Pattern Analysis and Applications (2024) 27:117
https://doi.org/10.1007/s10044-024-01341-5

THEORETICAL ADVANCES

Hidden Markov models with multivariate bounded asymmetric student's t-mixture model emissions

Ons Bouarada[1] · Muhammad Azam[2] · Manar Amayri[1] · Nizar Bouguila[1]

Received: 25 November 2022 / Accepted: 9 September 2024

Abstract
Hidden Markov models (HMMs) are popular methods for continuous sequential data modeling and classification tasks. In such applications, the observation emission densities of the HMM hidden states are generally continuous, can vary from one model to the other, and are typically modeled by elliptically contoured distributions, namely Gaussians or Student's t-distributions. In this context, this paper proposes a novel HMM with Bounded Asymmetric Student's t-Mixture Model (BASMM) emissions. Our new BASMMHMM is introduced in the light of the added robustness guaranteed by the BASMM in comparison to other popular emission distributions such as the Gaussian Mixture Model (GMM). In fact, GMMs generally have a limited performance with outliers in the data sets (observations) that the HMM is fitted to. Also, GMMs cannot sufficiently model skewed populations, which are typical in many fields, such as financial or signal processing-related data sets. An excellent alternative to solve this problem is found in Student's t-mixture models. They have similar behaviour and shape to GMMs, but with heavier tails. This allows to have more tolerance towards data sets that span extensive ranges and include outliers. Asymmetry and bounded support are also important features that can further extend the model's flexibility and fit the imperfections of real-world data. This leads us to explore the effectiveness of the BASMM as an observation emission distribution in HMMs, hence the proposed BASMMHMM. We will also demonstrate the improved robustness of our model by presenting the results of three different experiments: occupancy estimation, stock price prediction, and human activity recognition.

Keywords Hidden Markov models · Multivariate student's t-mixture · Bounded asymmetric student's t · Prediction · Recognition

1 Introduction

HMMs [1] are a simple, yet powerful, tool to represent and predict sequential events [2] and are widely used in many types of data-driven tasks. The concept of HMMs is primarily based on Markov Chains [3, 4] (proposed by Andrey Markov in the early 20th century) but was formally developed later in many works. The key idea of HMMs is that a latent variable or state variable evolves according to a discrete, first-order Markov process. More specifically, the modeled process/data is a sequence of states or values that are unknown (hidden), where each hidden state depends on the past hidden state in the sequence. This Markov Chain of hidden states is associated with an equal sequence of known values (observations). Every hidden state emits an observation that follows a well-defined probability distribution in the space of observations, and each observation is conditionally independent of every other observation, given the value of its associated hidden state. By their structure, HMMs are generally able to solve a variety of tasks mainly with three main functionalities [5, 6]: evaluation, decoding (inference), and learning. The evaluation is the computation of the probability of an observation sequence given an HMM. Decoding

✉ Nizar Bouguila
nizar.bouguila@concordia.ca

Ons Bouarada
o_bouara@live.concordia.ca

Muhammad Azam
mu_azam@encs.concordia.ca

Manar Amayri
manar.amayri@concordia.ca

[1] Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

[2] School of Computer Science and Technology, Algoma University, Sault Ste. Marie, Ontario, Canada

Published online: 18 September 2024

is the task of inferring the most probable sequence of hidden states given a defined HMM and a sequence of known observations. As for learning, it is the search for the best parameters of the HMM (learning the HMM) given an observation sequence and the set of possible hidden states in the model.

By their definition, HMMs are an excellent choice to tackle data tasks that involve non-observable sequential values, as their structure allows inferring these latent values from the observable signals or even predicting their future trends. This high flexibility makes HMMs a strong candidate to deal with a variety of applications such as genetics and biomedical engineering [7, 8], climate modeling [9], signal processing [10], stock market prediction [11], speech [12], video recognition [13], and information retrieval systems [14] to name a few.

The observation emission, i.e., the formulation of the conditional dependence between the observations and the hidden states of the HMM is generally a deciding factor for the behaviour of the model, and is also our area of interest in this paper. For continuous data, the observation emission probability distributions associated with the hidden states often have a specific form from a parametric class such as Gaussian, Gamma, or Poisson. In this regard, multiple works have further explored the emission distributions and introduced the mixture models as an alternative [15]. This has led to some very useful variants of HMMs, perhaps the most popular one being the Gaussian mixture model HMM (GMMHMM). This prevalence of the GMMHMMs stems from the convenience of the GMM, as it provides a natural way to cluster the data and has relatively simple implementations and parameters. However, Gaussian-based distributions do not account for multiple natural characteristics of real-world data sets, including the presence of outliers [16], their asymmetry, and their specific location in space. Ergo, HMMs with Gaussian-based emissions can be limited when dealing with outlier-heavy, or significantly asymmetric data, which is often the case. Some of these issues have been tackled in [17] by introducing a bounded asymmetric Gaussian mixture [18] as an emission distribution for the HMM, but the low outlier tolerance of the Gaussian distribution remains a problem.

On this matter, the Student's t-distribution [19] is an excellent alternative to the Gaussian when fitting skewed or heavy-tailed populations, thus, the multivariate finite Student's t-Mixture Model (SMM) [20] can provide a more robust fit than the GMM in the presence of significant proportions of outliers in the data. Multiple articles have explored the potential of the HMMs with SMM emissions as in [21–23], but the idea of customizing this model within the HMM to better fit the real-world data has not been examined yet. In fact, while SMMs are an excellent solution for handling outliers, they assume, by their mathematical definition, that the examined data is symmetric and spans over an unbounded range, which is not a realistic depiction of most data sets.

This motivated us to introduce BASMMHMM, a HMM with Bounded Asymmetric Mixture Model (BASMM) emissions. This model is an amelioration of the drawbacks observed in the previously proposed HMMs, as the emissions' distributions will not only fit observed data outliers (with heavy distribution tails), but also tolerate the natural imperfections of the data (with asymmetry) and take into account the fact that the data usually spans only finite regions of its space. We train our BASMMHMM using the Baum–Welch Expectation Maximization (EM) algorithm, and we apply it on a selection of popular real-world tasks, where the HMMs are a very efficient recourse: occupancy estimation [24], stock price prediction and human activity recognition [25].

This paper is laid out as follows: in the first section, we introduce the general scope and the motivations for this work. In the second section, we present the emission distribution of our proposed HMM, which is the multivariate Bounded Asymmetric Student's-t Mixture Model (BASMM). In the third section, we review the necessary mathematical definitions and present the HMM with BASMM emissions. The fourth section features experiments using the proposed model as well as their results. Following the experiments' results, we also establish a comparison between our proposed model and the other types of HMMs a variety of emissions. Finally, the fifth section concludes this work and discusses eventual paths of improvement.

The mathematical variables' notations used for the rest of the paper are detailed in Table 1.

2 Multivariate bounded asymmetric student's-t Mixture Model

The BASMM [26] is a generalized format of the SMM where the specific location of the modeled data in the space (bounded support) and its natural asymmetry are taken into consideration. Being based on the multivariate Student's t-distribution, the BASMM, and SMM are more robust than other popular mixture models like the GMM. In fact, unlike the Gaussian density function, the Student's t-density function has an additional parameter -the degrees of freedom ν- which is a robustness tuning parameter. As a result, the t-distribution provides a heavy-tailed alternative to the Gaussian distribution (see Fig. 1) for potential outliers in the data and therefore, the SMM can produce a clustering algorithm that is more outlier-tolerant than the GMM.

