi(u_{ijd}) - \ln(v_{ijd}). \tag{27}$$

In maximization step, we update variational factors as follows:

$$q(A) = \prod_{i=1}^{K} GDD(\boldsymbol{v'}_i, \boldsymbol{\omega'}_i) \tag{28}$$

$$q(\alpha) = \prod_{i=1}^{K}\prod_{j=1}^{M} q(\alpha_{ij}), \quad q(\alpha_{ij}) = \prod_{d=1}^{D}\mathcal{G}(\alpha_{ijl} \mid u_{ijl}^{\star}, v_{ijl}^{\star}) \tag{29}$$

$$q(\pi) = \mathcal{D}(\boldsymbol{v'}_{\pi}, \boldsymbol{\omega'}_{\pi}). \tag{30}$$

Considering [84], we have

$$q(\boldsymbol{\gamma}) = \prod_{i=1}^{K}\prod_{i'=1}^{K-1}\mathcal{G}\left(c+1, d - \langle \ln(1 - V_{ii'})\rangle\right) \tag{31}$$

$$q\left(\gamma_{\pi}\right) = \prod_{i=1}^{K-1}\mathcal{G}\left(\tau_{\pi 1}+1, \tau_{\pi 2} - \langle \ln(1 - V_{\pi i})\rangle\right). \tag{32}$$

$$u^{\star}_{ijl} = u_{ijl} + \mathcal{U}_{ijl}, \quad v^{\star}_{ijl} = v_{ijl} - \mathcal{V}_{ijl} \tag{33}$$

$$\mathcal{U}_{ijl} = \sum_{p=1}^{P} \langle Z_{pij} \rangle \bar{\alpha}_{ijl} \Bigg[\Psi\Bigg(\sum_{d=1}^{D} \bar{\alpha}_{ijd} \Bigg) - \Psi(\bar{\alpha}_{ijl}) \tag{34}$$

$$+ \sum_{d=1, d\neq l}^{D} \Psi'\Bigg(\sum_{d=1}^{D} \bar{\alpha}_{ijd} \Bigg) \bar{\alpha}_{ijd} (\langle \ln(\alpha_{ijd}) \rangle - \ln(\bar{\alpha}_{ijd})) \Bigg]$$

$$\mathcal{V}_{ijl} = \sum_{p=1}^{P} \langle Z_{pjd} \rangle \Bigg[\ln x_{pl} - \ln(1 - x_{pl}) - \ln\Bigg[1 + \sum_{d=1}^{D} \frac{x_d}{(1 - x_{pl})} \Bigg] \Bigg]. \tag{35}$$

$\psi(.)$ and $\psi'(.)$ in the above equations represent the digamma and trigamma functions. The value of $Z_{pij} = 1$ if X_{pt} belongs to state i and mixture component j, and zero, otherwise. Thus, $\langle Z_{pij} \rangle = \sum_{t=1}^{T} \phi^{C}_{pijt} = p(s = i, m = j \mid X)$, and we compute responsibilities through a simple forward–backward procedure [86].

$$\pi^{\star}_{i} \triangleq \exp[\langle \ln(\pi_i) \rangle_{q(\pi)}]. \tag{36}$$

4 Experimental Results

We tested our algorithm in human activity recognition (HAR). Providing information and discovering knowledge about individuals' physical activities is one of the most attractive and important topics in numerous fields of science and technology. Human activity recognition using various types of devices and sensor networks is broadly used in a vast range of applications such as health, athletics, and senior monitoring, rehabilitation, improving well-being, discovering patterns, and detecting activities for security. Several scientists have focused on this complex subject; however, there are lots of aspects to be addressed. In this application, data are collected by wearable, object, and ambient sensors. For instance, in medicine, caregivers could monitor and recognize the activities of patients who are suffering from morbid obesity, diabetes, dementia, or other mental disorders. This helps the healthcare system by preventing undesirable consequences based on predicting abnormal activities. Due to the sensitivity of domains in which HAR could be used, we tested our algorithm on this application as there are still issues for investigation in realistic conditions. We chose a real dataset, called opportunity [87, 88], in which information was collected with three types of sensors including external and wearable sensors. Figures 3 and 4 show the setup and some types of sensors. Some of these sensors were fixed in points of interest, and the others were attached to volunteer users.

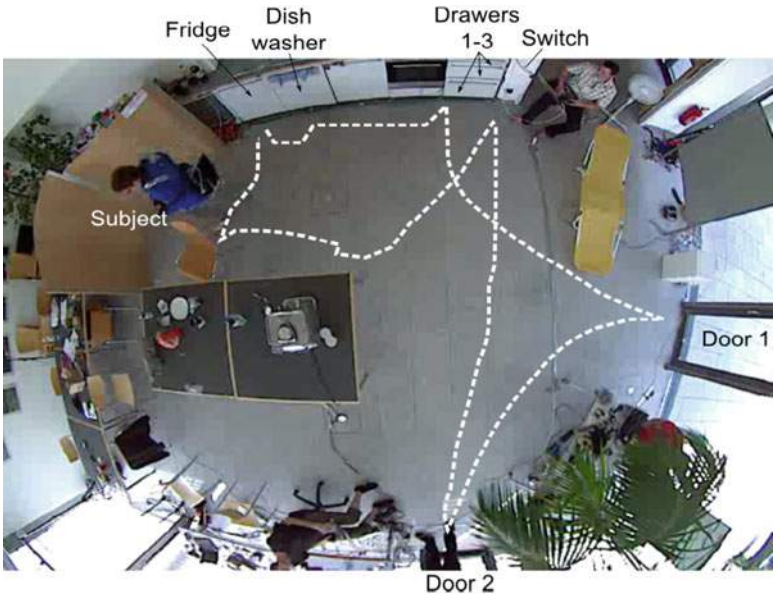


Fig. 3 Platform and sensor setup

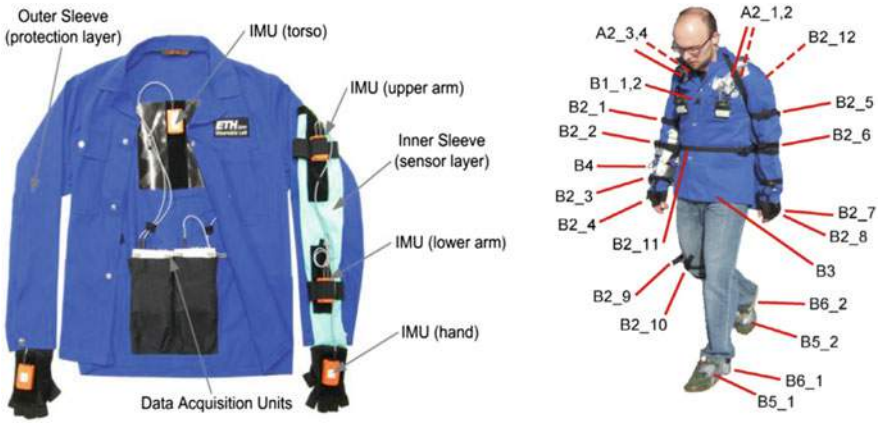


Fig. 4 Wearable sensors

This system was able to recognize activities of different levels as shown in Fig. 5. The detailed information about sensors is as follows:

– Body-worn sensors: 7 inertial measurement units (IMUs), 12 3D acceleration sensors, 4 3D coordinates from a localization system.
– Object sensors: 12 objects are instrumented with wireless sensors measuring 3D acceleration and 2D rate of turn.

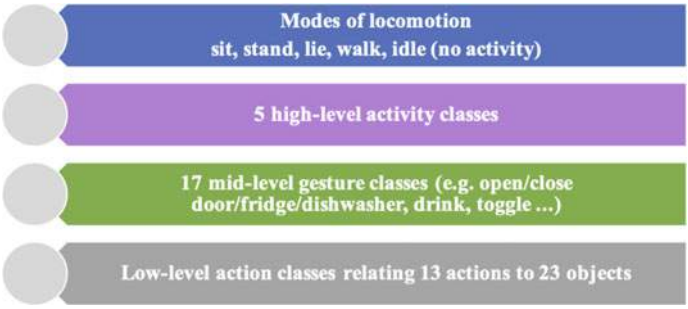


Fig. 5 Different levels of activities

– Ambient sensors: 13 switches and 8 3D acceleration sensors in kitchen appliances and furniture.

The experiment is based on data collected from 4 users and 6 runs per users including 5 Activity of Daily Living (ADL) and one "drill" run. ADL is associated with a very natural manner of daily activities, and in a drill, individuals execute a scripted sequence of activities.

We consider here the mode of collection for four actions: standing, walking, lying, and sitting. Also, we focused on the first individual and her/his 2 runs of activities (first and third) and tested our algorithms on them.

4.1 First Individual, First Run of Activities

This dataset includes 4 activities of the first individual and has 108 features. By analyzing data, we faced some challenges while testing our proposed algorithm on this dataset. We summarize the issues and solutions as follows:

1. Oversampling to handle unbalanced data: As it is shown in Fig. 6, the number of instances in each cluster is very different, and standing, walking, lying, and sitting have 59.7%, 17.4%, 19.9%, and 3% of share, respectively. It is worth noting that such inequality in the distribution of observations per class causes a frequency bias, and our model may place more emphasis on learning from instances with more common occurrence. We tackled this issue with the help of Synthetic Minority Oversampling Technique (SMOTE). In this approach, we generate new data points by interpolating between instances in the original dataset. So, we achieved having a balanced dataset with 22,380 instances in each cluster as shown in Fig. 7.
2. Feature scaling via normalization to handle various ranges of features:
   The second issue that we faced was a broad range of features in the dataset. We plotted some of the features in Fig. 8 to support our idea through visualization. These box plots indicate that the minimum and maximum values, as well as

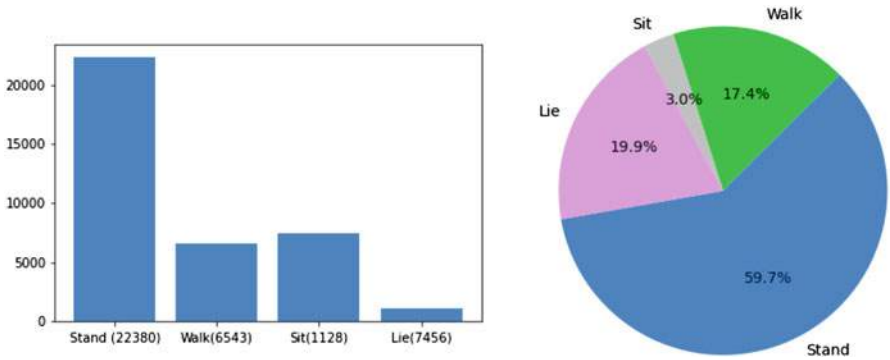


Fig. 6 Bar and pie chart of HAR dataset 1

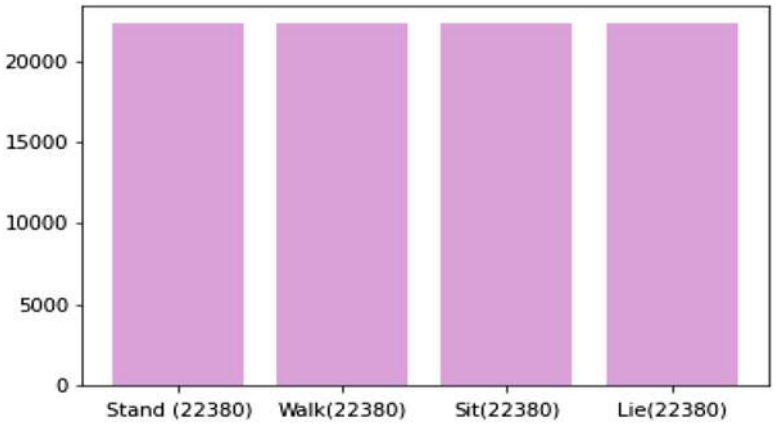


Fig. 7 Oversampling results with SMOTE

distribution of features, are so diverse. Also, Fig. 9 illustrates some examples of feature distribution vs. activity labels. The solution to tackle this problem is normalization or Min-Max scaling. This technique shifts and re-scale values in such a way that their ranges will end up between 0 and 1. To do this, we use the following formula:

$$\mathbf{X} = \frac{\mathbf{X} - \mathbf{X}_{min}}{\mathbf{X}_{max} - \mathbf{X}_{min}}. \tag{37}$$

3. Replacing missing values with the median of each feature: Similar to lots of cases while dealing with real-world applications, this dataset includes missing values. Table 1 indicates the number of missing values and their associated columns. As it was shown in Fig. 8, some features have outliers. Thus, our strategy in missing value imputation and minimizing the effect of outliers is replacing them with the median of each feature.

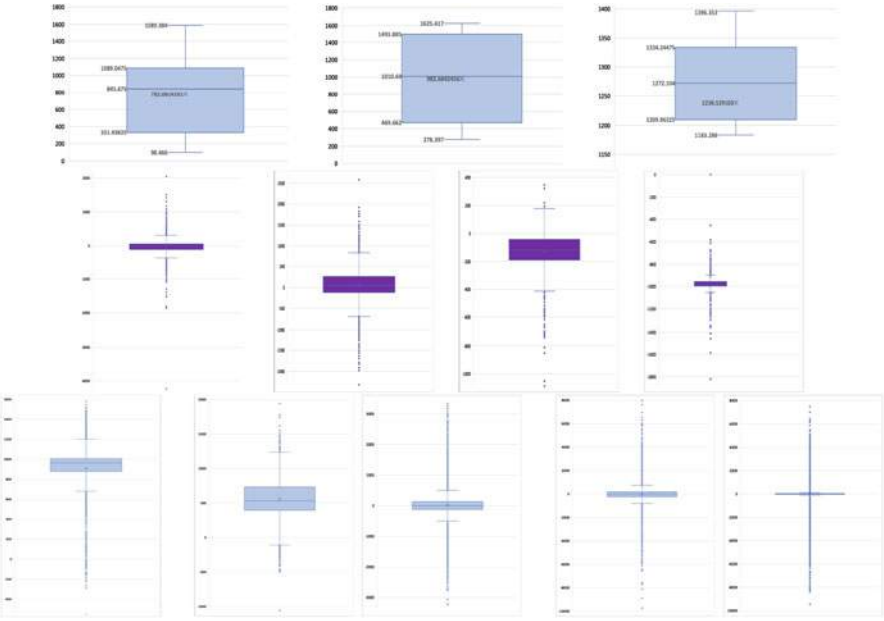


Fig. 8 Examples of different ranges of features

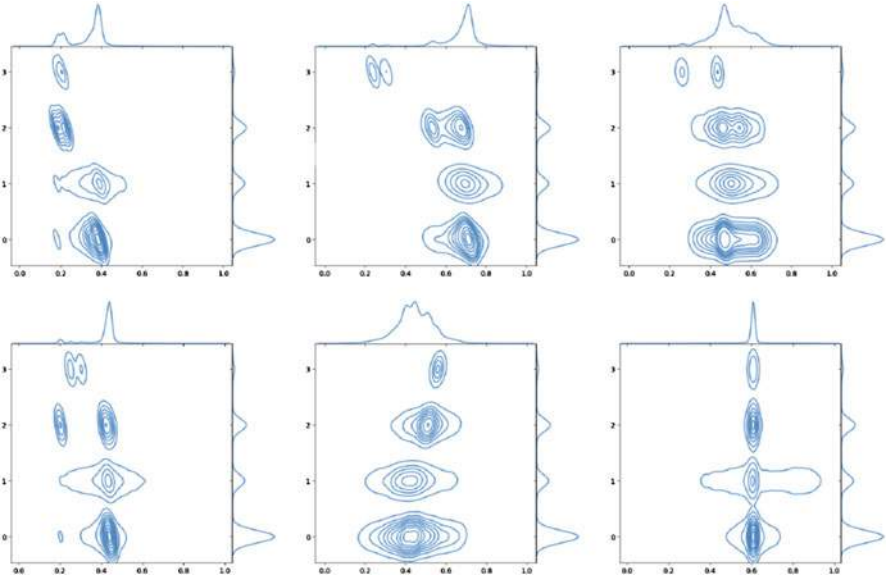


Fig. 9 Feature distribution vs. labels

Table 1 The number of Nan in each column

Column	Numbers of Nan in each column
1, 2, 3	454
4, 5, 6, 10, 11, 12 , 28, 29, 30	20
13, 14, 15	92
19, 20, 21	1681
22, 23, 24	311
34, 35, 36	37,507

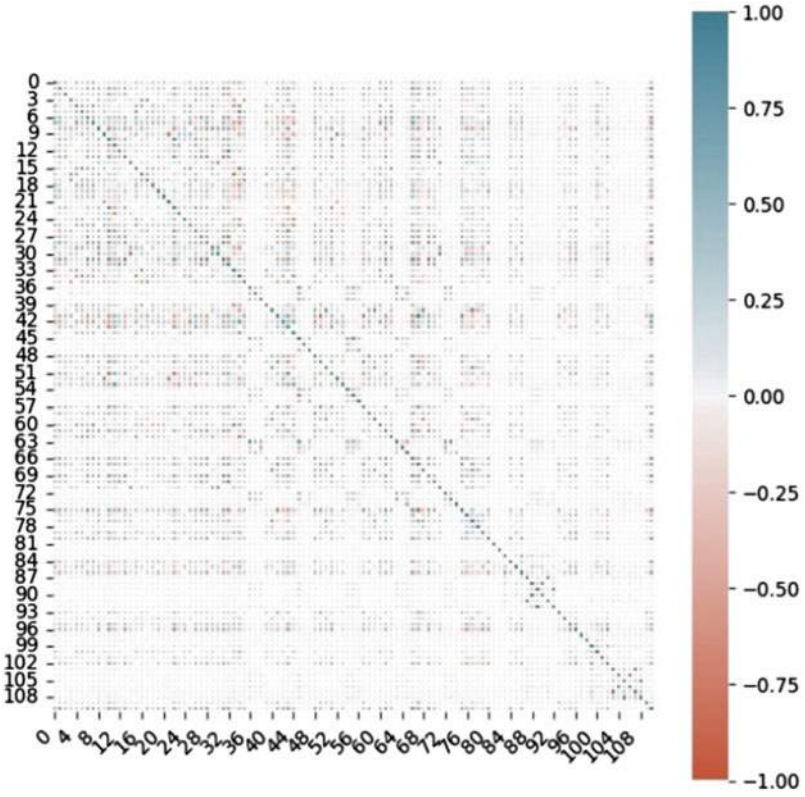


Fig. 10 Correlation matrix

4. Dimensionality reduction: This dataset has 108 attributes. Figure 10 illustrates correlation matrix of its features. As a part of preprocessing step, we reduced the number of attributes while saving as much of the variation in the dataset as possible. This helps us to prevent some issues such as reducing computational time, increasing the overall model performance, avoiding the curse of dimensionality, reducing the chance of overfitting, decreasing the probability of multicollinearity and high correlation among features, and removing noise by keeping just the

Table 2 Model performance evaluation results

Method	Accuracy	Precision	Recall	F1score
MB-HDP-HMM-VR	88.33	88.34	88.33	88.34
MB-HMM-VR	86.72	86.75	86.76	86.76
GMM-HMM	85.67	85.75	85.71	85.72

most important attributes. In our experiments, we applied Principal Component Analysis (PCA) to reduce dimensions.