Table 1 BASMMHMM notations

Notation	Definition
BASMMHMM	Bounded asymmetric student's-t mixture model hidden Markov model
$\mathcal{M} = \{\lambda_i^{t_0}, \lambda_{i,j}, s_j, y_t\}_{i,j,t=1}^{N,N,L}$	Full definition of a BASMMHMM $\mathcal{M}$
N	Number of hidden states
L	Length of the Markov chain / observation sequence
K	Number of t-mixture components for every hidden state emission
$\mathcal{S}$	Student's t-density function
$Y = \{y_t\}_{t=1}^{L}$	Set of observations
$s_i = \{\alpha_{i,k}, \mu_{i,k}, \Sigma_{i,k}, \nu_{i,k}\}_{k=1}^{K}$	Parameters of the kth component ith hidden state's t-mixture for $k \in \{1, \dots, K\}$
$\lambda = (\lambda_{i,j})_{1 \le i,j \le N}$	$N \times N$ matrix, where $\lambda_{i,j}$ is the transition probability from state s_i to s_j
$\lambda^{t0} = (\lambda_i^{t0})_{1 \le i \le N}$	Vector of initial probabilities of hidden states at $t = 0$
$\psi_i(y_t)$	Emission function of the observation y_t by the state s_j

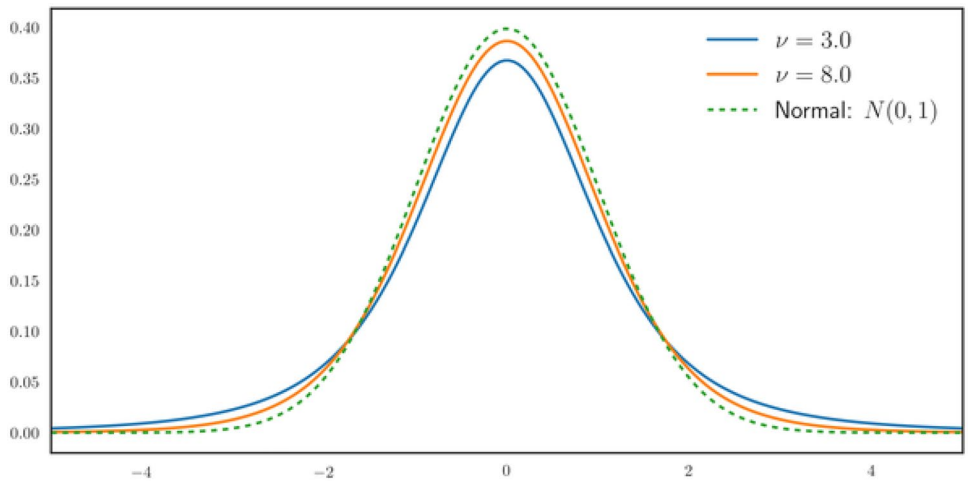

Fig. 1 Student's-t versus Gaussian probability density functions (univariate case)

2.1 Multivariate bounded asymmetric student's t-distribution

We begin this section by building up the mathematical definitions that hold the basis for our model, starting with the multivariate Student's t-distribution, and leading up to the BASMM.

Let t be a multivariate Student's-t probability density function with the following parameters: a mean μ, a covariance matrix Σ, and ν degrees of freedom. For a multivariate vector x of dimension d, and given the aforementioned parameters, Student's-t can be written as follows [27]:

$$t(x|\mu, \Sigma, \nu) = \frac{\Gamma\left(\frac{\nu+d}{2}\right)|\Sigma|^{-1/2}(\nu\pi)^{-d/2}}{\Gamma(\nu/2)[1 + \nu^{-1}\Delta(x, \mu;\Sigma)]^{(\nu+d)/2}} \tag{1}$$

where $\Gamma(x)$ is the Gamma function and $\Delta(x, \mu;\Sigma)$ is the squared Mahalanobis distance. Both functions have the following definitions, respectively:

$$\Gamma(y) = \int_0^\infty x^{y-1}e^{-x}\,dx \quad ; \quad y > 0 \tag{2}$$

$$\Delta(x, \mu;\Sigma) = (x - \mu)^T\Sigma^{-1}(x - \mu) \tag{3}$$

It is worth noting that the definition of the t-distribution density function differs from a univariate to a multivariate population. Considering that most of the real-world data-related tasks feature multivariate observations, we will not tackle the univariate case in this paper. Hence, all the probability density functions, as well as the rest of the mathematical construction of our model are presented for a multivariate random variable x. If we add the asymmetry to the multivariate t, where we have a left covariance Σ_l and a right covariance Σ_r, we would have the following density function $\mathcal{T}$:

$$\mathcal{T}(x|\mu, \Sigma_l, \Sigma_r, \nu) = \begin{cases} t(x|\mu, \Sigma_l, \nu) & \text{if } x \le 0 \\ t(x|\mu, \Sigma_r, \nu) & \text{otherwise} \end{cases} \tag{4}$$

We determine whether the multivariate vector x is less than the zero of $\mathbb{R}^d$ by calculating the sum A of all the components of the x:

$$A = \sum_{i=1}^{d} x_i \tag{5}$$

If $A < 0$, then $x < 0$, otherwise we consider $x \geq 0$. When we add bounded support Ω to the multivariate asymmetric t density function, we get the following probability density function $\mathcal{S}$:

$$\mathcal{S}(x|\theta) = \frac{\mathcal{T}(x|\mu, \Sigma_l, \Sigma_r, \nu) \times h(x, \Omega)}{\int_\Omega \mathcal{T}(y|\mu, \Sigma_l, \Sigma_r, \nu)\, dy} \tag{6}$$

where h is an indicator function that bounds the multivariate t by $\Omega \in \mathbb{R}^d$ and is defined as follows:

$$h(x, \Omega) = \begin{cases} 1 \text{ if } x \in \Omega \\ 0 \text{ otherwise} \end{cases} \tag{7}$$

where $\theta = \{\mu, \Sigma_l, \Sigma_r, \nu, \Omega\}$ is the set of parameters that fully define the multivariate bounded asymmetric t-distribution.

2.2 Relation to the multivariate Gaussian distribution

According to [27, 28], the multivariate t-distribution is conditionally related to the normal distribution: if the random variable x follows multivariate t-distribution with a mean μ, a covariance matrix Σ, and ν degrees of freedom, then given a precision parameter ϕ, x follows a multivariate Gaussian distribution n with mean μ and covariance $\frac{\Sigma}{\phi}$ and where the parameter ϕ is a Gamma-distributed [29] variable with both scale and shape parameters equal to $\frac{\nu}{2}$: $\phi \sim \mathcal{G}(\frac{\nu}{2}, \frac{\nu}{2})$ (See Eq. 8)).

$$\begin{aligned} x \sim t(\mu, \Sigma, \nu) \iff x|\phi \sim n\left(\mu, \frac{\Sigma}{\phi}\right) \text{ and} \\ \phi \sim \mathcal{G}\left(\frac{\nu}{2}, \frac{\nu}{2}\right) \end{aligned} \tag{8}$$

By applying Bayes' theorem, we find that the multivariate t-density function is the product of the Gaussian distribution and the Gamma distribution with the parameters explained above, which gives us Eq. 9).

$$t(x|\mu, \Sigma, \nu) = n\left(x|\mu, \frac{\Sigma}{\phi}\right) \times \mathcal{G}(\phi) \tag{9}$$

where $\mathcal{G}$ is the Gamma probability density function with both scale and shape parameters equal to $\frac{\nu}{2}$:

$$\mathcal{G}(\phi) = \frac{\left(\frac{\phi\nu}{2}\right)^{\frac{\nu}{2}} \exp\left(\frac{-\phi\nu}{2}\right)}{\phi\Gamma(\frac{\nu}{2})} \tag{10}$$

As for the multivariate Gaussian distribution with a mean vector μ and a covariance matrix Σ, the probability density function is:

$$n(x|\mu, \Sigma) = \frac{\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)}{\sqrt{(2\pi)^k|\Sigma|}} \tag{11}$$

Suppose we want to add bounded support and asymmetry to this definition of the multivariate Student's t. In that case, we can base it on an asymmetric multivariate Gaussian density function, then multiply it by the indicator function h (see Eq. 7)) and divide it by the integral over bounded the support region Ω, which yields the following density function:

$$\mathcal{S}(x|\theta) = \frac{\mathcal{N}\left(x|\mu, \frac{\Sigma_l}{\phi}, \frac{\Sigma_r}{\phi}\right) \times \mathcal{G}(\phi) \times h(x, \Omega)}{\int_\Omega \mathcal{T}(y|\mu, \Sigma_l, \Sigma_r, \nu)\, dy} \tag{12}$$

where $\mathcal{T}$ is the asymmetric multivariate t-probability density function (as presented in Eq. 13)), and where $\mathcal{N}$ is the asymmetric multivariate Gaussian density function, which takes as parameters a mean vector, a left covariance matrix, and a right covariance matrix. In order to define this density function, we follow the same approach stated in Sect. 2.1 for the multivariate asymmetric t:

$$\mathcal{N}(x|\mu, \Sigma_l, \Sigma_r) = \begin{cases} n(x|\mu, \Sigma_l) \text{ if } x \leq 0 \\ n(x|\mu, \Sigma_r) \text{ otherwise} \end{cases} \tag{13}$$

Building up these definitions of the multivariate Bounded Asymmetric Student's t-probability density function helps us understand the HMM observation emission probability that we will employ, to construct the entire model well.

2.3 Multivariate bounded asymmetric t-mixture model

Representing the distribution of a dataset X as a BASMM with K components implies that for every vector x_i of the dataset, the marginal probability density function of x_i is written as follows:

$$\begin{aligned} f(x_i|\Theta) = \sum_{k=1}^{K} c_k \times \mathcal{S}(x_i|\theta_k) = \sum_{k=1}^{K} c_k \\ \times \mathcal{S}(x_i|\mu_k, \Sigma_{l,k}, \Sigma_{r,k}, \nu_k, \Omega_k) \end{aligned} \tag{14}$$

where c_k and θ_k are the mixing proportion and the set of parameters for the *kth* mixture component respectively, and finally, the mixture's full set of parameters is $\Theta = \{\theta_1, \ldots, \theta_K; c_1, \ldots, c_K\}$. The mixing proportion c_k represents the prior probability that x_i belongs to the *kth* component, thus satisfies:

$$c_k \geq 0 \quad \text{and} \quad \sum_{k=1}^{K} c_k = 1 \tag{15}$$

2.4 Fitting the mixture model

Now to ensure that the mixture model fits the data in the most optimal way, we perform the EM algorithm [30] to adjust the parameters of the model Θ to find the closest representation to the modeled data. As its name suggests, the EM algorithm comprises two main steps: Expectation and Maximization.

2.4.1 Expectation step

The Expectation step consists of estimating the log-likelihood of the mixture model, i.e., how accurate is the representation of the data by the model with the current initialized set of parameters Θ. Here, at iteration t of the EM algorithm, we define the log-likelihood as the logarithm of the BASMM's probability density function of the data x. Since the data points are considered independent and identically distributed (IID), the BASMM's density function is the product over all the marginal density function values of the data vectors $(x_i)_{i=1}^{i=N}$. Thus the BASMM's log-likelihood is defined as follows:

$$\begin{aligned} L(\Theta) &= \log\Big(\prod_{i=1}^{N} f(x_i|\Theta)\Big) \\ &= \sum_{i=1}^{N} \log\Big(\sum_{k=1}^{K} c_k P(x_i|\theta_k)\Big) \end{aligned} \tag{16}$$

where $\Theta = \{\theta_1, \ldots, \theta_K; c_1, \ldots, c_K\}$ and $\theta_k = \{\mu_k, \Sigma_{k,l}, \Sigma_{k,r}, \nu_k, \Omega_k\}$ for $1 \leq k \leq K$. In the same expectation step, we define by z_{ik} the posterior probability that the vector x_i belongs to the *kth* component for $i \in \{1, \ldots, N\}$ and $k \in \{1, \ldots, K\}$. These posterior probabilities are called responsibilities in mixture models terminology. They signify how responsible a mixture component (a simple bounded asymmetric t-distribution in the case of BASMM) is for a data vector x_i, i.e., the amount of contribution of a distribution/mixture component θ_k over the quantity produced by the BASMM. At each iteration t of the Expectation step, the responsibility values $\left(z_{ik}^{(t)}\right)_{i=k=1}^{i=N,k=K}$ are computed by the following equation:

$$z_{ik}^{(t)} = \frac{c_k^{(t)} P(x_i|\theta_k^{(t)})}{\sum_{j=1}^{K} c_j^{(t)} P(x_i|\theta_j^{(t)})} \tag{17}$$

2.4.2 Maximization step

The goal of the Maximization step in the EM algorithm is to update the model parameters to maximize the previously calculated log-likelihood function [31]. As the logarithm is monotonically increasing, it is more suitable to minimize the negative log-likelihood function $J(\Theta) = -L(\Theta)$. The followed logic here is to calculate the partial derivatives of $J(\Theta)$ with respect to the different parameters and update those parameters as the solution to the equation:

$$\text{Partial derivative}(J) = 0$$

All solutions to this equation with respect to each parameter of the BASMM will require the knowledge of the responsibilities/posterior probabilities z_{ik} calculated in the expectation step. In turn, the responsibilities depend on the knowledge of the parameters of each mixture component θ_k. This explains the iterative nature of the EM algorithm. The M step of this algorithm, as well as the updated parameters' definitions are elaborated in details in [26].

3 Hidden Markov models

3.1 Bounded asymmetric student's t-mixture model hidden Markov model (BASMMHMM)

Here we present the main contribution of our model, which is the observation emission strategy. As discussed in the introduction, we aim to produce an HMM with emissions that are more robust to the observable data's outliers. In this context, the Student's t-distribution has been employed in modified versions as a non-Gaussian emission in [22, 32]. We build on these works by exploring asymmetry and bounded support along with the t-mixture for the emission. For this particular type of HMM, we consider that at the time t, the probability of observing y_t given a hidden state s_i follows a probability distribution formed by a mixture of bounded asymmetric Student's t-distributions with K components. We also consider that for all the hidden states of the HMM, the number of mixture components is the same. As a result, the probability of emitting the observation y_t from the hidden state s_i is defined in the following equation:

$$\begin{aligned} P(y_t|\Theta_i) &= \sum_{k=1}^{K} c_{i,k} \times \mathcal{S}(y_t|\theta_{i,k}) \\ &= \sum_{k=1}^{K} c_{i,k} \times \mathcal{S}(y_t|\mu_{i,k}, \Sigma_{i,k}^{l}, \Sigma_{i,k}^{r}, \nu_{i,k}, \Omega_{i,k}) \end{aligned} \tag{18}$$

With the definition of the multivariate t is given in Eqs. (1), (13). It's hard and computationally costly to run the EM algorithm when fitting the HMM. In this case, we employ the definition based on the bounded asymmetric Gaussian stated in 2.2. As a result, the probability of emitting the *tth* observation y_t by the hidden state s_i (which corresponds to

the emission mixture model Θ_i with the set of parameters $\left(\theta_{i,k} = \{\mu_{i,k}, \Sigma_{l,i,k}, \Sigma_{r,i,k}\, \nu_{i,k}, \Omega_{i,k}\}\right)_{k=1}^{K}$) is the following:

$$P(y_t|s_i) = \sum_{k=1}^{K} \frac{c_{i,k} \times \mathcal{N}\left(y_t|\mu_{i,k}, \frac{\Sigma_{l,i,k}}{\phi_{i,k}}, \frac{\Sigma_{r,i,k}}{\phi_{i,k}}\right) \times \mathcal{G}(\phi_{i,k}) \times h(y_t, \Omega_{i,k})}{\int_{\Omega_{i,k}} \mathcal{T}(y|\mu_{i,k}, \Sigma_{l,i,k}, \Sigma_{r,i,k}, \nu_{i,k})\, dy} \tag{19}$$

where $\phi_{i,k}$ is a precision parameter and $\phi_{i,k} \sim \mathcal{G}(\frac{\nu}{2}, \frac{\nu}{2})$ (see Sect. 2.2). We define also the observation indicators $\left(\delta_{i,t}\right)_{i=t=1}^{i=I,t=L}$ by:

$$\delta_{i,t} = \begin{cases} 1 \text{ if the observation } y_t \text{ is emitted from the hidden state } s_i \\ 0 \text{ otherwise} \end{cases} \tag{20}$$