After solving the above-mentioned issues, we tested our algorithm on this dataset. To assess the model performance, we used the four following criteria:

$$\begin{aligned} Accuracy &= \frac{TP + TN}{\text{Total number of observations}} \\ Precision &= \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \\ F1score &= \frac{2 \times precision \times recall}{precision + recall}. \end{aligned} \tag{38}$$

TP, TN, FP, and FN represent the total number of true positives, true negatives, false positives, and false negatives, respectively. Table 2 illustrates the evaluation results and comparing our proposed model with similar alternatives. As it is shown, MB-HDP-HMM-VR outperforms other models by 88.33%, 88.34%, 88.33 %, 88.34% of accuracy, precision, recall, and F1 score, respectively.

4.2 First Individual, Second Run of Activities

This dataset has 25,305 observations, including 10379, 6029, 7603, 1294 instances for standing, walking, sitting, lying, respectively. As illustrated in Fig. 11, we have the same issue of unbalancing that we had in the previous dataset. We solve this problem with SMOTE and get a balanced dataset as shown in Fig. 12.

Moreover, we need normalization as the ranges of attributes are broadly different. Figures 13 and 14 demonstrate characteristics of some of features.

The next challenge is replacing missing values. In Fig. 15, we show the number of missing values for the attributes. We take the same strategy as the previous case and replace them with the median of attributes.

To make sure that having high dimensionality will not affect model performance and to avoid potential issues that we discussed previously, we use PCA to reduce features. The correlation matrix of dataset is demonstrated in Fig. 16.

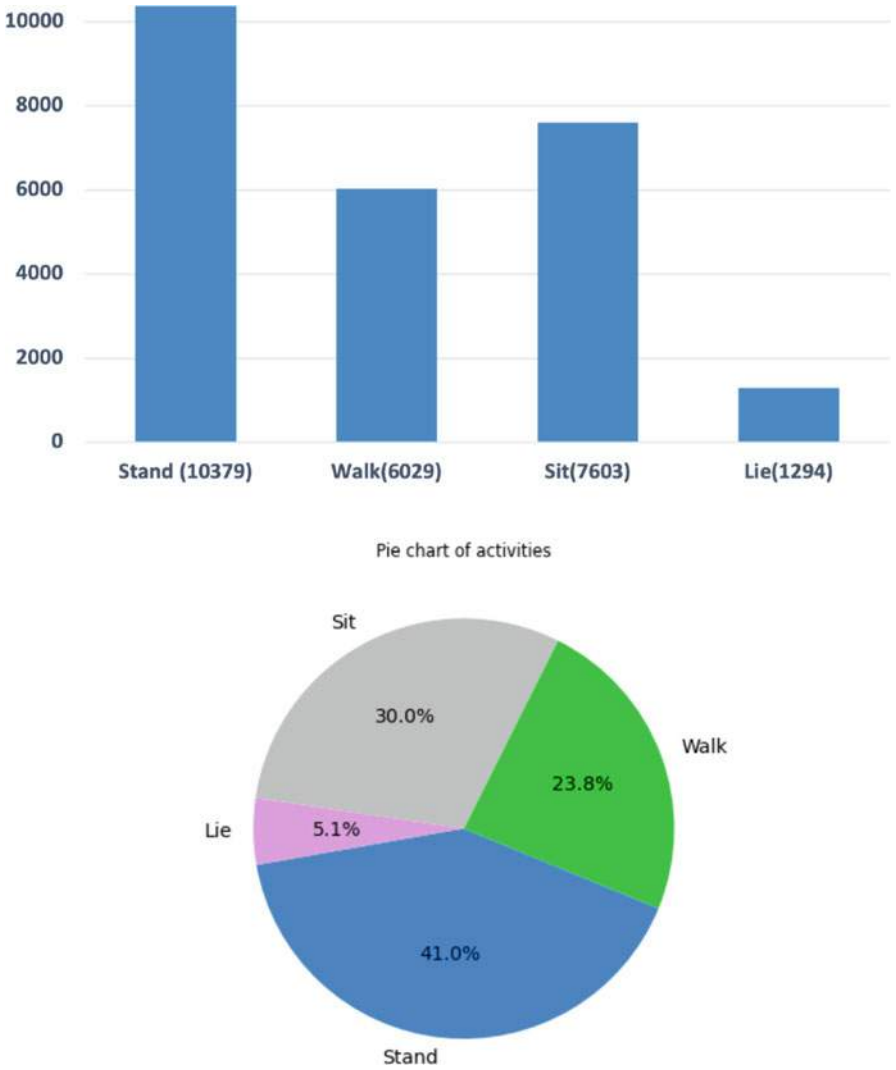


Fig. 11 Bar and pie chart of HAR dataset 2

Fig. 12 Oversampling results with SMOTE

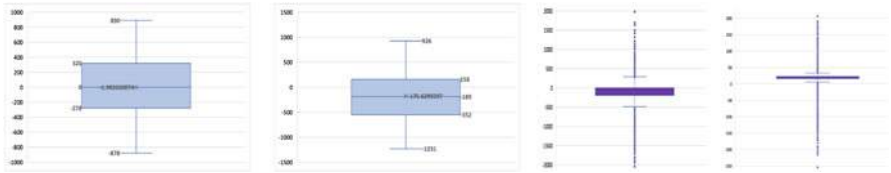


Fig. 13 Examples of different ranges of features

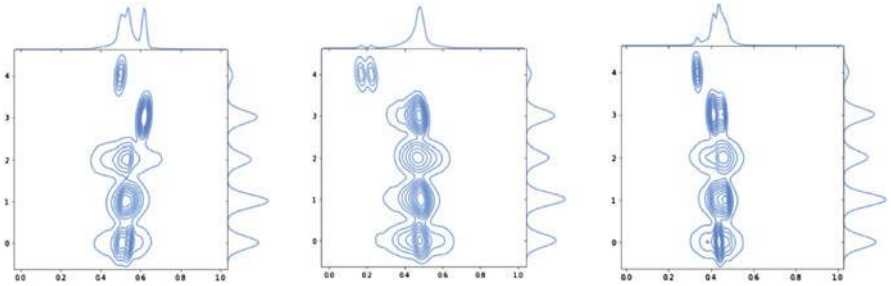


Fig. 14 Feature distribution vs. labels

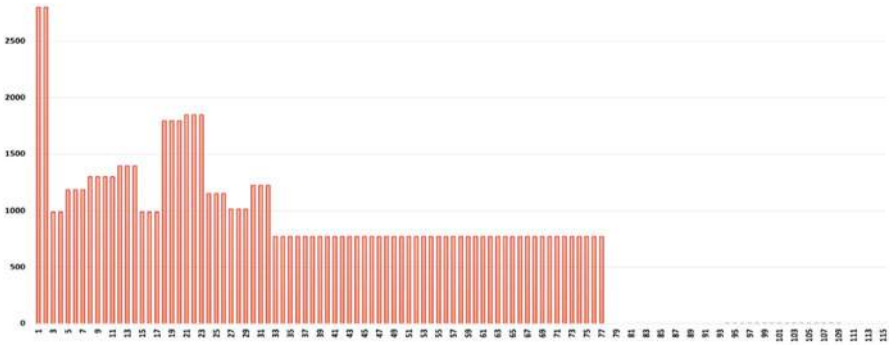


Fig. 15 The number of missing values in each feature

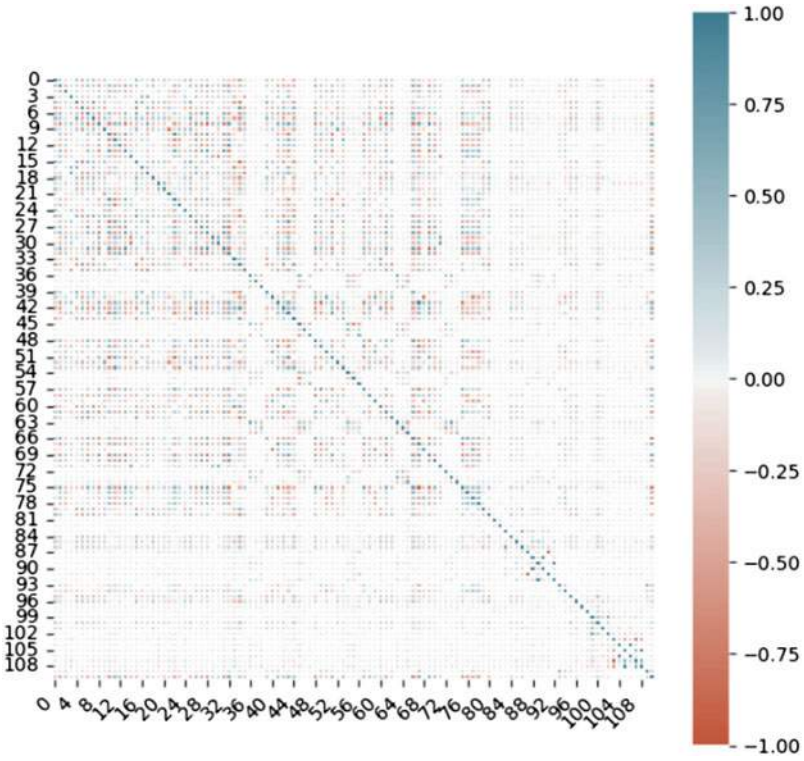


Fig. 16 Correlation matrix

Table 3 Model performance evaluation results

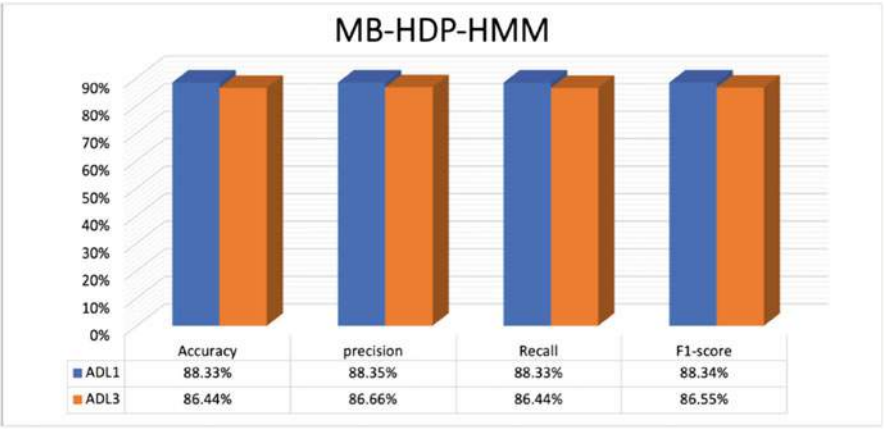
Method	Accuracy	Precision	Recall	F1score
MB-HDP-HMM-VR	86.43	86.66	88.43	86.55
MB-HMM-VR	84.88	84.87	84.88	84.87
GMM-HMM	84.37	84.39	84.37	84.38

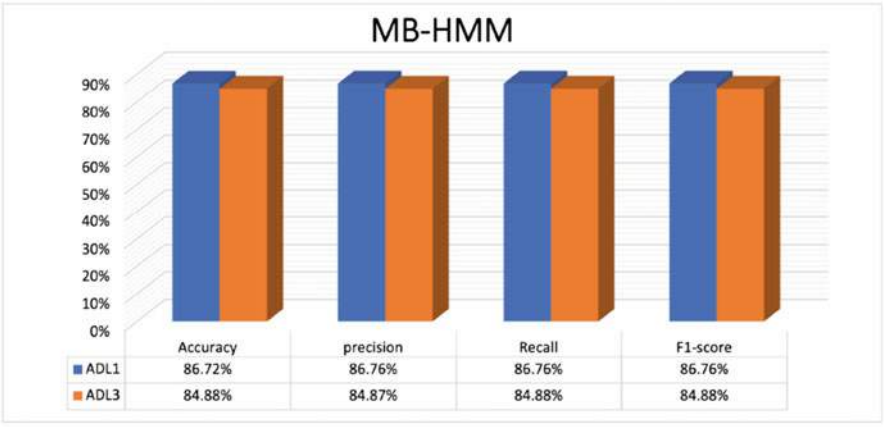
Table 3 illustrates the evaluation results of comparing our proposed model with similar alternatives. MB-HDP-HMM-VR has improved robustness with 86.43, 86.66, 88.43, 86.55 percentage of accuracy, precision, recall, and F1 score, respectively.

In Fig. 17, we compare the results of testing our model on two datasets. We have better results in the first dataset considering these graphs. One of the causes could be having more data points in the first dataset as its size is twice larger than the second dataset (22380 vs. 10379 in each cluster).

5 Conclusion

In this chapter, we proposed multivariate Beta-based hierarchical Dirichlet process hidden Markov models as a new extension of HMMs and applied it to two real datasets. The nonparametric structure of this model assists in handling issues such as defining the number of states. Another motivation to work on this novel algorithm was that we cannot generalize the assumption of Gaussianity in all cases. Over the past decades, other alternative distributions have been applied to numerous real-world datasets. One of the proper choices is multivariate Beta distribution that has demonstrated good potential and flexibility in fitting data. By changing its shape parameter, we could model data with various shapes such as symmetric, asymmetric, skewed ones. In our model, we assumed that emission probability distributions follow multivariate Beta mixture models. This modification may result in having better outputs compared to the conventional cases where we consider GMM-based HMM. To learn the model, we applied variational inference that is slightly faster than fully Bayesian inference and more precise compared to deterministic methods. This promising strategy is successful in various domains. Finally, we evaluated our model on two real datasets, and considering the outcomes, we could infer that our proposed model demonstrates more robustness. In future steps, we can focus on feature selection and integrate it into our model.