Also, given $\delta_{i,t} = 1$, we define the state-conditional mixture component indicators $\left(\eta_{i,k,t}\right)_{k=1}^{K}$ as follows:

$$\eta_{i,k,t} = \begin{cases} 1 \text{ if } y_t \text{ is emitted from the } k^{th} \text{ mixture component of the hidden state } s_i \\ 0 \text{ otherwise} \end{cases} \tag{21}$$

These indicators are latent variables that give information about the mixture component that each data point belongs to. We don't have this information, but defining it mathematically gives us a complete data representation: y^c, thus simplifying the equations, i.e., the complete data probability density function of each emission mixture:

$$P(y^c|s_i) = \prod_{k=1}^{K} \left[c_{i,k} \times \mathcal{N}\left(y|\mu_{i,k}, \frac{\Sigma_{l,i,k}}{\phi_{i,k}}, \frac{\Sigma_{r,i,k}}{\phi_{i,k}}\right) \times \frac{\mathcal{G}(\phi_{i,k}) \times h(y, \Omega_{i,k})}{\int_{\Omega_{i,k}} \mathcal{T}(y|\mu_{i,k}, \Sigma_{l,i,k}, \Sigma_{r,i,k}, \nu_{i,k})\, dy} \right]^{\eta_{i,k,t}} \tag{22}$$

After calculations, the log-likelihood of the emission mixture for the *ith* hidden state is given by:

$$\begin{aligned} \log P(y^c|s_i) &= \log\left[\prod_{k=1}^{K} c_{i,k} \times \mathcal{S}(y|\theta_{i,k})^{\eta_{i,k,t}}\right] \\ &= \sum_{k=1}^{K} \eta_{i,k,t} \times \left[-\log\Gamma\left(\frac{\nu_{i,k}}{2}\right) + \frac{\nu_{i,k}}{2}\left(\log\left(\frac{\nu_{i,k}}{2}\right) - \phi_{i,k} + \log\phi_{i,k}\right)\right. \\ &\quad - \frac{1}{2}\left(\log|\Sigma_{i,k}| + d\log(2\pi) + \phi_{i,k}\Delta(y, \mu_{i,k}; \Sigma_{i,k})\right) \\ &\quad \left. - \log\int_{\Omega_{i,k}} \mathcal{T}(y|\mu_{i,k}, \Sigma_{l,i,k}, \Sigma_{r,i,k}, \nu_{i,k})\, dy\right] \end{aligned} \tag{23}$$

where $\Sigma_{i,k}$ can be the left or the right covariance matrix based on whether $y \leq 0$ or otherwise.

3.2 Defining the log-likelihood of the BASMMHMM

The likelihood of the BASMMHMM $E(\mathcal{M})$ defines how well the model fits the data (set of observations). Thus, $E(\mathcal{M})$ is obtained by calculating the joint emission probabilities of the observation sequence $Y = \{y_t\}_{t=1}^{L}$ by every hidden state's BASMM:

$$\begin{aligned} E(\mathcal{M}) &= \left(\prod_{i=1}^{N} \lambda_i^{\delta_{i,1}}\right) \\ &\times \left(\prod_{i=1}^{N}\prod_{j=1}^{N}\prod_{t=1}^{L-1} \lambda_{i,j}^{\delta_{i,t}\times\delta_{j,t+1}}\right) \\ &\times \left(\prod_{j=1}^{N}\prod_{t=1}^{L} P(y_t^c|s_j)^{\delta_{j,t}}\right) \end{aligned} \tag{24}$$

Following this, the log likelihood of the BASMMHMM is given by:

$$\begin{aligned} \mathcal{L}(\mathcal{M}) &= \log\left(E(\mathcal{M})\right) \\ &= \sum_{i=1}^{N}\left(\delta_{i,1}\log\lambda_i + \sum_{j=1}^{N}\sum_{t=1}^{L-1}\delta_{i,t}\delta_{j,t+1}\log\lambda_{i,j}\right) \\ &\quad + \sum_{j=1}^{N}\sum_{t=1}^{L}\delta_{j,t}\log P(y_t^c|s_j) \end{aligned} \tag{25}$$

3.3 Training the BASMMHMM

The goal of training the Bounded Asymmetric Student's-t Hidden Markov Model is to find the optimal set of model parameters $\left\{\lambda_i, \lambda_{i,j}, s_j\right\}_{i,j=1}^{N,N}$ that best fits the sequence of observations $Y = \left(y_t\right)_{t=1}^{L}$. This is done by maximizing the likelihood (see Eq. 25)) in an EM algorithm. let $\rho_{i,t}$ and $\rho_{i,j,t}$ be the posterior emission probabilities defined as follows:

$$\rho_{i,t} = P(\delta_{i,t} = 1|y_t) \tag{26}$$

$$\rho_{i,j,t} = P(\delta_{j,t+1} = 1, \delta_{i,t} = 1|y_t) \tag{27}$$

To perform the training, we use the Baum–Welch algorithm. Our purpose here is to tune the parameters of the HMM, namely the state transition matrix, the emission matrix, and the initial state distribution, such that the model is maximally like the observed data. In short, Baum–Welch is a sort

of EM algorithm, where the E-step consists of forward and backward phases [33].

3.3.1 Baum–Welch: expectation

1. Calculate the forward value α, where $\alpha_t(i)$ is the probability of being in the *ith* state after the first t observations of the model, given the set of properties Θ.
2. Calculate the backward value β, where $\beta_t(i)$ is the probability of being in the *ith* state at the *tth* timestamp and seeing the observations from timestamp $t+1$ until the end of the sequence, given the set of properties Θ.
3. Calculate the posterior transition probabilities $\rho_{i,j,t}$: the probability of being in state i at time t then being in state j at time $t+1$. $\rho_{i,j,t}$ is calculated using the forward and backward values as follows:

$$\begin{aligned}\rho_{i,j,t} &= \frac{\alpha_t(i) \times \lambda_{i,j} P(y_{t+1}|s_j) \times \beta_{t+1}(j)}{P(Y|\Theta)} \\ &= \frac{\alpha_t(i) \times \lambda_{i,j} P(y_{t+1}|s_j) \times \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \left[\alpha_t(i) \times \lambda_{i,j} P(y_{t+1}|s_j) \times \beta_{t+1}(j)\right]}\end{aligned} \tag{28}$$

4. Calculate the posterior emission values $\rho_{i,t}$, i.e., the probability of being in the *ith* state at the time t, given the observations Y and the model Θ. We get the emission posteriors by summing over the $\rho_{i,j,t}$ values for all states:

$$\rho_{i,t} = \sum_{j=1}^{N} \rho_{i,j,t} \tag{29}$$

5. Calculate $\mathcal{Q}(\mathcal{M})$, the expectation of the log-likelihood of the BASMMHMM:

$$\begin{aligned}\mathcal{Q}(\mathcal{M}) &= E(\mathcal{L}(\mathcal{M})) \\ &= \sum_{i=1}^{N} \left(\rho_{i,1} \log \lambda_i + \sum_{j=1}^{N} \sum_{t=1}^{L-1} \rho_{i,j,t} \log \lambda_{i,j}\right) \\ &\quad + \sum_{j=1}^{N} \sum_{t=1}^{L} \rho_{j,t} E\big(\log P(y_t^c|s_j)\big)\end{aligned} \tag{30}$$

3.3.2 Baum–Welch: maximization

In maximization, we use the variables calculated in the expectation step to update the HMM properties: prior weights and emission mixtures for each hidden state. We proceed in the following steps:

1. Update the initial hidden state probabilities $\left(\lambda_i^{t_0}\right)_{i=0}^{N}$ by using the γ values:

$$\widehat{\lambda_i^{t_0}} = \rho_{i,t_0} \quad ; \quad i \in \{1, 2, \ldots, N\} \tag{31}$$

2. Update the state transition probabilities:

$$\begin{aligned}\widehat{\lambda_{i,j}} &= \frac{\text{number of transitions from } s_i \text{ to } s_j}{\text{number of transitions from } s_i} \\ &= \frac{\sum_{t=1}^{L-1} \rho_{i,j,t}}{\sum_{t=1}^{L} \rho_{i,t}}\end{aligned} \tag{32}$$

3. Update the properties of the BASMM for each hidden state of the model: the means $(\mu_{i,k})_{i=k=1}^{i=N,k=K}$, the covariances, the mixing weights and the degrees of freedom.

$$\widehat{\mu_{i,k}} = \frac{\sum_{t=1}^{L} \xi_{i,k,t}(u_{i,k}(y_t) y_t - A_{i,k})}{\sum_{t=1}^{L} \xi_{i,k,t} u_{i,k}(y_t)}; \tag{33}$$

where $\xi_{i,k,t}$ is the *ith* state's mixture component membership posterior, i.e., the probability that the observation y_t is emitted from the *kth* component of the *ith* hidden state:

$$\xi_{i,k,t} = \frac{\rho_{i,t} c_{i,k} \mathcal{S}(y_t|s_{i,k})}{\sum_{j=1}^{K} c_{i,j} \mathcal{S}(y_t|s_{i,j})} \tag{34}$$

And where $A_{i,k}$ is defined by using a sample of data points $(S_m)_{m=1}^{m=M}$ that is drawn from the *kth* component of the *ith* hidden state's mixture:

$$A_{i,k} = \frac{\sum_{m=1}^{M} (S_m - \mu_{i,k}) u_{i,k}(S_m) h(S_m, \Omega_{i,k})}{\sum_{l=1}^{M} h(S_l, \Omega_{i,k})} \tag{35}$$

And $u_{i,k,t}$ is the precision function for an observation y_t of dimension d:

$$u_{i,k}(y_t) = \frac{d + \nu_{i,k}}{\nu_{i,k} + \Delta(y_t, \mu_{i,k}; \Sigma_{i,k})} \tag{36}$$

The mixing weights $(c_{i,k})_{i=k=1}^{i=N,k=K}$ are updated by dividing the probability of emission from the *kth* mixture component of the *ith* hidden state by the total probability of being in that *ith* state at any timestamp in the Markov chain:

$$\begin{aligned}\widehat{c_{i,k}} &= \frac{\sum_{t=1}^{L} \xi_{i,k,t}}{\sum_{t=1}^{L} \sum_{l=1}^{K} \xi_{i,l,t}} \\ &= \frac{\sum_{t=1}^{L} \xi_{i,k,t}}{\sum_{t=1}^{L} \rho_{i,t}}\end{aligned} \tag{37}$$

The covariances $(\Sigma_{i,k})_{i=k=1}^{i=N,k=K}$ are updated as follows:

$$\widehat{\Sigma_{i,k}} = \frac{\sum_{t=1}^{L} \xi_{i,k,t} u_{i,k,t} \times (y_t - \mu_{i,k})(y_t - \mu_{i,k})^T}{\sum_{t=1}^{L} \xi_{i,k,t}} - B_{i,k} \tag{38}$$

where $B_{i,k}$ is given by:

$$B_{i,k} = \frac{\sum_{m=1}^{M} \left(\Sigma_{i,k} - (S_m - \mu_{i,k})(S_m - \mu_{i,k})^T u_{i,k}(S_m)\right) h(S_m, \Omega_{i,k})}{\sum_{m=1}^{M} h(S_m, \Omega_{i,k})} \tag{39}$$

Next, the update of the degrees of freedom for each hidden state's mixture component is the solution to the equation below:

$$\begin{aligned} & g(\nu_{i,k}, d) + 1 + \frac{1}{\sum_{t=1}^{L} \xi_{i,k,t}} \sum_{t=1}^{L} \xi_{i,k,t} \Big(\log u_{i,k}(y_t) - u_{i,k}(y_t) \Big) \\ & - \frac{1}{\sum_{m=1}^{M} h(S_m, \Omega_{i,k})} \sum_{m=1}^{M} \Big(g(\nu_{i,k}, d) + 1 + \log u_{i,k}(S_m) - u_{i,k}(S_m) \Big) \\ & = 0 \end{aligned} \tag{40}$$

where ψ is the digamma function and $g(\nu, d)$ is defined as:

$$g(\nu, d) = -\psi(\frac{\nu}{2}) + \log(\frac{\nu}{2}) + \psi(\frac{\nu + d}{2}) - \log(\frac{\nu + d}{2}) \tag{41}$$

There is no closed-form solution to the Eq. 40), so we use the Newton Raphson method [34] to derive the optimal update of $\nu_{i,k}$. Finally, we update the bounds of each hidden state's mixture model by fetching the minimums and maximums among the observations that were attributed to each mixture component in the expectation step.

4 Experiments and results

In this section, we select a few popular sequential data-based applications where we attempt to employ the BASM-MHMM, then evaluate its performance in comparison with baseline models among the following:

- Gaussian Hidden Markov Model (GHMM)
- Gaussian Mixture Hidden Markov Model (GMMHMM)
- Student Mixture Hidden Markov Model (SMMHMM)
- Student Hidden Markov Model (SHMM)

Our approach is to measure how much the Bounded Asymmetric Student's t-Mixture emissions can elevate the HMM's performance. That is why the baseline models mentioned above are all variants of HMM with different emission distributions.

4.1 Occupancy estimation

In the field of smart buildings, occupancy estimation is a frequently performed operation as it is useful for many tasks, namely energy saving, consumption tracking, and employee presence monitoring for companies. Therefore, we find that many works have extensively tackled this subject, like [35, 36]. So in this experiment, we also attempt to estimate the number of occupants in one room using signals from non-intrusive sensors.

4.1.1 Data

The dataset [37] that we used for this experiment comprises signals obtained from seven non-intrusive sensors of five different types: temperature, illumination, sound, CO2, and passive infrared (PIR). As Fig. 2 shows, sensor nodes S1-S4 were deployed at the desks (referred to as desk nodes). These desk nodes have temperature, light, and sound sensors only. Node S5 has a CO2 sensor kept in the middle to get the best possible measure in the room. Nodes S6 and S7 only contain PIR sensors and are put on the ceiling at an angle that maximizes the sensor's field of view for motion detection.

The obtained data from these nodes spans 21 days (from 22 December 2017 to 11 January 2018) and has been recorded every 30 s, which gives us a time series of 10129 timestamps. As for the ground truth room occupancy, it varies between 0 and 3. We model this information as the hidden state of our HMM, which would give us 4 hidden states. The observations are the signals sent by sensors, in the case of this experiment, these observations would be vectors of a dimension $d = 16$ as there are 16 distinct records taken from the sensors in total.