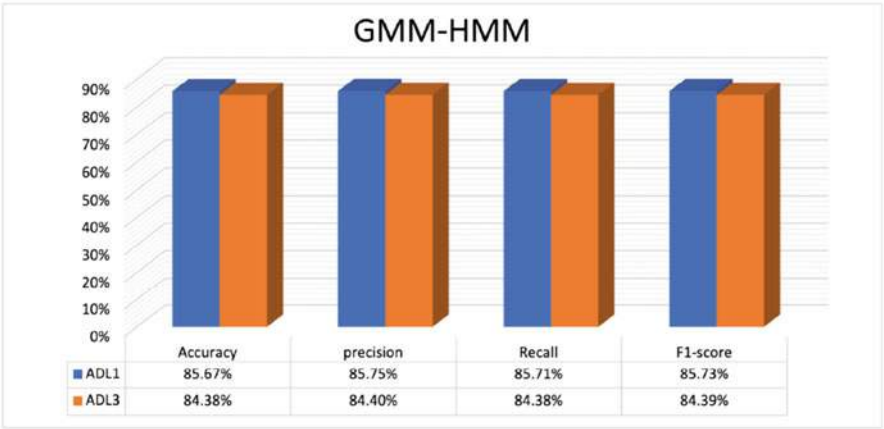


Fig. 17 Results comparison

Acknowledgments The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. M. Boeker, M.A. Riegler, H.L. Hammer, P. Halvorsen, O.B. Fasmer, P. Jakobsen, Diagnosing schizophrenia from activity records using hidden Markov model parameters, in *2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)* (IEEE, New York, 2021), pp. 432–437
2. N.F. Monroy, M. Altuve, Hidden Markov model-based heartbeat detector using different transformations of ECG and ABP signals, in *15th International Symposium on Medical Information Processing and Analysis*, vol. 11330 (International Society for Optics and Photonics, 2020), p. 113300S
3. Q. Huang, D. Cohen, S. Komarzynski, X.-M. Li, P. Innominato, F. Lévi, B. Finkenstädt, Hidden Markov models for monitoring circadian rhythmicity in telemetric activity data. J. R. Soc. Interface **15**(139), 20170885 (2018)
4. N.F. Monroy, M. Altuve, Joint exploitation of hemodynamic and electrocardiographic signals by hidden Markov models for heartbeat detection, in *Latin American Conference on Biomedical Engineering* (Springer, New York, 2019), pp. 208–217
5. J. Kim, S. Singh, E.D. Thiessen, A.V. Fisher, A hidden Markov model for analyzing eye-tracking of moving objects. Behav. Res. Methods **52**(3), 1225–1243 (2020)
6. M. Wang, S. Abdelfattah, N. Moustafa, J. Hu, Deep Gaussian mixture-hidden Markov model for classification of EEG signals. IEEE Trans. Emerg. Top. Comput. Intell. **2**(4), 278–287 (2018)
7. A. Dhar, D.K. Ralph, V.N. Minin, F.A. Matsen IV, A Bayesian phylogenetic hidden Markov model for B cell receptor sequence analysis. PLoS Comput. Biol. **16**(8), e1008030 (2020)
8. H. Ghimatgar, K. Kazemi, M.S. Helfroush, A. Aarabi, An automatic single-channel EEG-based sleep stage scoring method based on hidden Markov model. J. Neurosci. Methods **324**, 108320 (2019)
9. G. Zhang, B. Cai, A. Zhang, J.M. Stephen, T.W. Wilson, V.D. Calhoun, Y.-P. Wang, Estimating dynamic functional brain connectivity with a sparse hidden Markov model. IEEE Trans. Med. Imag. **39**(2), 488–498 (2019)
10. G. Manogaran, V. Vijayakumar, R. Varatharajan, P.M. Kumar, R. Sundarasekar, C.-H. Hsu, Machine learning based big data processing framework for cancer diagnosis using hidden Markov model and GM clustering. Wirel. Pers. Commun. **102**(3), 2099–2116 (2018)
11. R. Rastghalam, H. Danyali, M.S. Helfroush, M.E. Celebi, M. Mokhtari, Skin melanoma detection in microscopic images using HMM-based asymmetric analysis and expectation maximization. IEEE J. Biomed. Health Inf. **25**(9), 3486–3497 (2021)
12. S. Sharma, M. Rattan, An improved segmentation and classifier approach based on HMM for brain cancer detection. Open Biomed. Eng. J. **13**(1), 33–39 (2019)
13. C.J.A. Wolfs, N. Varfalvy, R.A.M. Canters, S.M.J.J.G. Nijsten, D. Hattu, L. Archambault, F. Verhaegen, External validation of a hidden Markov model for gamma-based classification of anatomical changes in lung cancer patients using EPID dosimetry. Med. Phys. **47**(10), 4675–4682 (2020)
14. M. Momenzadeh, M. Sehhati, H. Rabbani, Using hidden Markov model to predict recurrence of breast cancer based on sequential patterns in gene expression profiles. J. Biomed. Inf. **111**, 103570 (2020)
15. H. Zheng, R. Wang, W. Xu, Y. Wang, W. Zhu, Combining a HMM with a genetic algorithm for the fault diagnosis of photovoltaic inverters. J. Power Electron. **17**(4), 1014–1026 (2017)
16. H. Ding, Y. Tian, C. Peng, Y. Zhang, S. Xiang, Inference attacks on genomic privacy with an improved HMM and an RCNN model for unrelated individuals. Inf. Sci. **512**, 207–218 (2020)

17. H. Satori, O. Zealouk, K. Satori, F. ElHaoussi, Voice comparison between smokers and non-smokers using HMM speech recognition system. Int. J. Speech Technol. **20**(4), 771–777 (2017)
18. D. Palaz, M. Magimai-Doss, R. Collobert, End-to-end acoustic modeling using convolutional neural networks for HMM-based automatic speech recognition. Speech Commun. **108**, 15–32 (2019)
19. J. Novoa, J. Wuth, J.P. Escudero, J. Fredes, R. Mahu, N.B. Yoma, DNN-HMM based automatic speech recognition for HRI scenarios, in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction* (2018), pp. 150–159
20. T. Schatz, N.H. Feldman, Neural network vs. HMM speech recognition systems as models of human cross-linguistic phonetic perception, in *Proceedings of the Conference on Cognitive Computational Neuroscience* (2018)
21. J. Novoa, J. Fredes, V. Poblete, N.B. Yoma, Uncertainty weighting and propagation in DNN–HMM-based speech recognition. Comput. Speech Lang. **47**, 30–46 (2018)
22. R. Fatmi, S. Rashad, R. Integlia, Comparing ANN, SVM, and HMM based machine learning methods for American sign language recognition using wearable motion sensors, in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)* (IEEE, New York, 2019), pp. 0290–0297
23. A. Emdadi, C. Eslahchi, Auto-HMM-LMF: feature selection based method for prediction of drug response via autoencoder and hidden Markov model. BMC Bioinformatics **22**(1), 1–22 (2021)
24. J.-H. Zhang, X.-L. Liu, Z.-L. Hu, Y.-L. Ying, Y.-T. Long, Intelligent identification of multi-level nanopore signatures for accurate detection of cancer biomarkers. Chem. Commun. **53**(73), 10176–10179 (2017)
25. A. Silvina, J. Bowles, P. Hall, On predicting the outcomes of chemotherapy treatments in breast cancer, in *Conference on Artificial Intelligence in Medicine in Europe* (Springer, New York, 2019), pp. 180–190
26. M.Z. Uddin, Human activity recognition using segmented body part and body joint features with hidden Markov models. Multimedia Tools Appl. **76**(11), 13585–13614 (2017)
27. Z. Wang, Y. Chen, Recognizing human concurrent activities using wearable sensors: a statistical modeling approach based on parallel HMM. Sensor Rev. (2017). IF 1.583
28. M. Abreu, M. Barandas, R. Leonardo, H. Gamboa, Detailed human activity recognition based on multiple HMM, in *BIOSIGNALS* (2019), pp. 171–178
29. G. Liu, Y. Kang, H. Men, CHAR-HMM: An improved continuous human activity recognition algorithm based on hidden Markov model, in *Mobile Ad-hoc and Sensor Networks: 13th International Conference, MSN 2017, Beijing, December 17–20, 2017, Revised Selected Papers*, vol. 747 (Springer, New York, 2018), p. 271
30. X. Tong, Y. Su, Z. Li, C. Si, G. Han, J. Ning, F. Yang, A double-step unscented Kalman filter and HMM-based zero-velocity update for pedestrian dead reckoning using MEMS sensors. IEEE Trans. Ind. Electron. **67**(1), 581–591 (2019)
31. G. Chalvatzaki, X.S. Papageorgiou, C.S. Tzafestas, P. Maragos, Estimating double support in pathological gaits using an HMM-based analyzer for an intelligent robotic walker, in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (IEEE, New York, 2017), pp. 101–106
32. S. Yu, H. Chen, R.A. Brown, Hidden Markov model-based fall detection with motion sensor orientation calibration: a case for real-life home monitoring. IEEE J. Biomed. Health Inf. **22**(6), 1847–1853 (2017)
33. X. Chen, Z.-X. Wang, X.-M. Pan, HIV-1 tropism prediction by the XGboost and HMM methods. Sci. Rep. **9**(1), 1–8 (2019)
34. S.-T. Pan, W.-C. Li, Fuzzy-HMM modeling for emotion detection using electrocardiogram signals. Asian J. Control **22**(6), 2206–2216 (2020)
35. X. Wang, Y. Liu, Z. Wu, X. Mou, M. Zhou, M.A.G. Ballester, C. Zhang, Automatic labeling of vascular structures with topological constraints via HMM, in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer, New York, 2017), pp. 208–215

36. S. AlZu'bi, S. AlQatawneh, M. ElBes, M. Alsmirat, Transferable HMM probability matrices in multi-orientation geometric medical volumes segmentation. Concurr. Comput. Pract. Exp. **32**(21), e5214 (2020)
37. S.N. Kumar, S. Muthukumar, H. Kumar, P. Varghese, et al., A voyage on medical image segmentation algorithms. Biomed. Res. (0970-938X) (2018)
38. C.-H. Min, Automatic detection and labeling of self-stimulatory behavioral patterns in children with autism spectrum disorder, in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (IEEE, New York, 2017), pp. 279–282
39. S. Ali, F. Mehmood, Y. Ayaz, M. Sajid, H. Sadia, R. Nawaz, An experimental trial: Multi-robot therapy for categorization of autism level using hidden Markov model. J. Educ. Comput. Res. (2021). https://doi.org/10.1177/07356331211040405
40. P.S. Dammu, R.S. Bapi, Temporal dynamics of the brain using variational Bayes hidden Markov models: application in autism, in *International Conference on Pattern Recognition and Machine Intelligence* (Springer, New York, 2019), pp. 121–130
41. M. Chatterjee, N.V. Manyakov, A. Bangerter, D.A. Kaliukhovich, S. Jagannatha, S. Ness, G. Pandina, Learning scan paths of eye movement in autism spectrum disorder, in *Digital Personalized Health and Medicine* (IOS Press, Amsterdam, 2020), pp. 287–291
42. S. Priyadharshini, K. Sivaranjani, Investigating and statistical analysis of autism spectrum disorders: a survey. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **7**(7), 13–15 (2017)
43. J. Van Schependom, D. Vidaurre, L. Costers, M. Sjøgård, M.B. D'hooghe, M. D'haeseleer, V. Wens, X. De Tiège, S. Goldman, M. Woolrich, et al. Altered transient brain dynamics in multiple sclerosis: treatment or pathology? Hum. Brain Mapp. **40**(16), 4789–4800 (2019)
44. N. Esmaili, M. Piccardi, B. Kruger, F. Girosi, Analysis of healthcare service utilization after transport-related injuries by a mixture of hidden Markov models. PLoS One **13**(11), e0206274 (2018)
45. B. Muthu, C.B. Sivaparthipan, G. Manogaran, R. Sundarasekar, S. Kadry, A. Shanthini, A. Dasel, IOT based wearable sensor for diseases prediction and symptom analysis in healthcare sector. Peer-to-Peer Network. Appl. **13**(6), 2123–2134 (2020)
46. A. Vimont, H. Leleu, I. Durand-Zaleski, Machine learning versus regression modelling in predicting individual healthcare costs from a representative sample of the nationwide claims database in France. Eur. J. Health Econ. **23**(2), 211–223 (2021)
47. S. Fine, Y. Singer, N. Tishby, The hierarchical hidden Markov model: analysis and applications. Mach. Learn. **32**(1), 41–62 (1998)
48. E. Fox, E. Sudderth, M. Jordan, A. Willsky, Developing a tempered HDP-HMM for systems with state persistence. MIT LIDS (2007)
49. E.B. Fox, E.B. Sudderth, M.I. Jordan, A.S. Willsky, An HDP-HMM for systems with state persistence, in *Proceedings of the 25th International Conference on Machine Learning* (2008), pp. 312–319
50. E.B. Fox, E.B. Sudderth, M.I. Jordan, A.S. Willsky, A sticky HDP-HMM with application to speaker diarization. Ann. Appl. Stat. **5**(2A), 1020–1056 (2011)
51. A. Bargi, R.Y.D. Xu, M. Piccardi, An online HDP-HMM for joint action segmentation and classification in motion capture data, in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (IEEE, New York, 2012), pp. 1–7
52. N. Raman, S.J. Maybank, Action classification using a discriminative multilevel HDP-HMM. Neurocomputing **154**, 149–161 (2015)
53. A. Bargi, R.Y.D. Xu, M. Piccardi, AdOn HDP-HMM: an adaptive online model for segmentation and classification of sequential data. IEEE Trans. Neural Netw. Learn. Syst. **29**(9), 3953–3968 (2017)
54. J. Zhao, S. Basole, M. Stamp, Malware classification with GMM-HMM models (2021). Preprint. arXiv:2103.02753
55. F. Zhang, S. Han, H. Gao, T. Wang, A Gaussian mixture based hidden Markov model for motion recognition with 3d vision device. Comput. Electr. Eng. **83**, 106603 (2020)
56. F. Tian, Q. Zhou, C. Yang, Gaussian mixture model-hidden Markov model based nonlinear equalizer for optical fiber transmission. Optics Exp. **28**(7), 9728–9737 (2020)

57. Y. Li, B. Hu, T. Niu, S. Gao, J. Yan, K. Xie, Z. Ren, GMM-HMM-based medium-and long-term multi-wind farm correlated power output time series generation method. IEEE Access **9**, 90255–90267 (2021)
58. X. Cheng, B. Huang, J. Zong, Device-free human activity recognition based on GMM-HMM using channel state information. IEEE Access **9**, 76592–76601 (2021)
59. C.L.P. Lim, W.L. Woo, S.S. Dlay, B. Gao, Heartrate-dependent heartwave biometric identification with thresholding-based GMM–HMM methodology. IEEE Trans. Ind. Inf. **15**(1), 45–53 (2018)
60. L. Chen, D. Barber, J.-M. Odobez, Dynamical Dirichlet mixture model. Technical report, IDIAP, 2007
61. R. Nasfi, M. Amayri, N. Bouguila, A novel approach for modeling positive vectors with inverted Dirichlet-based hidden Markov models. Knowledge-Based Syst. **192**, 105335 (2020)
62. E. Epaillard, N. Bouguila, Data-free metrics for Dirichlet and generalized Dirichlet mixture-based HMMs–a practical study. Patt. Recogn. **85**, 207–219 (2019)
63. E. Epaillard, N. Bouguila, Variational Bayesian learning of generalized Dirichlet-based hidden Markov models applied to unusual events detection. IEEE Trans. Neural Netw. Learn. Syst. **30**(4), 1034–1047 (2018)
64. N. Bouguila, E. Epaillard, Data-free metrics for Dirichlet and generalized Dirichlet mixture-based HMMs-a practical study. Patt. Recogn. **85** (2018). https://doi.org/10.1016/j.patcog.2018.08.013
65. N. Manouchehri, N. Bouguila, W. Fan, Nonparametric variational learning of multivariate beta mixture models in medical applications. Int. J. Imag. Syst. Technol. **31**(1) 128–140 (2021)
66. N. Manouchehri, M. Kalra, N. Bouguila, Online variational inference on finite multivariate beta mixture models for medical applications. IET Image Process. **15**(6245), 1869–1882 (2021)
67. D.M. Blei, A. Kucukelbir, J.D. McAuliffe, Variational inference: a review for statisticians. J. Am. Stat. Assoc. **112**(518), 859–877 (2017)
68. A. Zhang, S. Gultekin, J. Paisley, Stochastic variational inference for the HDP-HMM, in *Artificial Intelligence and Statistics* (PMLR, 2016), pp. 800–808
69. Y. Wang, D. Blei, Variational Bayes under model misspecification. Adv. Neural Inf. Process. Syst. **32**, 13357–13367 (2019)
70. A. Vellido, The importance of interpretability and visualization in machine learning for applications in medicine and health care. Neural Comput. Appl. **32**(24), 18069–18083 (2020)
71. D. Gunning, Explainable artificial intelligence (XAI). *Defense Advanced Research Projects Agency (DARPA), nd Web*, 2 (2017)
72. R. Miotto, F. Wang, S. Wang, X. Jiang, J.T. Dudley, Deep learning for healthcare: review, opportunities and challenges. Brief. Bioinform. **19**(6), 1236–1246 (2018)
73. G. Litjens, T. Kooi, B.E. Bejnordi, A.A.A. Setio, F. Ciompi, M. Ghafoorian, J.A. Van Der Laak, B. Van Ginneken, C.I. Sánchez, A survey on deep learning in medical image analysis. Med. Image Anal. **42**, 60–88 (2017)
74. I. Olkin, R. Liu, A bivariate beta distribution. Stat. Probab. Lett. **62**(4), 407–412 (2003)
75. D.M. Blei, M.I. Jordan, Variational inference for Dirichlet process mixtures. Bayesian Anal. **1**(1), 121–143 (2006)
76. Y.W. Teh, M.I. Jordan, M.J. Beal, D.M. Blei, Hierarchical Dirichlet processes. J. Am. Stat. Assoc. **101**(476), 1566–1581 (2006)
77. T.S. Ferguson, A Bayesian analysis of some nonparametric problems. Ann. Stat. **1**(2), 209–230 (1973)
78. R.J. Connor, J.E. Mosimann, Concepts of independence for proportions with a generalization of the Dirichlet distribution. J. Am. Stat. Assoc. **64**(325), 194–206 (1969)
79. T.-T. Wong, Generalized Dirichlet distribution in Bayesian analysis. Appl. Math. Comput. **97**(2–3), 165–181 (1998)
80. H. Ishwaran, L.F. James, Gibbs sampling methods for stick-breaking priors. J. Am. Stat. Assoc. **96**(453), 161–173 (2001)
81. M.J. Beal, *Variational Algorithms for Approximate Bayesian Inference* (University of London, University College London (United Kingdom), 2003)

82. M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, L.K. Saul, An introduction to variational methods for graphical models. Learning in graphical models. Mach. Learn. **37**, 183–233, 105162 (1999)
83. M.M. Ichir, A. Mohammad-Djafari, A mean field approximation approach to blind source separation with l p priors, in *2005 13th European Signal Processing Conference* (IEEE, New York, 2005), pp. 1–4
84. J. Paisley, L. Carin, Hidden Markov models with stick-breaking priors. IEEE Trans. Sig. Proc. **57**(10), 3905–3917 (2009)
85. W. Fan, N. Bouguila, D. Ziou, Variational learning for finite Dirichlet mixture models and applications. IEEE Trans. Neural Netw. Learn. Syst. **23**(5), 762–774 (2012)
86. L. Rabiner, B. Juang, An introduction to hidden Markov models. IEEE ASSP Mag. **3**(1), 4–16 (1986)
87. D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, et al. Collecting complex activity datasets in highly rich networked sensor environments, in *2010 Seventh International COnference on Networked Sensing Systems (INSS)* (IEEE, New York, 2010), pp. 233–240.
88. H. Sagha, S.T. Digumarti, J. del R Millán, R. Chavarriaga, A. Calatroni, D. Roggen, G. Tröster, Benchmarking classification techniques using the opportunity human activity dataset, in *2011 IEEE International Conference on Systems, Man, and Cybernetics* (IEEE, New York, 2011)

Shifted-Scaled Dirichlet-Based Hierarchical Dirichlet Process Hidden Markov Models with Variational Inference Learning

Ali Baghdadi, Narges Manouchehri, Zachary Patterson, and Nizar Bouguila

1 Introduction

The hidden Markov model (HMM) is a type of probabilistic model in which each data point is in a hidden state produced by a probability distribution called an emission probability [1]. This approach has been used in a variety of time series applications and sequential data like anomaly detection [2], facial identification [3], speech recognition [4], machine translation [5], financial analysis [6], healthcare [7], human activity recognition [8], and gesture recognition [9]. The mathematical basis of HMM was initially established by Baum and Petrie [10]. Its primary structure is a Markov chain of latent variables. One of the simplest methods to represent sequential patterns in time series data is to use a Markov chain. This method keeps generality while loosening the assumption of independent identically distributed variables [11]. Depending on the type of data (which can be continuous or discrete), HMM and its emission probability are continuous or discrete [12–15]. Section 2 summarizes the theoretical explanation and main features of HMMs.