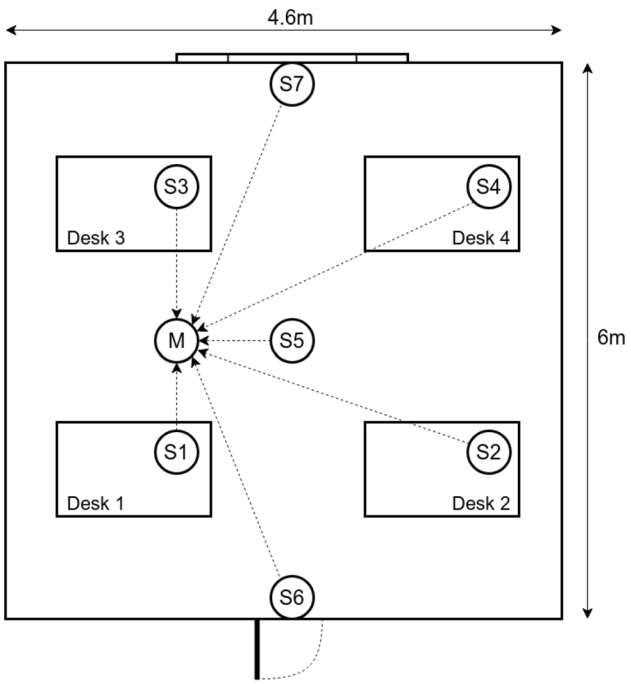

Fig. 2 Sensors' layout in the room

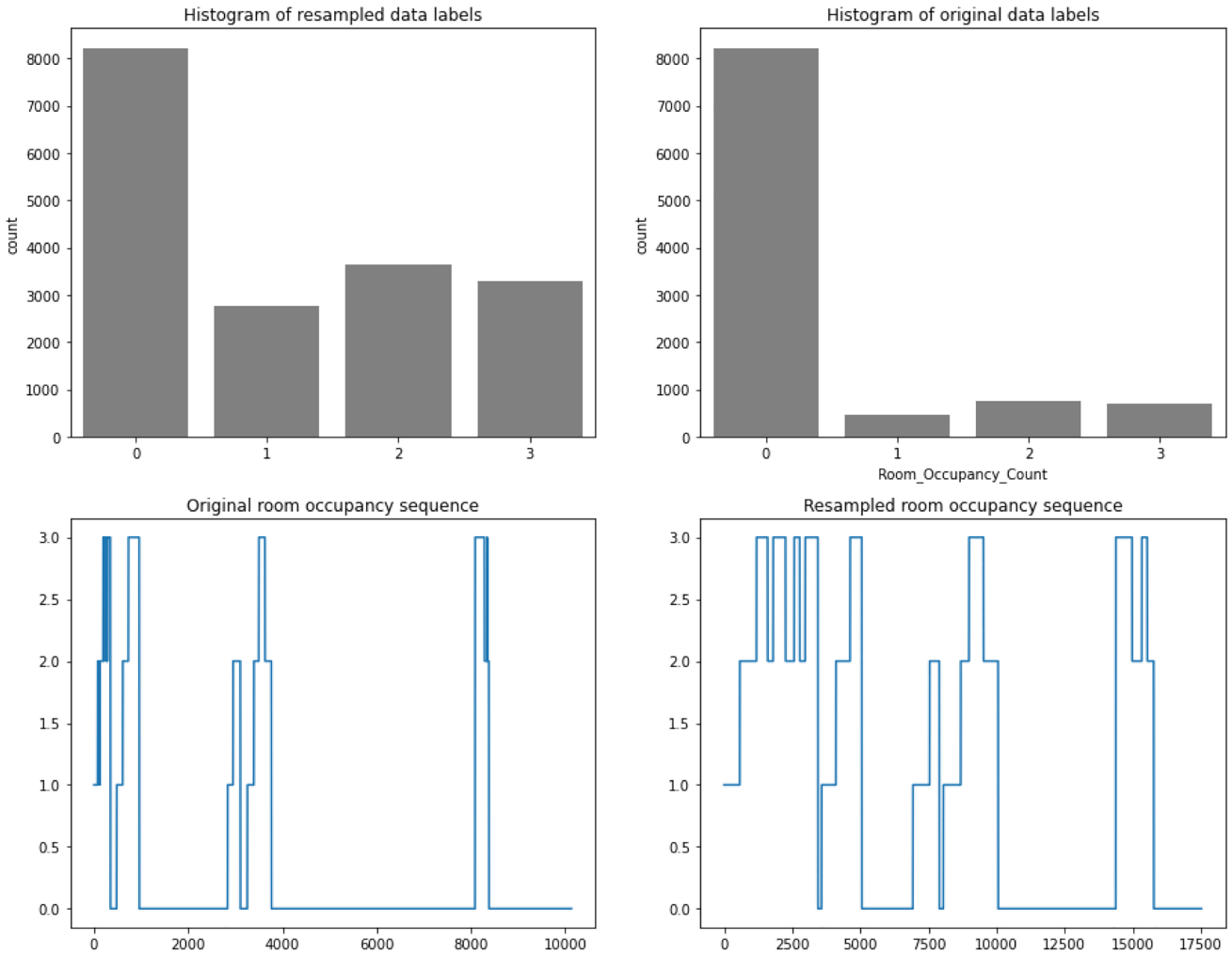

Fig. 3 Original data versus resampled data

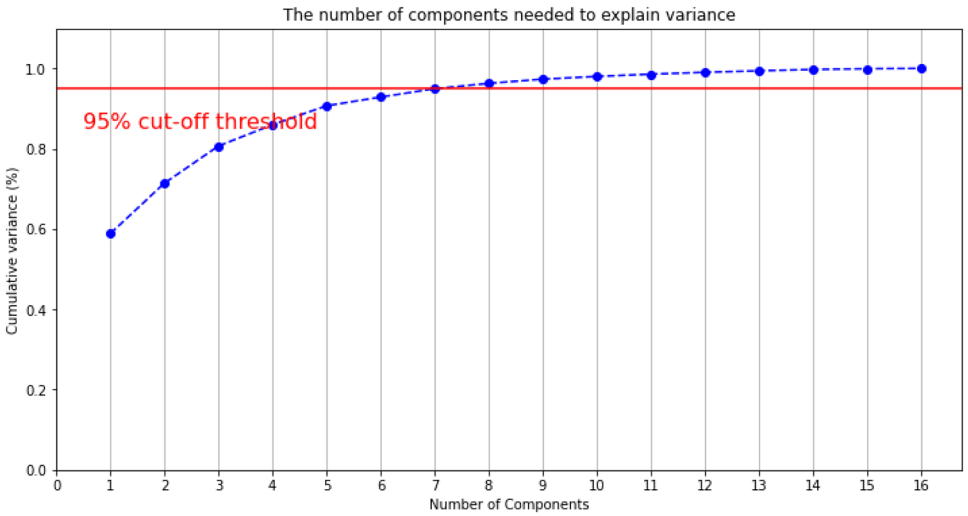

Fig. 4 Data variance depending on the number of principal components

Table 2 Occupancy estimation: accuracy and F1 score weighted averages for different models

Algorithm	Accuracy	Precision	Recall	Average F1
BASMMHMM	0.86	0.87	0.86	0.86
SMMHMM	0.82	0.82	0.82	0.82
SHMM	0.77	0.77	0.77	0.77
GMMHMM	0.74	0.73	0.74	0.73
GHMM	0.71	0.84	0.71	0.69

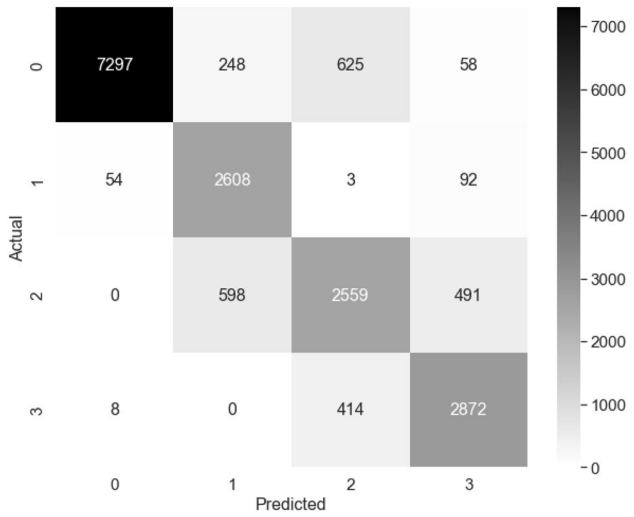

Fig. 5 Occupancy estimation: confusion matrix of BASMMHMM

4.1.2 Preprocessing

When we observe the labels (number of occupants over time), we find a clear imbalance, as for most of the recording time, there's no one in the room, thus, the number of occupants is zero.

We cope with the imbalance by oversampling the minority classes. For that, we use the SMOTE technique [38]. However, we don't make the classes equally partitioned, and this is to keep some outliers and the overall occupancy sequence patterns. The results of oversampling are shown in the Fig. 3.

After oversampling, we scale the data using the MinMax method. We then perform a PCA to reduce the number of features and the computation complexity. The number of principal components is chosen in a way that keeps the variance of the data above 0.95. Based on Fig. 4, we choose eight principal components.

4.1.3 Results

We run the BASMMHMM and a selection of other benchmark models (SMMHMM, SHMM, GMMHMM, GHMM) on the preprocessed data, taking the room occupancy numbers as hidden states. When fitting the models, we run the EM algorithm for a number of iterations ranging from 1 to 100, and we take the number of iterations that gives the best result for each model. After multiple experiments with the different mixture-based HMMs on the data, we take $K = 3$ as the number of mixture components, as it produces the best fit for the data-set. The weighted averages of the accuracy, precision, recall, and F-1 score are presented in the following Table 2.

According to the results above, the BASMMHMM clearly performed better than the rest, as it produced the highest accuracy and F1-score of 0.86, where the second best results were an accuracy and an F1-score of 0.82 for the SMMHMM. The models based on Student's-t emissions gave better metrics than those based on the Gaussian emissions. This is mainly due to a bad prediction of the outliers (hidden states 1, 2 and 3) by the Gaussian-based models because as mentioned earlier, there is a dominant label in the time series (0 occupants most of the time). What is common between all the models is that they performed well with the majority hidden state 0. The confusion matrix in Fig. 5 shows that the BASMMHMM predicts well all the classes/hidden states of the data, despite their imbalance (class 0 is more occurrent than the rest). In comparison, the confusion matrices of the other models show in Fig. 6 show a limited prediction of the non-majority classes. The weighted averages of the accuracy, precision, recall, and F-1 score when using the original data without oversampling are presented in Table 3. According to the results, we can see that the BASMMHMM still gives relatively good results, considering the complexity of this highly imbalanced data, while outperforming the other models.

4.2 Stock price prediction

The stock market is an important indicator that reflects economic growth: when the economy grows, this typically translates into an upward trend in stock prices. In contrast, when the economy slows, stock prices tend to be more mixed. For traders, it is important to predict the behaviour of these numbers (stock prices) to take the appropriate action and achieve profit. But this prediction task is not easy, as several uncertain parameters like economic conditions, policy changes, supply and demand between investors, etc, determine the price trend. These parameters vary, thus making stock markets volatile.

4.2.1 Data and preprocessing

We use the stock price time-series made available by Yahoo Finance API. This API contains records of multiple

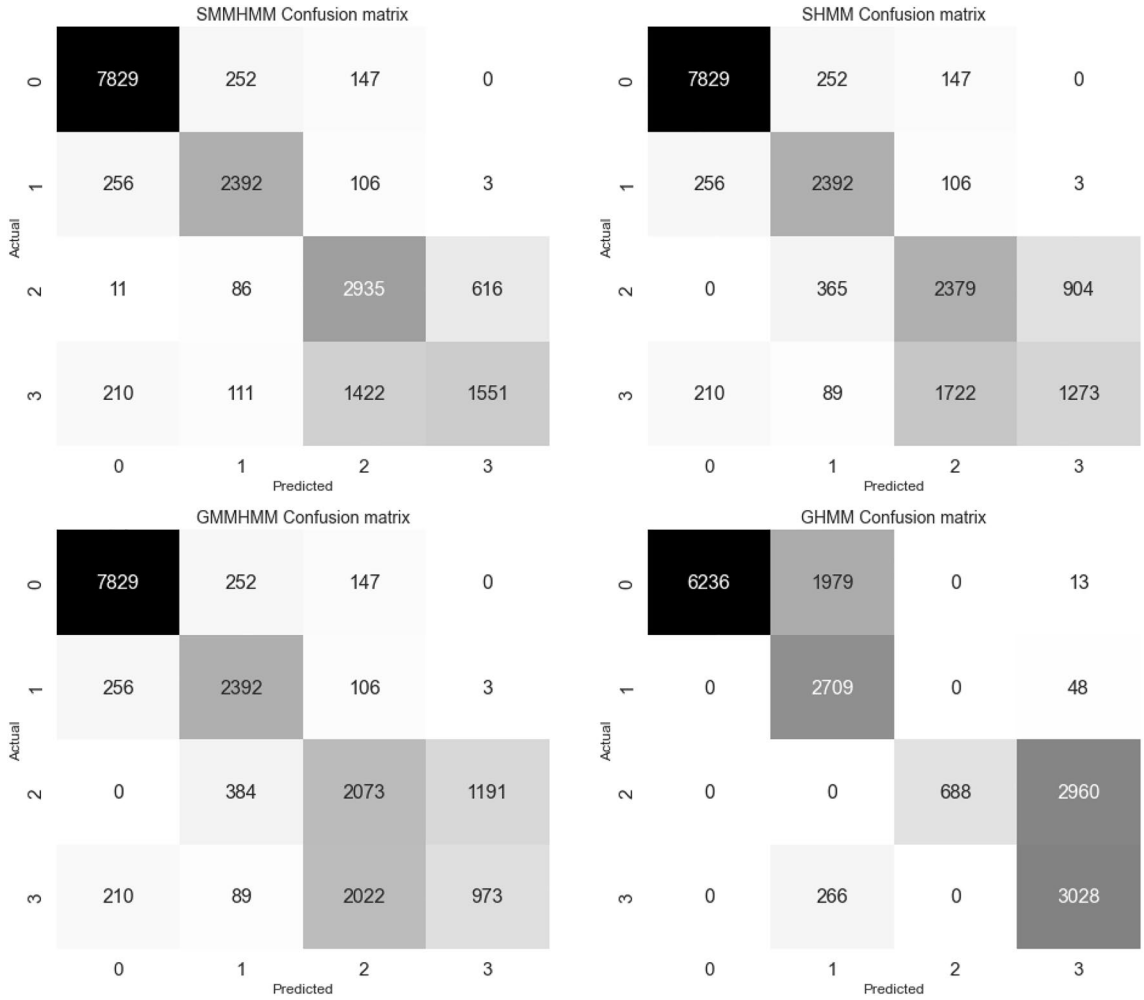

Fig. 6 Occupancy estimation: confusion matrices of other HMMs

Table 3 Occupancy estimation: accuracy and F1 score weighted averages for different models using original data

Algorithm	Accuracy	Precision	Recall	Average F1
BASMMHMM	0.74	0.75	0.73	0.73
SMMHMM	0.70	0.69	0.69	0.70
SHMM	0.67	0.65	0.64	0.65
GMMHMM	0.63	0.64	0.63	0.63
GHMM	0.61	0.60	0.61	0.60

companies' stock prices spanning long periods of time. For our experiment, we select three different companies' datasets: Amazon (AMZN), Apple (AAPL), and Google (GOOGL). For each of these three companies, the