The application of HMMs for discrete and Gaussian data has been the focus of most previous research [1, 16]. However, many efforts have recently been made to adapt the learning equations to non-Gaussian continuous data. In fact, they showed that a Gaussian-based HMM is not the best option for modeling some types of data (like proportional data) because of the symmetric aspect and unbounded support of the Gaussian distribution [12, 17–19]. Inspired by these efforts, we are motivated to select an alternative for the emission probability distribution. This work is developed

A. Baghdadi · N. Manouchehri · Z. Patterson · N. Bouguila (✉)
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: ali.baghdadi@mail.concordia.ca; narges.manouchehri@mail.concordia.ca; zachary.patterson@concordia.ca; nizar.bouguila@concordia.ca

N. Bouguila et al. (eds.), *Hidden Markov Models and Applications*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-99142-5_11

based on a continuous HMM framework for modeling proportional data using mixtures of shifted-scaled Dirichlet (SSD) distributions [20]. It is worth mentioning that the Dirichlet distribution is the most commonly used distribution in modeling proportional data. However, the shifted-scaled Dirichlet distribution has previously demonstrated that it performs better than the Dirichlet distribution in modeling this kind of data [21–25]. To the best of our knowledge, the model that we introduce in this work is novel and we call it a "shifted-scaled Dirichlet-based hidden Markov model" (SSD-HMM).

Some extensions of traditional HMMs, such as the hierarchical hidden Markov model [26] and hierarchical Dirichlet process hidden Markov model (HDP-HMM) [27–29], have been developed in the last two decades. Because of its nonparametric structure, HDP-HMM provides a lot of versatility and has been used in a lot of different applications. This method allows us to learn more complicated emission distributions while simultaneously determining the appropriate number of states throughout the learning process. Therefore, as an extension of our novel algorithm, we propose a Bayesian nonparametric approach as the second part of our work and we call it the "SSD-based hierarchical Dirichlet process hidden Markov model" (SSD-HDP-HMM).

The expectation maximization technique is commonly used to train an HMM. However, because this method includes a summation of all conceivable combinations of hidden states and mixture components, it is computationally intractable. Furthermore, using the maximum likelihood method might result in overfitting and convergence to a local rather than a global maximum. There are some alternative approaches for training HMM models such as fully Bayesian methods like Markov Chain Monte Carlo (MCMC). However, MCMC requires incredibly extensive calculations that are extremely time consuming [30–32]. Variational learning (VL) techniques have recently been offered as a computationally tractable way of learning HMMs. VL allows us to overcome the drawbacks of the previously mentioned methods [33, 34]; therefore, we employ this method for learning our models.

Finally, we evaluate our models with two real-life applications, activity recognition (AR) [35] and texture clustering [36]. Action recognition has received a lot of attention in previous decades in application domains such as healthcare monitoring [37], robotics [38], fitness tracking [39], and security [40, 41]. We tested our models on a dataset that includes accelerometer data (collected with a smartphone worn on the waist). Individuals were engaged in a variety of activities, including walking, walking upstairs, walking downstairs, sitting, standing, and laying.

For the second application, we focused on texture clustering. The major elements used to characterize pictures are texture, shape, and color. Many image processing applications rely on texture information, such as medical image processing [42], texture classification [43–46], natural object recognition [47], and so on. These different applications motivated us to test our model on this challenging application with the publicly available UIUC dataset that has already been used in other research [48, 49]. We employ VGG16 as a robust feature selection approach to extract features from UIUC images and then compare our proposed model with other similar alternatives. This technique is a popular neural network architecture that has been already tested in feature selection tasks [50–52].

In summary, the main contributions of this work are:

- Considering mixtures of SSD distribution as the emission probability distribution for HMM and showing its outperformance in modeling proportional data in comparison with Dirichlet and Gaussian mixture models
- The entire derivation of the SSD-HMM model equations
- The entire derivation of the SSD-HDP-HMM model equations

The rest of this chapter is organized as follows: HMM is discussed in Sect. 2. Section 3 is devoted to parameter estimation using variational learning for SSD-HMM and SSD-HDP-HMM models. We represent the results of analyzing our suggested models in Sect. 4, and then we conclude in Sect. 5.

2 Hidden Markov Models

Hidden Markov models are a common technique for modeling time series data. They have been used in speech recognition systems, text clustering, and pattern recognition applications for decades. We can consider HMMs as a generalized version of mixture models. That is, the probability density functions produced by an HMM across all observable states can be seen as a mixture of densities formed by each state [19, 53]. The hidden Markov model is defined by two basic features; first, it presupposes that an observation at time t is the result of a process in state h_t which is hidden from the observer. Second, the present state h_t, given the value of $h_t - 1$, is independent of all previous states of time $t - 1$. The second feature is called the Markov property. To develop our HMM model, we introduce some notation. $\mathcal{X} = \{X_1, \ldots, X_T\}$ is the generated sequence of observations by hidden states $S = \{s_1, \ldots, s_t, \ldots, s_T\}$ and $s_t \in [1, N]$, where N is the number of states. $A = \left\{A_{ii'} = P\left(s_t = i' \mid s_{t-1} = i\right)\right\}$ is transition probabilities matrix that presents the probabilities of transition between the states. $C = \left\{C_{ij} = P\left(m_t = j \mid s_t = i\right)\right\}$ is the emission probabilities matrix for continuous case (CHMM), where $i \in [1, N]$, $j \in [1, K]$. The number of mixture components in set $L = \{m_1, \ldots, m_t, \ldots, m_K\}$ is denoted by K which is assumed to be uniform for all the states. The initial probability vector, π_i, represents the probability of starting the observation sequence from state i. In summary, using all the abovementioned notation, an HMM is defined as $\lambda = \{A, C, \pi, \Theta\}$, where Θ is the set of mixture model parameters [1].

3 Variational Learning

3.1 Shifted-Scaled Dirichlet-Based Hidden Markov Model

In this section, we start with an explanation of mixtures of shifted-scaled Dirichlet distributions and then derive the equations of the variational approach to update

our model's parameters. As we mentioned before, in the previous work, emission probability distributions for continuous observations are frequently assumed to have a Gaussian distribution [54, 55]. The ability of the Dirichlet and scaled Dirichlet (SD) [56, 57] mixture models to fit proportional data motivated us to use a more general form of them called the shifted-scaled Dirichlet mixture model as the emission probability for HMM [20–25, 58–61].

In variational learning, all of the parameters including HMM parameters (A, C, and π) and emission distribution parameters $\theta = \{\alpha_{ijl}, \beta_{ijl}, \tau_{ij}\}$ are treated as random variables. We consider $\mathcal{X} = (\overrightarrow{X}_1, \ldots, \overrightarrow{X}_t, \ldots, \overrightarrow{X}_T)$ as a set of T independent identically distributed observations in which each $\overrightarrow{X}_t$ is a D-dimensional positive vector which is generated from a mixture of SSD distributions. The finite SSD mixture model which is a linear combination of K components is expressed as follows:

$$p\left(\overrightarrow{X}_t \mid \overrightarrow{\mathcal{W}}, \overrightarrow{\theta}\right) = \sum_{j=1}^{K} \mathcal{W}_j \, p(\overrightarrow{X}_t \mid \overrightarrow{\theta_j}) \tag{1}$$

where $\mathcal{W}_j$ is a mixing coefficient (weight) which should satisfy two conditions, $\sum_{j=1}^{K} \mathcal{W}_j = 1$ and $0 < \mathcal{W}_j < 1$. As a result, the likelihood function of the SSD mixture model is

$$p(\mathcal{X} \mid \overrightarrow{\mathcal{W}}, \overrightarrow{\theta_j}) = \prod_{t=1}^{T} \left\{ \sum_{j=1}^{K} \mathcal{W}_j \, p(\overrightarrow{X_t} \mid \overrightarrow{\theta_j}) \right\} \tag{2}$$

Also, $p(\overrightarrow{X}_t \mid \overrightarrow{\theta_j})$ is a mixture component with parameter θ_j that in our model is an SSD distribution that is defined as follows [20]:

$$p(\overrightarrow{X_t} \mid \overrightarrow{\theta_j}) = \frac{\Gamma(\alpha_{ij}+)}{\prod_{l=1}^{D} \Gamma(\alpha_{ijl})} \frac{1}{\tau_{ij}^{D-1}} \frac{\prod_{l=1}^{D} \beta_{ijl}^{-\frac{\alpha_{ijl}}{\tau_{ij}}} X_{tl}^{(\frac{\alpha_{ijl}}{\tau_{ij}}-1)}}{\left(\sum_{l=1}^{D} (\frac{X_{tl}}{\beta_{ijl}}) \frac{1}{\tau_{ij}} \right)^{\alpha_{ij}+}} \tag{3}$$

where $\alpha = \{\alpha_{ijl}\}_{i,j,l}^{N,K,D}$, $\beta = \{\beta_{ijl}\}_{i,j,l}^{N,K,D}$, and $\tau = \{\tau_{ij}\}_{i,j}^{N,K}$ are positive SSD parameters. Also $X_t > 0$, $\sum_{l=1}^{D} X_{tl} = 1$, and $\alpha_{ij}+ = \sum_{l=1}^{D} \alpha_{ijl}$. $\Gamma(.)$ indicates the Gamma function which is $\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} dx$.

For each X_{vt} (where v is the vth observed vector), we introduce a latent variable Z_{vij} that shows which cluster and state are assigned to X_{vt}. In other words, $Z_{vij} = 1$ if the X_{vt} belongs to state i and cluster j, otherwise $Z_{vij} = 0$. Also, Z_{vij} must satisfy this condition $\sum_{j=1}^{K} Z_{vij} = 1$.

In HMMs, the probability of the complete data can be stated as follows for given model parameters:

$$p(X, S, L \mid A, C, \pi, \theta) = \pi_{s_1} \left[\prod_{t=2}^{T} A_{s_{t-1}, s_t} \right] \left[\prod_{t=1}^{T} C_{s_t, m_l} p\left(X_t \mid \theta_{s_t, m_l}\right) \right] \tag{4}$$

where X is a sequence of T observations, S is the set of hidden states, and L stands for the set of mixture components. It is worth mentioning that for the sake of simplification, the model is derived for a single observation series. In order to incorporate more observation sequences (which is recommended to prevent overfitting), the related equations need to be updated to include a summation of these sequences. Therefore, the likelihood of X given model parameters is expressed as follows:

$$p(X \mid A, C, \pi, \theta) = \sum_{S} \sum_{L} \pi_{s_1} \left[\prod_{t=2}^{T} A_{s_{t-1}, s_t} \right] \left[\prod_{t=1}^{T} C_{s_t, m_l} p\left(X_t \mid \theta_{s_t, m_l}\right) \right] \tag{5}$$

A precise computation of this equation is impossible because it requires the summation of all possible combinations of mixture components and states. The most common approach for solving it is to use the Baum-Welch algorithm to maximize the data likelihood with regard to the model parameters [1]. However, there are several flaws with this strategy such as the potential for overfitting and the lack of a convergence guarantee. Another tested solution to compute intractable equations (5) is variational learning. This approach calculates the marginal likelihood of data using an approximate distribution Q of the true posterior p. In the SSD-HMM model, data marginal likelihood is expressed as

$$p(X) = \tag{6}$$
$$\int dA dC d\pi d\alpha d\beta d\tau \sum_{S,L} p(A, C, \pi, \alpha, \beta, \tau) \times p(X, S, L \mid A, C, \pi, \alpha, \beta, \tau)$$

The variational learning is based on this equation [34]:

$$\ln(p(X)) = \mathcal{L}(Q) + \mathrm{KL}(Q \| P) \tag{7}$$

where $\mathcal{L}(Q)$ is the variational lower bound for $\ln p(X)$ and defined by

$$\mathcal{L}(Q) = \int Q(\Theta) \ln\left(\frac{p(X \mid \Theta) p(\Theta)}{Q(\Theta)}\right) d\Theta \tag{8}$$

and the Kullback–Leibler divergence between the approximation Q and the posterior p is represented by $\mathrm{KL}(Q\|P)$:

$$KL(Q \mid\mid P) = -\int Q(\Theta) \ln(\frac{Q(\Theta)}{p(\Theta \mid X)}) d\Theta \tag{9}$$

where $\Theta = \{A, C, \pi, \alpha, \beta, \tau, S, L\}$. Minimizing KL allows the best approximation of the true posterior p, and due to the fact that $KL \geq 0$, this can be accomplished by maximizing $\mathcal{L}(Q)$. Having Q and (8), we can take the lower bound as follows [12]:

$$\begin{aligned}
\ln(p(X)) &= \ln\{\int dAdCd\pi d\alpha d\beta d\tau \sum_{S,L} p(A, C, \pi, \alpha, \beta, \tau) \\
&\quad \times p(X, S, L \mid A, C, \pi, \alpha, \beta, \tau)\} \\
&\geq \int dAdCd\pi d\alpha d\beta d\tau \sum_{S,L} Q(A, C, \pi, \alpha, \beta, \tau, S, L) \\
&\quad \times \ln\left\{\frac{p(A, C, \pi, \alpha, \beta, \tau) p(X, S, L \mid A, C, \pi, \alpha, \beta, \tau)}{Q(A, C, \pi, \alpha, \beta, \tau, S, L)}\right\}
\end{aligned} \tag{10}$$

Now, using the mean-field assumption [62], we take a restricted family of distributions to be able to calculate $Q(\Theta)$. Therefore, we have factorized $Q(\Theta)$:

$$Q(A, C, \pi, \alpha, \beta, \tau, S, L) = Q(A)Q(C)Q(\pi)Q(\alpha)Q(\beta)Q(\tau)Q(S, L) \tag{11}$$

Using (10) and (11), the lower bound can be written as follows:

$$\begin{aligned}
&\ln p(X) \geq \sum_{S,L} \int dAdCd\pi d\alpha d\beta d\tau Q(\pi)Q(A)Q(C)Q(\alpha)Q(\beta)Q(\tau)Q(S, L) \\
&\times \{\ln(p(\pi)) + \ln(p(A)) + \ln(p(C)) + \ln(p(\alpha)) + \ln(p(\beta)) + \ln(p(\tau)) \\
&+ \ln\left(\pi_{s_1}\right) + \sum_{t=2}^{T} \ln\left(A_{s_{t-1}, s_t}\right) + \sum_{t=1}^{T} \ln\left(C_{s_t, m_t}\right) + \sum_{t=1}^{T} \ln\left(f\left(X_t \mid \theta_{s_t, m_t}\right)\right) \\
&- \ln(Q(S, L)) - \ln(Q(A)) - \ln(Q(\pi)) - \ln(Q(C)) - \ln(Q(\alpha)) \\
&- \ln(Q(\beta)) - \ln(Q(\tau))\} = F(Q(\pi)) + F(Q(C)) + F(Q(A)) \\
&+ F(Q(S, L)) + F(Q(\alpha)) + F(Q(\beta)) + F(Q(\tau))
\end{aligned} \tag{12}$$

The above lower bound, in general, has several maxima; hence, it is not convex. As a result, the solution depends on initialization. We are now going to define prior distributions for model parameters to be able to evaluate (12). The priors for the parameters A, C, and π are selected as Dirichlet distributions $\mathcal{D}$ since their

coefficients are positive and less than one:

$$p(\pi) = \mathcal{D}(\pi_1, \ldots, \pi_N \mid \phi_1^{\pi}, \ldots, \phi_N^{\pi}) \tag{13}$$

$$p(A) = \prod_{i=1}^{N} \mathcal{D}(A_{i_1}, \ldots, A_{i_N} \mid \phi_{i_1}^{A}, \ldots, \phi_{i_N}^{A}) \tag{14}$$

$$p(C) = \prod_{i=1}^{N} \mathcal{D}(C_{i_1}, \ldots, C_{i_K} \mid \phi_{i_1}^{C}, \ldots, \phi_{i_K}^{C}) \tag{15}$$

The conjugate priors for SSD parameters are chosen as follows [63]:

$$p(\alpha_{ijl}) = \mathcal{G}(\alpha_{ijl} \mid u_{ijl}, v_{ijl}) = \frac{v_{ijl}^{u_{ijl}}}{\Gamma(u_{ijl})} \alpha_{ijl}^{u_{ijl}-1} e^{-v_{ijl}\alpha_{ijl}} \tag{16}$$

$$p(\beta_{ijl}) = \mathcal{D}(\beta_{ijl} \mid \overrightarrow{h_{ij}}) = \frac{\Gamma(\sum_{l=1}^{D} h_{ijl})}{\prod_{l=1}^{D} \Gamma(h_{ijl})} \prod_{l=1}^{D} \beta_{ijl}^{h_{ijl}-1} \tag{17}$$

$$p(\tau_{ij}) = \mathcal{G}(\tau_{ij} \mid q_{ij}, s_{ij}) = \frac{q_{ij}^{s_{ij}}}{\Gamma(q_{ij})} \tau_{ij}^{q_{ij}-1} e^{-s_{ij}\tau_{ij}} \tag{18}$$

where u_{ijl}, v_{ijl}, h_{ijl}, q_{ij}, and s_{ij} are positive SSD hyperparameters and $\mathcal{G}$ is the Gamma distribution. Since the variables are considered statistically independent, the prior distributions for SSD parameters are

$$p(\overrightarrow{\alpha}) = \prod_{i=1}^{N} \prod_{j=1}^{K} \prod_{l=1}^{D} p(\alpha_{ijl}) \tag{19}$$

$$p(\overrightarrow{\beta}) = \prod_{i=1}^{N} \prod_{j=1}^{K} \prod_{l=1}^{D} p(\beta_{ijl}) \tag{20}$$

$$p(\overrightarrow{\tau}) = \prod_{i=1}^{N} \prod_{j=1}^{K} p(\tau_{ij}) \tag{21}$$

Now we can optimize each factor $F(Q(.))$ in (12) by maximizing the lower bound with respect to that factor because they are independent of each other. First we will optimize $Q(\pi)$, $Q(A)$, and $Q(C)$ which are independent of the SSD parameters and

have already been studied [12, 64, 65]. Therefore, according to the previous work, we can optimize $Q(A)$ using the Gibbs inequality in the following procedure:

$$F(Q(A)) = \int dA\, Q(A) \ln \left[\frac{\prod_{i=1}^{N} \prod_{i'=1}^{N} A_{ii'}^{w_{ii'}^{A}-1}}{Q(A)} \right] \tag{22}$$

$$Q(A) = \prod_{i=1}^{N} \mathcal{D}\left(A_{i1}, \ldots, A_{iN} \mid w_{i1}^{A}, \ldots, w_{iN}^{A}\right) \tag{23}$$

where

$$w_{ij}^{A} = \sum_{t=2}^{T} \gamma_{ijt}^{A} + \phi_{ij}^{A} \tag{24}$$

$$\gamma_{ijt}^{A} \triangleq Q\left(s_{t-1} = i, s_t = j\right) \tag{25}$$

Similarly for the $Q(\pi)$ and $Q(C)$, we have

$$Q(\pi) = \mathcal{D}\left(\pi_1, \ldots, \pi_N \mid w_1^{\pi}, \ldots, w_N^{\pi}\right) \tag{26}$$

$$w_i^{\pi} = \gamma_i^{\pi} + \phi_i^{\pi} \tag{27}$$

$$\gamma_i^{\pi} \triangleq Q\left(s_1 = i\right) \tag{28}$$

and

$$Q(C) = \prod_{i=1}^{N} \mathcal{D}\left(C_{i1}, \ldots, C_{iK} \mid w_{i1}^{C}, \ldots, w_{iK}^{C}\right) \tag{29}$$

$$w_{ij}^{C} = \sum_{t=1}^{T} \gamma_{ijt}^{C} + \phi_{ij}^{C} \tag{30}$$

$$\gamma_{ijt}^{C} \triangleq Q\left(s_t = i, m_t = j\right) \tag{31}$$

The forward–backward procedure may then be used to derive the values of responsibilities γ_{ijt}^{A}, γ_{i}^{π}, and γ_{ijt}^{C} [66].

The next step is optimizing $Q(\alpha)$. Using (19) and (12), we have

$$F(Q(\alpha)) = \int d\alpha\, Q(\alpha) \ln\{\frac{p(\overrightarrow{\alpha}) \prod_{t=1}^{T} p(X_t \mid \alpha_{ijl})^{\gamma^{c}{}_{ijt}}}{Q(\alpha)}\} \tag{32}$$

Similarly for $Q(\beta)$ and $Q(\tau)$, we can obtain

$$F(Q(\beta)) = \int d\beta\, Q(\beta) \ln\{\frac{p(\overrightarrow{\beta}) \prod_{t=1}^{T} p(X_t \mid \beta_{ijl})^{\gamma^{c}{}_{ijt}}}{Q(\beta)}\} \tag{33}$$

and

$$F(Q(\tau)) = \int d\tau\, Q(\tau) \ln\{\frac{p(\overrightarrow{\tau}) \prod_{t=1}^{T} p(X_t \mid \tau_{ij})^{\gamma^{c}{}_{ijt}}}{Q(\tau)}\} \tag{34}$$

The optimal values for $Q(\alpha)$, $Q(\beta)$, and $Q(\tau)$ can be calculated by

$$Q\left(\alpha_{ijl}\right) = \prod_{i=1}^{N} \prod_{j=1}^{K} \prod_{l=1}^{D} \mathcal{G}(\alpha_{ijl} \mid u_{ijl}^{\star}, v_{ijl}^{\star}) \tag{35}$$

$$Q\left(\beta_{ijl}\right) = \prod_{i=1}^{N} \prod_{j=1}^{K} \prod_{l=1}^{D} \mathcal{D}(\beta_{ijl} \mid \overrightarrow{h}_{ijl}^{\star}) \tag{36}$$

$$Q\left(\tau_{ij}\right) = \prod_{i=1}^{N} \prod_{j=1}^{K} \mathcal{G}(\tau_{ij} \mid q_{ij}^{\star}, s_{ij}^{\star}) \tag{37}$$

We may calculate the above hyperparameters using Appendix 1. The $\star$ superscript illustrates the optimized value of these parameters.

Finally, we optimize the value of $Q(S, L)$. We can write $F(Q(S, L))$ as follows [12]:

$$F(Q(S, L)) = \sum_{S,L} Q(S, L) \times \ln\left(\frac{\pi_{s_1}^{\star} \prod_{t=2}^{T} A_{s_{t-1},s_t}^{\star} \prod_{t=1}^{T} C_{s_t,m_t}^{\star} p^{\star}\left(X_t \mid \theta_{s_t,m_t}\right)}{Q(S, L)}\right) \tag{38}$$

The optimized $Q(S, L)$ is written as:

$$Q(S, L) = \frac{1}{\Omega} \pi_{s_1}^{\star} \prod_{t=2}^{T} A_{s_{t-1},s_t}^{\star} \prod_{t=1}^{T} C_{s_t,l_t}^{\star} p^{\star}\left(X_t \mid \theta_{s_t,l_t}\right) \tag{39}$$

where

$$\Omega = \sum_{S,L} \pi^{\star}_{s_1} \prod_{t=2}^{T} A^{\star}_{s_{t-1},s_t} \prod_{t=1}^{T} C^{\star}_{s_t,l_t} p^{\star}\left(X_t \mid \theta_{s_t,l_t}\right) \tag{40}$$

and

$$\pi_i^{\star} \triangleq \exp\left[\mathbb{E}_Q \ln\left(\pi_i\right)_{Q(\pi)}\right] \tag{41}$$

$$\pi_i^{\star} = \exp\left[\Psi\left(w_i^{\pi}\right) - \Psi\left(\sum_i w_i^{\pi}\right)\right]$$

$$A^{\star}_{jj'} \triangleq \exp\left[\mathbb{E}_Q \ln\left(A_{jj'}\right)_{Q(A)}\right] \tag{42}$$

$$A^{\star}_{jj'} = \exp\left[\Psi\left(w^A_{jj'}\right) - \Psi\left(\sum_{j'} w^A_{jj'}\right)\right]$$

$$C^{\star}_{ij} \triangleq \exp\left[\mathbb{E}_Q \ln\left(C_{ij}\right)_{Q(C)}\right] \tag{43}$$

$$C^{\star}_{ij} = \exp\left[\Psi\left(w^C_{ij}\right) - \Psi\left(\sum_{j} w^C_{ij}\right)\right]$$

and also

$$\ln p^{\star}\left(X_t \mid \theta_{s_t,l_t}\right) = \int Q(\theta) \ln(p\left(X_t \mid \theta_{s_t,l_t}\right)) d\theta \tag{44}$$

In this work, $p\left(X_t \mid \theta_{s_t,l_t}\right) = \left[SSD(\alpha,\beta,\tau)\right]^{\gamma^C_{ijt}}$ with SSD is defined in (3). Therefore, we have

$$\ln p^{\star}\left(X_t \mid \theta_{s_t,l_t}\right) = \gamma^C_{ijt} \int Q(\theta) \ln\left(\frac{\Gamma(\alpha_{ij}+)}{\prod_{l=1}^{D}\Gamma(\alpha_{ijl})}\right) d\alpha \quad + \tag{45}$$

$$\gamma^C_{ijt} \int Q(\theta) \ln\left(\frac{1}{\tau_{ij}^{D-1}} \frac{\prod_{l=1}^{D} \beta_{ijl}^{-\frac{\alpha_{ijl}}{\tau_{ij}}} X_{tl}^{(\frac{\alpha_{ijl}}{\tau_{ij}}-1)}}{\left(\sum_{l=1}^{D} (\frac{X_{tl}}{\beta_{ijl}}) \frac{1}{\tau_{ij}}\right)^{\alpha_{ij}+}}\right)$$

The expected value of the first part of this equation is analytically intractable. So, we use the lower bound introduced in [34] as an approximation for it:

$$\mathbb{E}_Q \ln\left(\frac{\Gamma(\alpha_{ij}+)}{\prod_{l=1}^{D}\Gamma(\alpha_{ijl})}\right) \geq \bar{\alpha}_{ijl}\ln\left(\alpha_{ijl}\right)\left\{\Psi\left(\sum_{l=1}^{D}\bar{\alpha}_{ijl}\right) - \Psi\left(\bar{\alpha}_{ijl}\right) + \sum_{d=1,d\neq l}^{D}\bar{\alpha}_{ijd}\Psi'\left(\sum_{l=1}^{D}\bar{\alpha}_{ijl}\right)\left(\mathbb{E}_Q\ln\left(\alpha_{ijd}\right) - \ln\left(\bar{\alpha}_{ijd}\right)\right)\right\} \tag{46}$$

The second integral of (45) can be rewritten as follows:

$$\mathbb{E}_Q \ln\left(\frac{1}{\tau_{ij}^{D-1}}\frac{\prod_{l=1}^{D}\beta_{ijl}^{-\frac{\alpha_{ijl}}{\tau_{ij}}}X_{tl}^{(\frac{\alpha_{ijl}}{\tau_{ij}}-1)}}{\left(\sum_{l=1}^{D}\left(\frac{X_{tl}}{\beta_{ijl}}\right)^{\frac{1}{\tau_{ij}}}\right)^{\alpha_{ij}+}}\right) = -(D-1)\bar{\tau}_{ij}+ \tag{47}$$

$$\sum_{l=1}^{D}\left\{\left(-\frac{\bar{\alpha}_{ijl}}{\bar{\tau}_{ij}}\right)\ln(\beta_{ijl}) + \left(\frac{\bar{\alpha}_{ijl}}{\bar{\tau}_{ij}} - 1\right)\ln(X_{tl})\right\} -$$

$$(\alpha_{ij}+)\ln\left(\sum_{l=1}^{D}\left(\frac{X_{tl}}{\beta_{ijl}}\right)^{\frac{1}{\tau_{ij}}}\right)$$

The last term in the above equation is again analytically intractable, following [63], and we take the lower bound described below as its approximation:

$$\ln\left(\sum_{l=1}^{D}\left(\frac{X_{tl}}{\beta_{ijl}}\right)^{\frac{1}{\tau_{ij}}}\right) \geq \frac{-\ln\tau_{ij}}{\bar{\tau}_{ij}}\frac{\sum_{l=1}^{D}\left(\frac{x_{tl}}{\bar{\beta}_{ijl}}\right)^{\frac{1}{\bar{\tau}_{ij}}}\ln\left(\frac{x_{tl}}{\bar{\beta}_{ijl}}\right)}{\sum_{l=1}^{D}\left(\frac{x_{tl}}{\bar{\beta}_{ijl}}\right)^{\frac{1}{\bar{\tau}_{ij}}}} + const. \tag{48}$$

It is worth mentioning that by comparing (40) and (5), we can see that this equation represents the estimated likelihood of the improved model, which the forward–backward method can efficiently compute. Because $F(Q(.))$ reflects the model's log-marginal likelihood, the number of states N and the number of mixture components K may be calculated using it as a model selection criterion in the proposed model. Consequently, we may run our variational learning algorithm with various N and K values and choose ones that result in the greatest $F(Q)$.

Algorithm 1 presents our suggested algorithm for learning the SSD-HMM model using the variational inference approach.

Algorithm 1 Variational learning of SSD-HMM

1. Initialize ϕ^A, ϕ^C, and ϕ^π.
2. Initialize u_{ijl}, v_{ijl}, h_{ijl}, q_{ij} and s_{ij}.
3. Draw the initial responsibilities γ^A, γ^C, and γ^π from prior distributions using (13), (14), and (15).
4. Calculate w^A, w^C, and w^π with (24), (30) and (27).
5. Initialize A, C, and π using (41), (42), and (43).
6. **repeat**
7. Calculate data likelihood using X, u_{ijl}, v_{ijl}, h_{ijl}, q_{ij} and s_{ij} using (3).
8. Calculate responsibilities γ^A, γ^C and γ^π using forward–backward procedure with (25) , (31), and (28).
9. Update hyperparameters using Appendix 1.
10. Update w^A, w^C, and w^π using γ^A, γ^C, and γ^π with (24), (30) and (27).
11. Update A, C, and π using w^A, w^C, w^π with (41), (42), and (43).
12. **until convergence criterion is reached**.

3.2 Shifted-Scaled-Based Hierarchical Dirichlet Process Hidden Markov Model

The hierarchical Dirichlet process is built on a hierarchical framework that employs several Dirichlet processes (DPs) [67–69]. Our proposed model, SSD-HDP-HMM, uses the hierarchical Dirichlet process over hidden Markov models. This structure comprises at least two layers, with the base measure of the DP dispersed by other DPs at each level. For the purpose of simplicity, we will use a two-level HDP model in this work following previously suggested HDP-HMM models [27, 29]. As the first layer, we take G_0 to be a top-level (global level) Dirichlet process. G_0 has H as its base distribution and γ as the concentration parameter. Therefore, we can write $G_0 \sim \mathrm{DP}(\gamma, H)$. Moreover, G_0 is the base distribution of an unlimited number of second-level (local level) Dirichlet processes in the HDP. Thus, G_0 is shared between all the i states. A grouped dataset $\mathcal{X}$ with N sets exists at the second level. Each set has a G_i with $i \in \{1, \ldots, N\}$, where $G_i \sim \mathrm{DP}(\lambda, G_0)$. In our SSD-HDP-HMM model, G_i is the transition distribution for state i, where N is the number of states.

Stick-breaking construction will be used in the creation of our model since it is a very straightforward method to HDP model implementation [67, 70]. Since we have a two-level HDP, we use two stick-breaking structures in this work, one for the global level and the other one for the local level. Therefore, applying stick-breaking

construction, the global level distribution G_0 would be expressed by

$$G_0 = \sum_{i=1}^{\infty} \psi_i \delta_{\theta_i}, \quad \theta_i \sim H, \quad \sum_{i=1}^{\infty} \psi_i = 1 \tag{49}$$
$$\psi_i = \psi_i' \prod_{s=1}^{i-1} (1 - \psi_s'), \quad \psi_i' \sim \text{Beta}(1, \gamma)$$

θ_i is a set of independent random variables derived from the base distribution H which in our model is an SSD distribution. δ_{θ_i} represents an atom at θ_i which is accessible for all G_i. Using the same manner, we can obtain the local level for the infinite number of DP's G_i using stick-breaking construction as follows:

$$G_i = \sum_{j=1}^{\infty} \varepsilon_{ij} \delta_{\varpi_{ij}}, \quad \varpi_{ij} \sim G_0, \quad \sum_{j=1}^{\infty} \varepsilon_{ij} = 1 \tag{50}$$
$$\varepsilon_{ij} = \varepsilon_{ij}' \prod_{s=1}^{j-1} (1 - \varepsilon_{is}'), \quad \varepsilon_{ij}' \sim \text{Beta}(1, \lambda)$$

Following the previous work in [71], in order to map two HDP levels together, we take a binary latent variable $W_{ij,i'}$, which is equal to 1 if ϖ_{ij} is associated with $\theta_{i'}$; otherwise, it is equal to 0. Also, in order to produce a sequence of data $\mathcal{X} = \{X_1, \ldots, X_T\}$ for our HMM framework, we consider first $\theta' = \{\theta_1', \ldots, \theta_T'\}$ as a sequence of parameters. Then, we draw θ_1' from G_0 and the rest of the parameters from G_{Z_t}, $\theta_{i+1}' \sim G_{Z_t}$, where Z_t is a state index that equals i if $\theta_t' = \theta_i$. Finally, the sequence of data $\mathcal{X}$ is generated using these θ' parameters, $X_t \sim P(X \mid \theta_t')$. Because each sequence is modeled independently, the joint likelihood of the SSD-HDP-HMM can be written as [27]

$$p(X, \theta, \psi', \varepsilon', W, Z) = p(\theta) p(\psi') p(\varepsilon') p(W \mid \psi') \times \prod_{v} p\left(Z_v \mid \varepsilon', W\right) \prod_{t} p\left(X_{vt} \mid \theta, Z_{vt}\right) \tag{51}$$

where v is the number of the observed sequences. As we mentioned in the previous section, in the variational learning approach, we are trying to find an estimation $Q(\Theta)$ for the true posterior $p(\Theta \mid \mathcal{X})$ since it is intractable. By applying mean-field theory, we can rewrite $Q(\Theta)$ as disjoint tractable components:

$$Q(\theta, \psi', \varepsilon', W, Z) = Q(\theta) Q(\psi') Q(\varepsilon') Q(W) Q(Z) = \prod_{i} Q\left(\theta_i\right) Q\left(\psi_i'\right) \prod_{i,j} Q\left(\varepsilon_{ij}'\right) Q\left(W_{ij}\right) \prod_{v} Q\left(Z_v\right) \tag{52}$$

where $\Theta = \{\theta, \psi', \varepsilon', W, Z\}$. Then, using $Q(\Theta)$, we maximize the lower bound $\mathcal{L}(Q)$ introduced in (10) with respect to each Q factor to minimize the KL divergence between Q and the posterior P. In the SSD-HDP-HMM model, we need the transition matrix $A_{ii'}$ to be able to calculate $Q(\theta)$ and $Q(Z)$. However, we have a challenge for this term:

$$\mathbb{E}_Q \ln A_{ii'} = \mathbb{E}_Q \ln \sum_j W_{iji'} \varepsilon_{ij} \tag{53}$$

Because we need to sum all of the sticks allocated to atom θ_i' (for atom θ_i stick-breaking construction), this expectation is not tractable. Therefore, according to what they did in [27], we derive a lower bound for solving this issue as follows:

$$\begin{aligned} \mathbb{E}_Q \ln \sum_j W_{iji'} \varepsilon_{ij} &\geq \mathbb{E}_Q \ln \sum_j W_{iji'} e^{\mathbb{E}_Q \ln \varepsilon_{ij}} \\ &\approx \ln \sum_j \mathbb{E}_Q \left[W_{iji'} \right] e^{\mathbb{E}_Q \ln \varepsilon_{ij}} \end{aligned} \tag{54}$$

Therefore,

$$\begin{aligned} A_{ii'}^{\star} &= \exp \left\{ \mathbb{E}_Q \ln A_{ii'} \right\} \\ &\approx \exp \left\{ \ln \sum_j \mathbb{E}_Q \left[W_{iji'} \right] e^{\mathbb{E}_Q \ln \varepsilon_{ij}} \right\} \end{aligned} \tag{55}$$

The other solution is that we could take the variational distribution $Q(W)$ and $Q(\varepsilon)$ instead of the above approximation to obtain $\mathbb{E}_Q \ln A_{ii'}$. The performance would stay almost the same, but the algorithm would be more time consuming in this case [27].

3.2.1 Update $Q(Z_v)$ and $Q(\theta_i)$

We have

$$\tilde{p}\left(X_{vt} \mid \theta_i\right) = \exp \left\{ \mathbb{E}_Q \ln p^{\star}\left(X_{vt} \mid \theta_i\right) \right\} \tag{56}$$

and we already obtained $\ln p^{\star}\left(X_{vt} \mid \theta_i\right)$ in (44) to (48). Also, the forward algorithm can be written as follows:

$$\alpha_{vt}(i) = \tilde{p}\left(X_{vt} \mid \theta_i\right) \sum_{s=1}^{\infty} \alpha_{v,t-1}(s) \tilde{A}_{si} \tag{57}$$

which is the variational joint probability of $Z_{vt} = i$ and the sequence X_v until step t [27, 62]. Also, the backward algorithm is

$$\beta_{vt}(i) = \sum_{s=1}^{\infty} \tilde{A}_{is}\, \tilde{p}\left(X_{v,t+1} \mid \theta_s\right) \beta_{v,t+1}(s) \tag{58}$$

which is the variational probability of the sequence X_v after t given $Z_{vt} = i$ [27, 62]. Then, we can calculate $\gamma_{vt}(i)$ which is the marginal of Z_{vt} for $Q(Z_v)$:

$$\gamma_{vt}(i) = \frac{\alpha_{vt}(i)\beta_{vt}(i)}{\sum_s \alpha_{vt}(s)\beta_{vt}(s)} \tag{59}$$

For our SSD-HDP-HMM model, $\theta = \{\alpha, \beta, \tau\}$; therefore,

$$Q\left(\alpha_{ijl}\right) = \prod_{i=1}^{N}\prod_{j=1}^{K}\prod_{l=1}^{D} \mathcal{G}(\alpha_{ijl} \mid u^{\star}_{ijl}, v^{\star}_{ijl}) \tag{60}$$

$$Q\left(\beta_{ijl}\right) = \prod_{i=1}^{N}\prod_{j=1}^{K}\prod_{l=1}^{D} \mathcal{D}(\beta_{ijl} \mid \overrightarrow{h}^{\star}_{ijl}) \tag{61}$$

$$Q\left(\tau_{ij}\right) = \prod_{i=1}^{N}\prod_{j=1}^{K} \mathcal{G}(\tau_{ij} \mid q^{\star}_{ij}, s^{\star}_{ij}) \tag{62}$$

where the hyperparameters can be updated using Appendix 2.

3.2.2 Update $Q(W)$, $Q(\psi')$, and $Q(\varepsilon')$

In the previous section, we used the lower bound $\mathcal{L}(Q)$ to update $Q(Z)$ and $Q(\theta)$, but for updating $Q(W)$, $Q(\psi')$, and $Q(\varepsilon')$ the case is different. Since our approximation of the expected log of A in (54) does not produce tractable variational parameter updates for $Q(W)$, $Q(\psi')$, and $Q(\varepsilon')$, we employ a latent variable S_v and a variational distribution $Q(S_v \mid Z_v)$ for the vth observed sequence to lower bound $\mathcal{L}(Q)$ locally [27]. S_v interacts with Z_v and $W_{iji'}$ in the following way: the tuple $(Z_{v,t-1} = i, S_{vt} = j)$ denotes that the next state Z_{vt} may be discovered by selecting the jth stick of the ith DP and setting $Z_{vt} = i'$ if $W_{ij,i'} = 1$. In the paper [27], instead of using $Q(Z_{v,t-1}, Z_{vt})$ to obtain the state transition, they introduced

a triple $Q(Z_{vt}, S_{v,t+1}, W_{Z_{vt},S_{v,t+1}})$. Therefore, they came up with this local lower bound:

$$\mathbb{E}_Q 1\left(Z_{v,t-1}=i, Z_{vt}=i'\right) \ln \sum_j W_{ij,i'} \varepsilon_{ij} \tag{63}$$
$$\geq \mathbb{E}_Q \sum_j W_{ij,i'} 1\left(Z_{v,t-1}=i, S_{vt}=j\right) \ln \varepsilon_{ij}$$

Also, the joint variational distribution of S_v and Z_v is

$$Q\left(S_v, Z_v\right) = Q\left(S_v \mid Z_v\right) Q\left(Z_v\right) = Q\left(Z_v\right) \prod_t Q\left(S_{vt} \mid Z_v\right) \tag{64}$$

In the forward–backward algorithm, we already updated $Q(Z)$. The expectation value of $Q\left(Z_{v,t-1}=i, S_{vt}=j\right)$ can be updated by

$$Q\left(Z_{v,t-1}=i, S_{vt}=j\right) = \mathbb{E}_Q 1\left(Z_{v,t-1}=i, S_{vt}=j\right) \tag{65}$$
$$\equiv \xi_{vt}(i, j)$$

where $\xi_{vt}(i, j)$ is equivalent to γ_{ijt}^C in (31) for the vth observed sequence and can be obtained by

$$\xi_{vt}(i, j) \propto \alpha_{v,t-1}(i) \exp\left\{\mathbb{E}_Q \ln \varepsilon_{ij}\right\} \times \prod_{i'} \left[\exp\left\{\mathbb{E}_Q\left[\ln \theta_{i',X_{vt}}\right]\right\} \beta_{vt}\left(i'\right)\right]^{\varphi_{ij,i'}} \tag{66}$$

with

$$\mathbb{E}_Q \ln \varepsilon_{ij} = \mathbb{E}_Q \ln \varepsilon'_{ij} + \sum_s \mathbb{E}_Q \ln\left(1-\varepsilon'_{is}\right) \tag{67}$$

Here, α and β are the forward and backward algorithms, respectively. We recall that W_{ij} is the atom associated with the jth stick in the ith state.
Also, we have

$$Q\left(W_{ij,i'}=1\right) \equiv \varphi_{ij,i'} \tag{68}$$

where

$$\varphi_{ij,i'} \propto \exp\left\{\mathbb{E}_Q \ln \psi_{i'} + \sum_{v,t} \xi_{vt}(i, j) \mathbb{E}_Q \ln \theta_{i',X_{vt}}\right\} \tag{69}$$

with

$$\mathbb{E}_Q \ln \psi_{i'} = \mathbb{E}_Q \ln \psi'_{i'} + \sum_{s<i'} \mathbb{E}_Q \ln \left(1 - \psi'_s\right) \tag{70}$$

Using variational distributions $Q(\psi'_i) = \text{Beta}\,(c_i, d_i)$ and $Q(\varepsilon'_{ij}) = \text{Beta}\left(a_{ij}, b_{ij}\right)$, we have

$$\mathbb{E}_Q \ln \psi'_i = \Psi\,(c_i) - \Psi\,(c_i + d_i) \tag{71}$$

$$\mathbb{E}_Q \ln \left(1 - \psi'_i\right) = \Psi\,(d_i) - \Psi\,(c_i + d_i) \tag{72}$$

$$\mathbb{E}_Q \ln \varepsilon'_{ij} = \Psi\left(a_{ij}\right) - \Psi\left(a_{ij} + b_{ij}\right) \tag{73}$$

$$\mathbb{E}_Q \ln \left(1 - \varepsilon'_{ij}\right) = \Psi\left(b_{ij}\right) - \Psi\left(a_{ij} + b_{ij}\right) \tag{74}$$

where

$$a_{ij} = 1 + \sum_{v,t} \xi_{vt}(i, j) \tag{75}$$

$$b_{ij} = \lambda + \sum_{v,t} \sum_{j'>j} \xi_{vt}\left(i, j'\right) \tag{76}$$

$$c_i = 1 + \sum_{i',j} \mathbb{E}_Q W_{i'j,i} \tag{77}$$

$$d_i = \gamma + \sum_{i',j} \sum_{s>i} \mathbb{E}_Q W_{i'j,s} \tag{78}$$

Our suggested algorithm for variational learning of the SSD-HDP-HMM is described in Algorithm 2.

Algorithm 2 Variational learning of SSD-HDP-HMM

1. Initialize λ, γ, u_{ijl}, v_{ijl}, h_{ijl}, q_{ij} and s_{ij}.
2. **repeat**
3. Update $Q(Z_v)$ using the forward and backward algorithms using (57) to (59).
4. Update $Q(\theta)$ using (60) to (62) and appendix 2.
5. Update $Q\,(S_{vt} \mid Z_v)$ using (66).
6. Update $Q\,(W)$ using (68).
7. Update $Q\left(\psi'\right)$ and $Q\left(\varepsilon'\right)$ using (71) to (78).
8. **until convergence criterion is reached**.

4 Experimental Results

We tested our proposed models on two real-world applications: an activity recognition dataset [35] and a texture clustering [36]. Then, to assess how successful they are, we compare them to three other models: HDP-HMM, DMM-HMM (hidden Markov model with Dirichlet mixture model emissions), and GMM-HMM (hidden Markov model with Gaussian mixture model emissions). Using both real and anticipated labels of clusters, we use four metrics to assess our model's performance:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{79}$$

$$Precision = \frac{TP}{TP + FP} \tag{80}$$

$$Recall = \frac{TP}{TP + FN} \tag{81}$$

$$F1 - score = 2 * \frac{precision * recall}{precision + recall} \tag{82}$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

We present the average results after testing our model ten times with each dataset in order to deliver a more accurate result. Since our model is based on mixtures of shifted-scaled Dirichlet distributions, the data must be proportional as we discussed in Sect. 3. Therefore, we first normalize each record (row of the dataset) making it between 0 and 1 and then divide it by its summation of dimensions to make it proportional. Then, we use principal component analysis (PCA) to filter the most significant features of data [72].

4.1 Activity Recognition

The introduction of smartphones has significantly affected human lives. The existence of advanced features in these devices resulted in their continuous presence in our lives. Consequently, this presence provides a chance to keep track of our activities using a variety of sensors [73, 74]. Data generated by such devices could be used in many applications such as healthcare monitoring, life and fitness tracking, and transportation planning.

In our work, we applied an activity recognition (AR) dataset in which data were collected via accelerometer [75] and gyroscope sensors in a Samsung Galaxy S II. To gather the information, behavior of 30 participants aged between 19 and 48 [74] was monitored while wearing smartphones on the waist. This method of collecting data is one of the easiest ways because we do not need any additional equipment [74]. This is in contrast to other existing techniques for collecting AR data, which rely on special devices. Data are collected during six activities: walking, walking upstairs, walking downstairs, sitting, standing, and lying down. This dataset is randomly divided into two partitions: 70% of data are considered as a training dataset and 30% as a test dataset, and we have 10,299 data points in total. According to Fig. 1, nearly identical numbers of data points were received from all of the participants. Also, by looking at Fig. 2, we can see that the share of information associated with various activities is relatively similar. The lowest number of recorded activities belongs to the walking downstairs with 1406 instances. Therefore, to have a balanced dataset (consist of the same number of samples in all output classes), we randomly select the same number of data points from other categories before feeding our model (overall 8436 observations).

By looking at the probability density function (PDF) distribution in Fig. 3, we can see that motionless activities have a totally different distribution from moving activities. Besides, we can understand that the moving activity distributions are similar to each other, which is the same case for motionless activities. To have a better understanding of the data, we provide a 2D scatterplot of our data using different colors for each cluster in Fig. 4. As we can see in this plot, almost all of the features can be separated into different regions except standing and sitting, although there is considerable overlap with data points from other clusters especially near the boundaries. Therefore, we expect that dividing standing and sitting into two separate clusters may be a challenging task for a machine learning algorithm. We

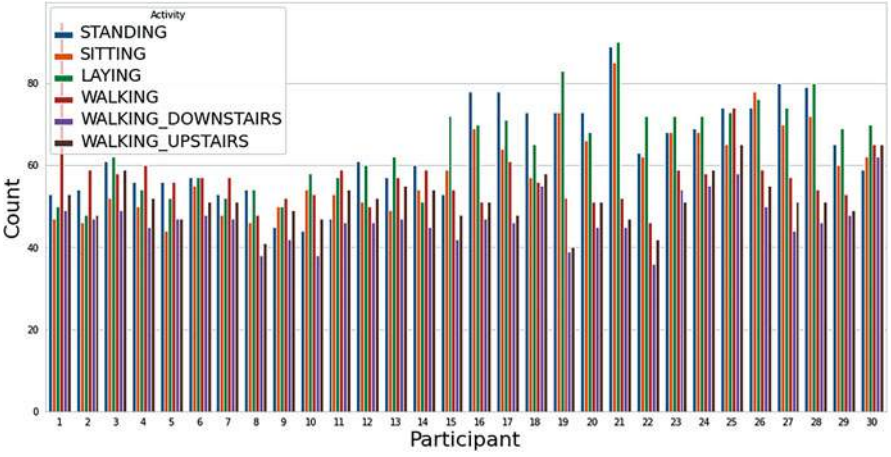


Fig. 1 Activity count per participant

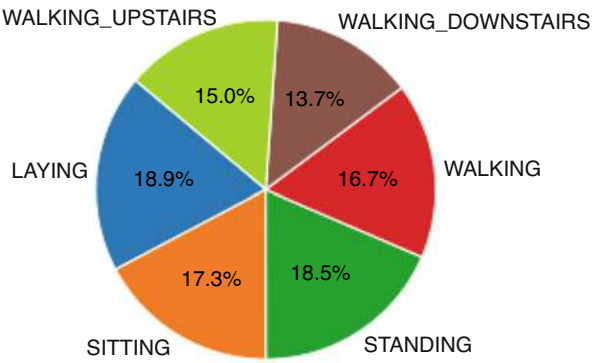


Fig. 2 Percentage of each category

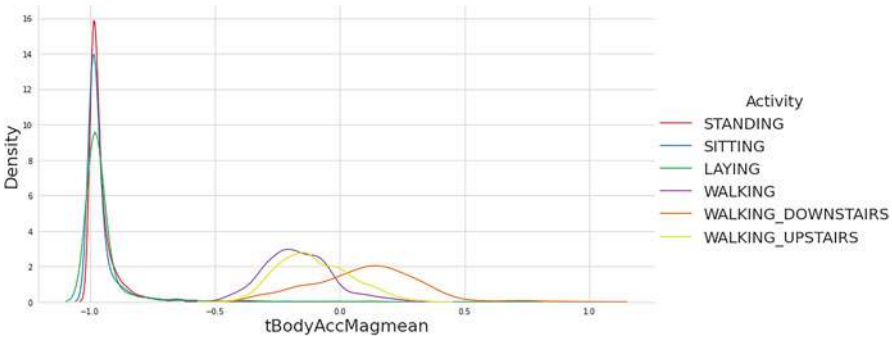


Fig. 3 Probability density function of the AR dataset

should mention that by using PCA, we reduced the number of features from 562 to 122 in order to filter the most important features. The result of testing our models is presented in Table 1. According to this table, the SSD-HDP-HMM model has an accuracy of 97.66%, which is a better result than the previously studied model, HDP-HMM. Also, the SSD-HMM model obtained higher accuracy than DMM-HMM and GMM-HMM models. As a result, using SSD as the parent distribution helped our models to fit the data better than either Dirichlet or Gaussian mixture model. In addition, by comparing the SSD-HDP-HMM model to the SSD-HMM model, we can find that hierarchical models are better in learning complicated data than non-hierarchical models.

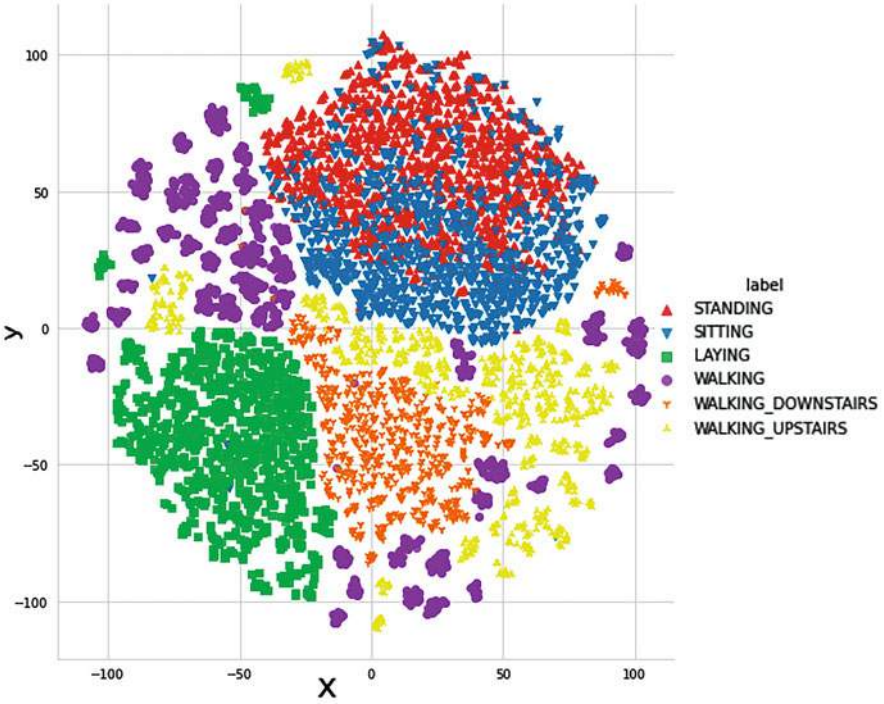


Fig. 4 2D scatterplot of 2 features of the AR dataset

Table 1 Activity recognition dataset results

Method	Accuracy	Precision	Recall	F1-score
SSD-HDP-HMM	97.66	97.66	97.63	97.64
HDP-HMM	95.29	95.38	95.36	95.36
SSD-HMM	93.88	94.03	94.06	94
DMM-HMM	86.65	86.12	86.36	86.11
GMM-HMM	93.59	93.11	93.13	93.11

4.2 Texture Clustering

As the second application, we chose to use a challenging dataset called the University of Illinois Urbana Champaign (UIUC) texture dataset. This dataset has 25 classes, each with 40 images of size 480×640. Figure 5 shows some sample images of from UIUC dataset. The diversity of 2D and 3D transformations, as well as lighting fluctuations in this dataset, made it a difficult application for machine learning algorithms. Therefore, we applied VGG16 to extract features of images. VGG16 is a popular strong deep learning model which has already shown its capabilities in feature extraction. We also acknowledge that this model is not designed for feature selection and we used an arbitrary intermediate layer to extract

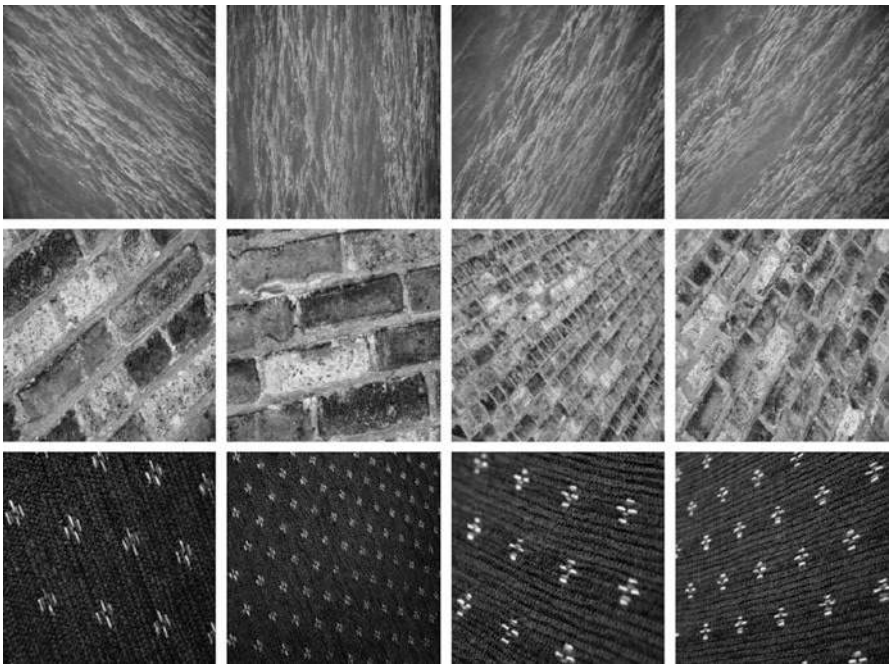


Fig. 5 Samples of UIUC texture dataset

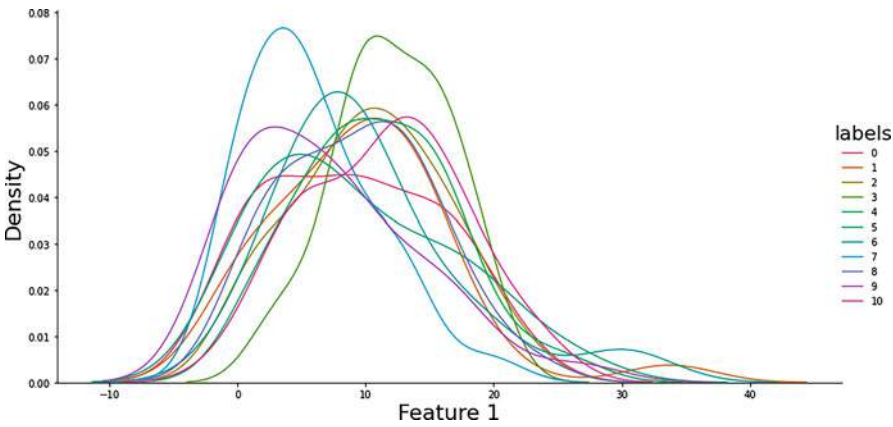


Fig. 6 Probability density function of UIUC texture dataset

features of images. Then using PCA, we reduced the number of features extracted by VGG16 from 4096 to 1251. Figure 6 shows the PDF distribution of one of the most important features. As we can see in this figure, the distribution of classes is quite similar to one another, and we verified that this is true for other features as well. Therefore, this makes the clustering task hard. This conclusion may also be

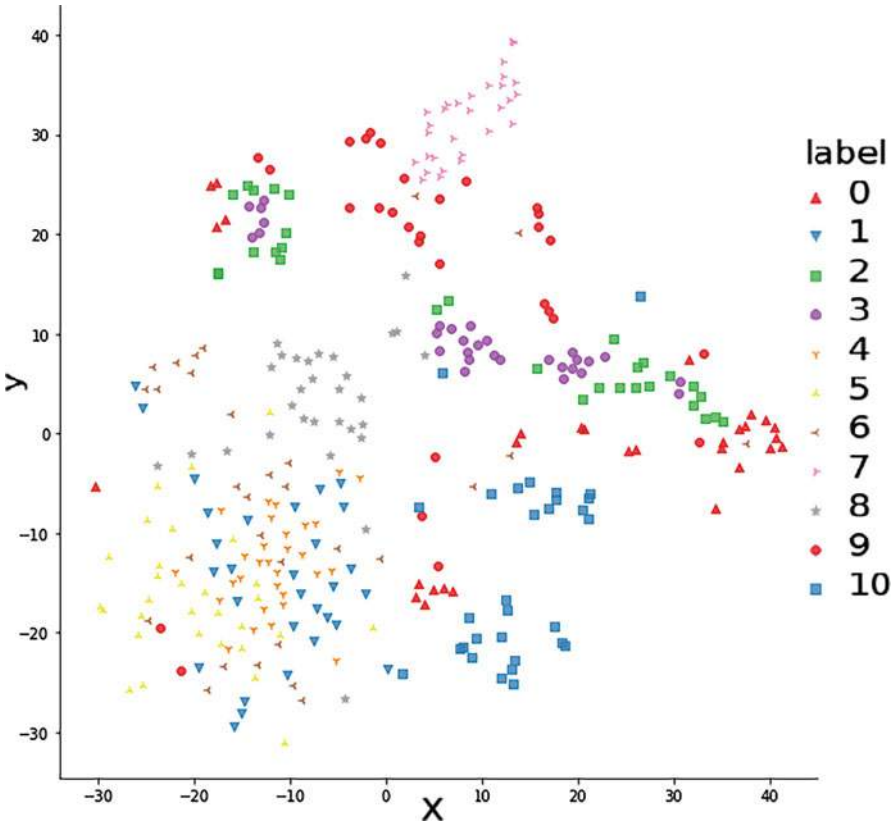


Fig. 7 2D scatterplot of 2 features of the UIUC dataset

obtained by looking at Fig. 7, which is a 2D scatterplot. In this figure, some of the clusters are in totally separated regions and most of them have overlap with one another. As a result, we chose to test our model with the almost-separated classes 3, 4, and 7 (see Fig. 5). As we mentioned before, each of the classes has 40 images, so our dataset is already balanced. Table 2 displays the results of our model testing. Our proposed hierarchical model, SSD-HDP-HMM, has the greatest accuracy of the examined models, with 81.8 percent. This demonstrates that, owing to the SSD distribution, this model fits the data better than the classical HDP-HMM model. Furthermore, the SSD-HMM model outperforms the basic Dirichlet and Gaussian HMM mixture models.

Table 2 UIUC texture dataset results

Method	Accuracy	Precision	Recall	F1-score
SSD-HDP-HMM	81.8	80.7	69.7	80.6
HDP-HMM	80.2	77.2	75.3	75.2
SSD-HMM	77.2	81.1	80.1	73
DMM-HMM	75.7	71.9	69.7	69.7
GMM-HMM	56	53.8	54.1	48

5 Conclusion

In this work, we introduced two novel models, the "SSD-HMM" and the "SSD-HDP-HMM," respectively. Also, we derived a variational learning algorithm for each of them. These learning methods have various advantages that help overcome the disadvantages of other learning algorithms, for example, tractable learning algorithms, reliable approximations, and ensured convergence. Also, the flexibility of the SSD distribution in fitting data particularly in proportional cases was the major incentive to use it for emission probabilities in our research. In the experimental section, we demonstrated the benefits of using mixtures of the SSD distributions instead of Gaussian and Dirichlet mixture models in the HMM framework. We can identify the proper number of clusters, which is the number of hidden states throughout the learning process, using a nonparametric Bayesian model. Finally, we got promising results when we applied our proposed models to two challenging real-world applications: activity recognition and texture clustering. We may concentrate on feature selection and integrate it into our models in the future works.

Acknowledgments The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC).

Appendices

Appendix 1

$$u^{\star}_{ijl} = u_{ijl} + \sum_{v=1}^{V} \mathbb{E}_Q Z_{vij} \times \overline{\alpha}_{ijl} \left[\Psi(\sum_{s=1}^{D} \overline{\alpha}_{ijs}) - \Psi\left(\overline{\alpha}_{ijl}\right) + \sum_{s\neq d}^{D} \Psi'\Big(\sum_{s=1}^{D} \overline{\alpha}_{ijs}\Big) \right.$$

$$\left. \times \overline{\alpha}_{ijs}(\mathbb{E}_Q \ln \alpha_{ijs} - \ln \overline{\alpha}_{ijs}) \right] \tag{83}$$

$$v^{\star}_{ijl} = v_{ijl} - \sum_{v=1}^{V} \mathbb{E}_Q Z_{vij} \times \Big[\frac{1}{\tau_{ij}} \ln \frac{\beta_{ijl}}{x_{vl}} + \ln \Big(\sum_{s=1}^{D} \Big(\frac{x_{vs}}{\beta_{ijs}} \Big) \frac{1}{\tau_{ij}} \Big) \Big] \tag{84}$$

$$h^{\star}_{ijl} = h_{ijl} + \sum_{v=1}^{V} \mathbb{E}_Q Z_{vij} \times \Big[\frac{-\bar{\alpha}_{ijl}}{\bar{\tau_{ij}}} + \frac{\bar{\alpha_{ijl}}}{\bar{\tau_{ij}}} \times \Big(\frac{x_{vl}}{\bar{\beta_{ijl}}} \Big)^{\frac{1}{\bar{\tau_{ij}}}} \times \frac{1}{\sum_{s=1}^{D} \Big(\frac{x_{vs}}{\bar{\beta_{ijs}}} \Big)^{\bar{\tau_{ij}^{-1}}}} \Big] \tag{85}$$

$$q^{\star}_{ij} = q_{ij} + \sum_{v=1}^{V} \mathbb{E}_Q Z_{vij} \times \Big[1 - D + \frac{(\alpha_{ij}+)}{\tau_{ij}} \frac{\sum_{l=1}^{D} \Big(\frac{x_{vl}}{\beta_{ijl}} \Big)^{\tau_{ij}^{-1}} \ln \Big(\frac{x_{vl}}{\beta_{ijl}} \Big)}{\sum_{l=1}^{D} \Big(\frac{x_{vl}}{\beta_{ijl}} \Big)^{\tau_{ij}^{-1}}} \Big] \tag{86}$$

$$s^{\star}_{ij} = s_{ij} - \sum_{v=1}^{V} \mathbb{E}_Q Z_{vij} \times \Big[\sum_{l=1}^{D} \frac{\alpha_{ijl}}{{\tau_{ij}}^2} \ln \Big(\frac{x_{vl}}{\beta_{ijl}} \Big) \Big] \tag{87}$$

Also, the expected values are given by

$$\bar{\alpha}_{ijl} = \mathbb{E}_Q \alpha_{ijl} = \frac{u^{\star}_{ijl}}{v^{\star}_{ijl}}, \quad \overline{\beta}_{ijl} = \mathbb{E}_Q \beta_{ijl} = \frac{h^{\star}_{ijl}}{\sum_{l=1}^{D} h^{\star}_{ijl}}$$

$$\bar{\tau}_{ij} = \mathbb{E}_Q \tau_{ijl} = \frac{q^{\star}_{ij}}{s^{\star}_{ij}}, \quad \mathbb{E}_Q Z_{vij} = \sum_{t=1}^{T} \gamma^{C}_{vijt} = p(s = i, m = j \mid X)$$

$$\mathbb{E}_Q \ln \alpha_{ijl} = \Psi(u^{\star}_{ijl}) - \ln v^{\star}_{ijl} \tag{88}$$

Appendix 2

$$u^{\star}_{ijl} = u_{ijl} + \sum_{i'=1}^{N} \varphi_{ij,i'} \sum_{t=1}^{T} \sum_{v=1}^{V} \xi_{vt}(i, j) \times \overline{\alpha}_{ijl} \times \tag{89}$$

$$\Big[\psi\Big(\sum_{s=1}^{D} \overline{\alpha}_{ijs} \Big) - \psi\big(\overline{\alpha}_{ijl} \big) + \sum_{s \neq d}^{D} \psi'\Big(\sum_{s=1}^{D} \overline{\alpha}_{ijs} \Big) \times \overline{\alpha}_{ijs} (\mathbb{E}_Q \ln \alpha_{ijs} - \ln \overline{\alpha}_{ijs}) \Big]$$

$$v^{\star}_{ijl} = v_{ijl} - \sum_{i'=1}^{N} \varphi_{ij,i'} \sum_{t=1}^{T} \sum_{v=1}^{V} \xi_{vt}(i, j) \Big[\frac{1}{\tau_{ij}} \ln \frac{\beta_{ijl}}{X_{tl}} + \ln \Big(\sum_{s=1}^{D} (\frac{X_{ts}}{\beta_{ijs}})^{\frac{1}{\tau_{ij}}} \Big) \Big] \tag{90}$$

$$h^{\star}_{ijl} = h_{ijl} + \sum_{i'=1}^{N} \varphi_{ij,i'} \sum_{t=1}^{T} \sum_{v=1}^{V} \xi_{vt}(i, j) \Big[\frac{-\bar{\alpha}_{ijl}}{\bar{\tau_{ij}}} + \frac{\bar{\alpha_{ijl}}}{\bar{\tau_{ij}}} \times \Big(\frac{X_{tl}}{\bar{\beta_{ijl}}} \Big)^{\frac{1}{\bar{\tau_{ij}}}} \times \frac{1}{\sum_{s=1}^{D} \Big(\frac{X_{ts}}{\bar{\beta_{ijs}}} \Big)^{\bar{\tau_{ij}^{-1}}}} \Big] \tag{91}$$

$$q_{ij}^{\star} = q_{ij} + \sum_{i'=1}^{N} \varphi_{ij,i'} \sum_{t=1}^{T} \sum_{v=1}^{V} \xi_{vt}(i,j) \Big[1 - D + \frac{(\alpha_{ij}+)}{\tau_{ij}} \frac{\sum_{l=1}^{D} \left(\frac{X_{tl}}{\beta_{ijl}} \right)^{\tau_{ij}^{-1}} \ln \left(\frac{X_{tl}}{\beta_{ijl}} \right)}{\sum_{l=1}^{D} \left(\frac{X_{tl}}{\beta_{ijl}} \right)^{\tau_{ij}^{-1}}} \Big] \tag{92}$$

$$s_{ij}^{\star} = s_{ij} - \sum_{i'=1}^{N} \varphi_{ij,i'} \sum_{t=1}^{T} \sum_{v=1}^{V} \xi_{vt}(i,j) \Big[\sum_{l=1}^{D} \frac{\alpha_{ijl}}{{\tau_{ij}}^2} \ln \left(\frac{X_{tl}}{\beta_{ijl}} \right) \Big] \tag{93}$$

References

1. L. Rabiner, B. Juang, An introduction to hidden Markov models. IEEE ASSP Mag. **3**(1), 4–16 (1986)
2. T. Fuse, K. Kamiya, Statistical anomaly detection in human dynamics monitoring using a hierarchical dirichlet process hidden Markov model. IEEE Trans. Intell. Transp. Syst. **18**(11), 3083–3092 (2017)
3. M. Rahul, N. Kohli, R. Agarwal, S. Mishra, Facial expression recognition using geometric features and modified hidden Markov model. Int. J. Grid Util. Comput. **10**(5), 488–496 (2019)
4. M.K. Mustafa, T. Allen, K. Appiah, A comparative review of dynamic neural networks and hidden Markov model methods for mobile on-device speech recognition. Neural Comput. Appl. **31**(2), 891–899 (2019)
5. W. Wang, D. Zhu, T. Alkhouli, Z. Gan, H. Ney, Neural hidden Markov model for machine translation, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (2018), pp. 377–382
6. X. Zhang, Y. Li, S. Wang, B. Fang, S. Yu. Philip, Enhancing stock market prediction with extended coupled hidden Markov model over multi-sourced data. Knowl. Inf. Syst. **61**(2), 1071–1090 (2019)
7. G. Manogaran, V. Vijayakumar, R. Varatharajan, P.M. Kumar, R. Sundarasekar, C.-H. Hsu, Machine learning based big data processing framework for cancer diagnosis using hidden Markov model and gm clustering. Wirel. Pers. Commun. **102**(3), 2099–2116 (2018)
8. Md. Z. Uddin, Human activity recognition using segmented body part and body joint features with hidden Markov models. Multimedia Tools Appl. **76**(11), 13585–13614 (2017)
9. K.M. Sagayam, D.J. Hemanth, A probabilistic model for state sequence analysis in hidden Markov model for hand gesture recognition. Computational Intelligence **35**(1), 59–81 (2019)
10. L.E. Baum, T. Petrie, Statistical inference for probabilistic functions of finite state Markov chains. Ann. Math. Stat. **37**(6), 1554–1563 (1966)
11. C.M. Bishop, Pattern recognition. Machine Learning **128**(9), (2006)
12. E. Epaillard, N. Bouguila, Variational bayesian learning of generalized dirichlet-based hidden Markov models applied to unusual events detection. IEEE Trans. Neural Netw. Learn. Syst. **30**(4), 1034–1047 (2018)
13. E. Epaillard, N. Bouguila, Hidden Markov models based on generalized dirichlet mixtures for proportional data modeling, in *Artificial Neural Networks in Pattern Recognition - 6th IAPR TC 3 International Workshop, ANNPR 2014, Montreal, QC, Canada, October 6–8, 2014. Proceedings*, vol. 8774 of *Lecture Notes in Computer Science*, ed. by N.El. Gayar, F. Schwenker, C. Suen (Springer, 2014), pp. 71–82

14. E. Epaillard, N. Bouguila, D. Ziou, Classifying textures with only 10 visual-words using hidden Markov models with dirichlet mixtures, in *Adaptive and Intelligent Systems - Third International Conference, ICAIS 2014, Bournemouth, UK, September 8–10, 2014. Proceedings*, vol. 8779 of *Lecture Notes in Computer Science*, ed. by A. Bouchachia (Springer, 2014), pp. 20–28
15. E. Epaillard, N. Bouguila, Hybrid hidden Markov model for mixed continuous/continuous and discrete/continuous data modeling, in *17th IEEE International Workshop on Multimedia Signal Processing, MMSP 2015, Xiamen, China, October 19–21, 2015* (IEEE, 2015), pp. 1–6
16. L. Tao, E. Elhamifar, S. Khudanpur, G.D. Hager, R. Vidal, Sparse hidden Markov models for surgical gesture classification and skill evaluation, in *International Conference on Information Processing in Computer-Assisted Interventions* (Springer, 2012), pp. 167–177
17. W. Fan, R. Wang, N. Bouguila, Simultaneous positive sequential vectors modeling and unsupervised feature selection via continuous hidden Markov models. Pattern Recognition **119**, 108073 (2021)
18. E. Epaillard, N. Bouguila, Data-free metrics for dirichlet and generalized dirichlet mixture-based hmms—a practical study. Pattern Recognition **85**, 207–219 (2019)
19. R. Nasfi, M. Amayri, N. Bouguila, A novel approach for modeling positive vectors with inverted dirichlet-based hidden Markov models. Knowl.-Based Syst. **192**, 105335 (2020)
20. G.S. Monti, G. Mateu i Figueras, V. Pawlowsky-Glahn, J.J. Egozcue, et al., The shifted-scaled dirichlet distribution in the simplex (2011)
21. Z. Ma, Y. Lai, W. Bastiaan Kleijn, Y.-Z. Song, L. Wang, J. Guo, Variational bayesian learning for dirichlet process mixture of inverted dirichlet distributions in non-gaussian image feature modeling. IEEE Trans. Neural Netw. Learn. Syst. **30**(2), 449–463 (2019)
22. J. Chen, Z. Gong, W. Liu, A dirichlet process biterm-based mixture model for short text stream clustering. Applied Intelligence **50**(5), 1609–1619 (2020)
23. S. Frühwirth-Schnatter, G. Malsiner-Walli, From here to infinity: sparse finite versus dirichlet process mixtures in model-based clustering. Adv. Data Anal. Classif. **13**(1), 33–64 (2019)
24. K. Meguelati, B. Fontez, N. Hilgert, F. Masseglia, High dimensional data clustering by means of distributed dirichlet process mixture models, in *2019 IEEE International Conference on Big Data (Big Data)* (IEEE, 2019), pp. 890–899
25. M. Beraha, A. Guglielmi, F.A. Quintana, The semi-hierarchical dirichlet process and its application to clustering homogeneous distributions. Preprint (2020). arXiv:2005.10287
26. S. Fine, Y. Singer, N. Tishby, The hierarchical hidden Markov model: Analysis and applications. Machine Learning **32**(1), 41–62 (1998)
27. A. Zhang, S. Gultekin, J. Paisley, Stochastic variational inference for the hdp-hmm, in *Artificial Intelligence and Statistics* (PMLR, 2016), pp. 800–808
28. E.B. Fox, E.B. Sudderth, M.I. Jordan, A.S. Willsky, An hdp-hmm for systems with state persistence, in *Proceedings of the 25th International Conference on Machine Learning* (2008), pp. 312–319
29. E.B. Fox, E.B. Sudderth, M.I. Jordan, A.S. Willsky, A sticky hdp-hmm with application to speaker diarization. Ann. Appl. Stat., 1020–1056 (2011)
30. W. Fan, H. Sallay, N. Bouguila, Online learning of hierarchical pitman-yor process mixture of generalized dirichlet distributions with feature selection. IEEE Trans. Neural Netw. Learn. Syst. **28**(9), 2048–2061 (2017)
31. M.D. Hoffman, Learning deep latent gaussian models with Markov chain monte carlo, in *International Conference on Machine Learning* (PMLR, 2017), pp. 1510–1519
32. M.-A. Sato, Online model selection based on the variational bayes. Neural Computation **13**(7), 1649–1681 (2001)
33. D.M. Blei, A. Kucukelbir, J.D. McAuliffe, Variational inference: A review for statisticians. J. Am. Stat. Assoc. **112**(518), 859–877 (2017)
34. W. Fan, N. Bouguila, D. Ziou, Variational learning for finite dirichlet mixture models and applications. IEEE Trans. Neural Netw. Learn. Syst. **23**(5), 762–774 (2012)

35. D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz, et al., A public domain dataset for human activity recognition using smartphones, in *Esann*, vol. 3, p. 3 (2013)
36. http://www-cvr.ai.uiuc.edu/
37. G. Ogbuabor, R. La, Human activity recognition for healthcare using smartphones, in *Proceedings of the 2018 10th International Conference on Machine Learning and Computing* (2018), pp. 41–46
38. A. Roitberg, A. Perzylo, N. Somani, M. Giuliani, M. Rickert, A. Knoll, Human activity recognition in the context of industrial human-robot interaction, in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific* (IEEE, 2014), pp. 1–10
39. T. De Pessemier, L. Martens, Heart rate monitoring, activity recognition, and recommendation for e-coaching. Multimedia Tools Appl. **77**(18), 23317–23334 (2018)
40. M. Babiker, O.O. Khalifa, K.K. Htike, A. Hassan, M. Zaharadeen, Automated daily human activity recognition for video surveillance using neural network, in *2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)* (IEEE, 2017), pp. 1–5
41. M. Vrigkas, C. Nikou, I.A. Kakadiaris, A review of human activity recognition methods. Front. Robot. AI **2**, 28 (2015)
42. R. Peyret, A. Bouridane, F. Khelifi, M.A. Tahir, S. Al-Maadeed, Automatic classification of colorectal and prostatic histologic tumor images using multiscale multispectral local binary pattern texture features and stacked generalization. Neurocomputing **275**, 83–93 (2018)
43. S. Fekriershad, F. Tajeripour, Color texture classification based on proposed impulse-noise resistant color local binary patterns and significant points selection algorithm. Sensor Review (2017)
44. N. Bouguila, D. Ziou, Unsupervised learning of a finite discrete mixture model based on the multinomial dirichlet distribution: Application to texture modeling, in *Pattern Recognition in Information Systems, Proceedings of the 4th International Workshop on Pattern Recognition in Information Systems, PRIS 2004, In Conjunction with ICEIS 2004, Porto, Portugal, April 2004*, ed. by A.L.N. Fred (INSTICC Press, 2004), pp. 118–127
45. N. Bouguila, Infinite liouville mixture models with application to text and texture categorization. Pattern Recognit. Lett. **33**(2), 103–110 (2012)
46. N. Bouguila, T. Elguebaly, A bayesian approach for texture images classification and retrieval, in *2011 International Conference on Multimedia Computing and Systems* (2011), pp. 1–6
47. X. Hu, Y. Huang, X. Gao, L. Luo, Q. Duan, Squirrel-cage local binary pattern and its application in video anomaly detection. IEEE Trans. Inf. Forensics Secur. **14**(4), 1007–1022 (2018)
48. Y. Xu, S. Huang, H. Ji, C. Fermuller, Combining powerful local and global statistics for texture description, in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2009), pp. 573–580
49. Y. Xu, X. Yang, H. Ling, H. Ji, A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid, in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE, 2010), pp. 161–168
50. D. Zhao, D. Zhu, J. Lu, Y. Luo, G. Zhang, Synthetic medical images using f&bgan for improved lung nodules classification by multi-scale vgg16. Symmetry **10**(10), 519 (2018)
51. D.M.S. Arsa, A.A.N.H. Susila, Vgg16 in batik classification based on random forest, in *2019 International Conference on Information Management and Technology (ICIMTech)*, vol. 1 (IEEE, 2019), pp. 295–299
52. D.I. Swasono, H. Tjandrasa, C. Fathicah, Classification of tobacco leaf pests using vgg16 transfer learning, in *2019 12th International Conference on Information & Communication Technology and System (ICTS)* (IEEE, 2019), pp. 176–181
53. Z. Ghahramani, M.I. Jordan, Factorial hidden Markov models. Machine Learning **29**(2), 245–273 (1997)

54. M. Bicego, U. Castellani, V. Murino, A hidden Markov model approach for appearance-based 3D object recognition. Pattern Recognit. Lett. **26**(16), 2588–2599 (2005)
55. E.L. Andrade, S. Blunsden, R.B. Fisher, Hidden Markov models for optical flow analysis in crowds, in *18th international conference on pattern recognition (ICPR'06)*, vol. 1 (IEEE, 2006), pp. 460–463
56. B.S. Oboh, N. Bouguila, Unsupervised learning of finite mixtures using scaled dirichlet distribution and its application to software modules categorization, in *IEEE International Conference on Industrial Technology, ICIT 2017, Toronto, ON, Canada, March 22–25, 2017* (IEEE, 2017), pp. 1085–1090
57. N. Zamzami, R. Alsuroji, O. Eromonsele, N. Bouguila, Proportional data modeling via selection and estimation of a finite mixture of scaled dirichlet distributions. Comput. Intell. **36**(2), 459–485 (2020)
58. Z. Ma, A. Leijon, Super-dirichlet mixture models using differential line spectral frequencies for text-independent speaker identification, in *Twelfth Annual Conference of the International Speech Communication Association* (2011)
59. N. Manouchehri, O. Dalhoumi, M. Amayri, N. Bouguila, Variational learning of a shifted scaled dirichlet model with component splitting approach, in *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)* (IEEE, 2020), pp. 75–78
60. N. Manouchehri, H. Nguyen, P. Koochemeshkian, N. Bouguila, W. Fan, Online variational learning of dirichlet process mixtures of scaled dirichlet distributions. Inf. Syst. Front. **22**(5), 1085–1093 (2020)
61. H. Nguyen, M. Rahmanpour, N. Manouchehri, K. Maanicshah, M. Amayri, N. Bouguila, A statistical approach for unsupervised occupancy detection and estimation in smart buildings, in *2019 IEEE International Smart Cities Conference (ISC2)* (IEEE, 2019), pp. 414–419
62. M.J. Beal, *Variational Algorithms for Approximate Bayesian Inference* (University of London, University College London, UK, 2003)
63. Z. Arjmandiasl, N. Manouchehri, N. Bouguila, J. Bentahar, Variational learning of finite shifted scaled dirichlet mixture models, in *Learning Control* (Elsevier, 2021), pp. 175–204
64. S. Ji, B. Krishnapuram, L. Carin, Variational bayes for continuous hidden Markov models and its application to active learning. IEEE Trans. Pattern Anal. Mach. Intell. **28**(4), 522–532 (2006)
65. S.P. Chatzis, D.I. Kosmopoulos, A variational bayesian methodology for hidden Markov models utilizing student's-t mixtures. Pattern Recognition **44**(2), 295–306 (2011)
66. L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
67. D. Li, S. Zamani, J. Zhang, P. Li, Integration of knowledge graph embedding into topic modeling with hierarchical dirichlet process, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (2019), pp. 940–950
68. Y.W. Teh, M.I. Jordan, Hierarchical bayesian nonparametric models with applications. Bayesian Nonparametrics **1**, 158–207 (2010)
69. W. Fan, H. Sallay, N. Bouguila, S. Bourouis, Variational learning of hierarchical infinite generalized dirichlet mixture models and applications. Soft Computing **20**(3), 979–990 (2016)
70. H. Zhang, S. Huating, X. Wu, Topic model for graph mining based on hierarchical dirichlet process. Stat. Theory Relat. Fields **4**(1), 66–77 (2020)
71. C. Wang, J. Paisley, D. Blei, Online variational inference for the hierarchical dirichlet process, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings* (2011), pp. 752–760
72. A. Malhi, R.X. Gao, Pca-based feature selection scheme for machine defect classification. IEEE Trans. Instrum. Meas. **53**(6), 1517–1525 (2004)
73. A. Ignatov, Real-time human activity recognition from accelerometer data using convolutional neural networks. Appl. Soft Comput. **62**, 915–922 (2018)

74. D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz, Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine, in *International Workshop on Ambient Assisted Living* (Springer, 2012), pp. 216–223
75. F.R. Allen, E. Ambikairajah, N.H. Lovell, B.G. Celler, Classification of a known sequence of motions and postures from accelerometry data using adapted gaussian mixture models. Physiological Measurement **27**(10), 935 (2006)

Index

N. Bouguila et al. (eds.), *Hidden Markov Models and Applications*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-99142-5