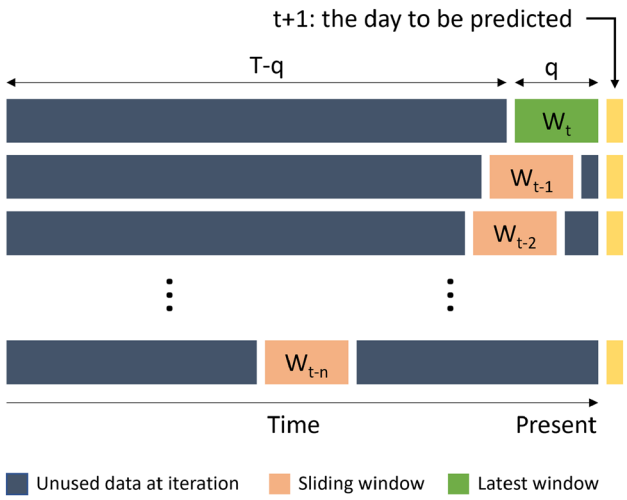

Fig. 7 Forecasting the t+1 stock prices based on a sliding window of past k days

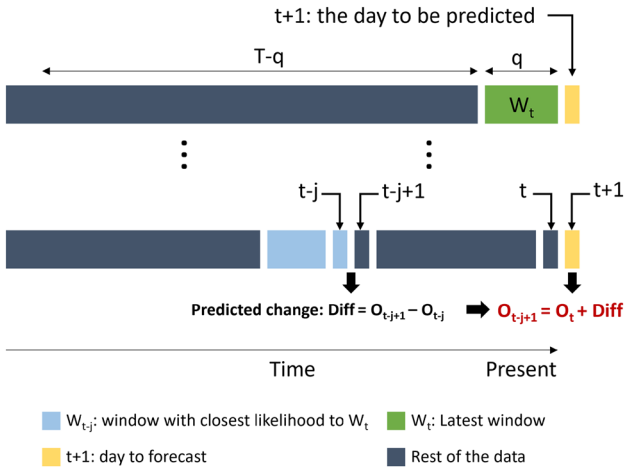

Fig. 8 Predicted time-series calculation

time-series that we used spans over the 12 years from 1 January 2010 to 1 January 2022 and is multivariate with four variables: opening price, high price, low price, and closing price. As for the preprocessing, we perform a Min-Max scaling on the data before passing it to the HMM. After the forecasting, we unscale the results produced by the model, and we compare them to the unscaled ground-truth data to view the model's performance.

4.2.2 Forecasting approach

Our task is to predict the stock prices for a given day t. To do this, we adopt the following method: First, we fit the BASMMHMM to the data (the time-series of the until the day $t-1$), then we proceed to predict based on sliding time windows W_j of fixed length q (where W_j is the data of last q-day sequence ending with the day j): we calculate the log-likelihood[1] of each sliding window, take the window with the closest log-likelihood to W_t and calculate the day $t+1$ predictions based on that chosen window.

Table 4 AMZN stock price prediction: performance metrics for different models

Parameter	BASMMHMM		SMMHMM		GMMHMM	
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
Open price	0.00889	0.41951	0.01292	0.67489	0.01594	0.81723
High price	0.00615	0.06994	0.01054	0.15782	0.01382	0.20840
Low price	0.00951	0.31669	0.01429	0.43916	0.01396	0.39053
Close price	0.00751	0.12392	0.01276	0.27641	0.01520	0.30048

Table 5 AAPL stock price prediction: performance metrics for different models

Parameter	BASMMHMM		SMMHMM		GMMHMM	
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
Open price	0.00720	0.01989	0.01135	0.05712	0.01300	0.06293
High price	0.00728	0.15320	0.01027	0.30822	0.00982	0.21833
Low price	0.00925	0.19009	0.01263	0.35702	0.01392	0.29666
Close price	0.00862	0.10429	0.01304	0.23833	0.01328	0.27142

Table 6 GOOGL stock price prediction: performance metrics for different models

Parameter	BASMMHMM		SMMHMM		GMMHMM	
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
Open price	0.00674	0.30320	0.01248	0.42088	0.01298	0.48512
High price	0.00602	0.14920	0.02015	0.32612	0.01602	0.37298
Low price	0.00749	0.08447	0.01894	0.31086	0.01978	0.29172
Close price	0.00740	0.03534	0.02381	0.10664	0.02146	0.15840

[1] The log-likelihood of a sequence of observations given the BASMMHMM that we trained on the data.

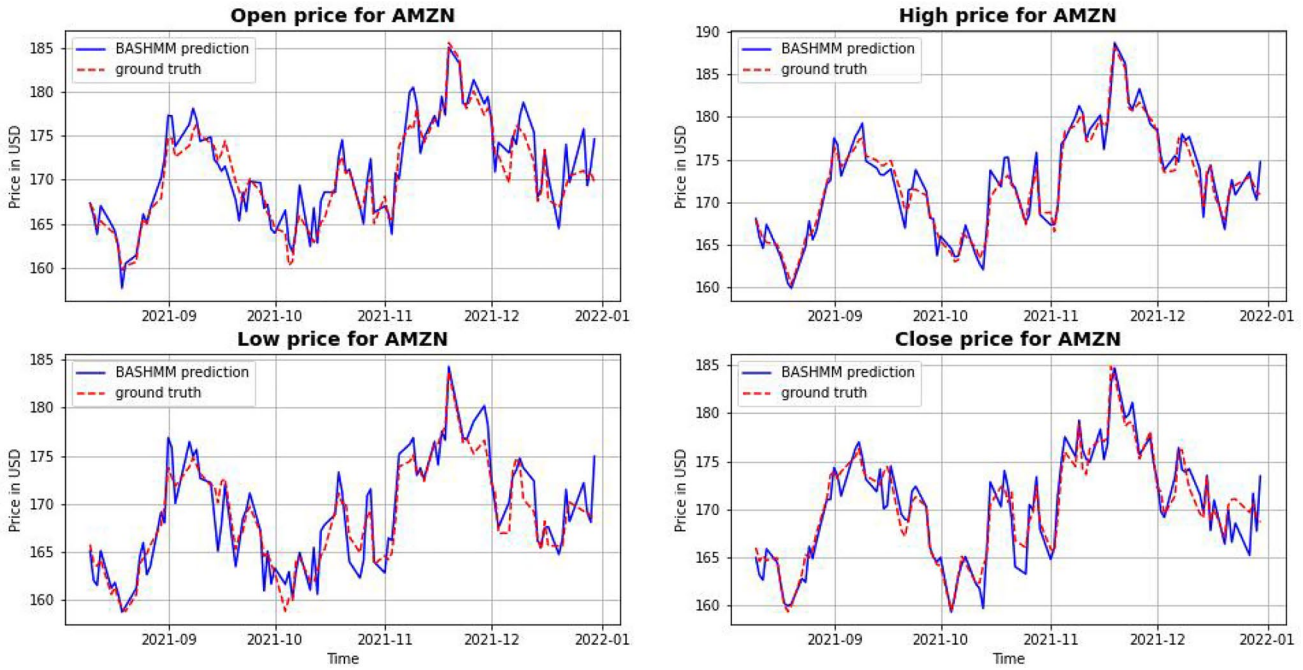

Fig. 9 Amazon stock prices: BASMMHMM prediction versus ground truth

The adopted approach is further explained in Figs. 7 and 8 below.

4.2.3 Results

After performing the forecasting, we established a comparison between BASMMHMM and a selection of other models using the two following performance metrics:

- *MAPE:* Short for Mean Absolute Percentage Error, is the average absolute error between the actual and predicted stock values in percentage. The formula is:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{y_i - x_i}{x_i} \times 100 \tag{42}$$

  where n is the length of the time-series, and for $i \in \{1, 2, \ldots, n\}$, y_i is the predicted value and x_i is the actual value.
- *RMSE:* The Root Mean Square Error is the square root of the mean of the square of all of the errors between the actual and the predicted data. The RMSE is widely used, and it is considered an excellent general purpose error metric for numerical predictions. Considering the notations used in Eq. 42), the RMSE formula is the following:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \tag{43}$$

Tables 4, 5 and 6 indicate the metrics found after the forecasting of the stock prices of Amazon, Apple, and Google, respectively. The prediction on multivariate stock price data with four variables: Open, High, Low, and Close prices, but in the tables, we focus mainly on the High price variable. The BASMMHMM has been run with a custom number of hidden states N and sliding window size q. The BASMMHMM with the combination $\{N, q\}$ that gives the best performance is elected. As for the number of mixture components of the emissions, it is selected using the Minimum Message Length criterion [30]. In this experiment, the BASMMHMM is compared to the SMMHMM and GMMHMM.

According to the tables above, BASMMHMM generally performed better than SMMHMM and GMMHMM. This is mainly explained by the outliers and the local minima/maxima being better predicted by the BASMMHMM. It is also worth mentioning that the models based on Student's t-mixture emissions (BASMMHMM, SMMHMM) performed better than the GMMHMM, which is based on Gaussian mixture emissions. We can see the graphs in Figs. 9, 10 and 11 a more clear picture of the predicted versus the actual stock prices.

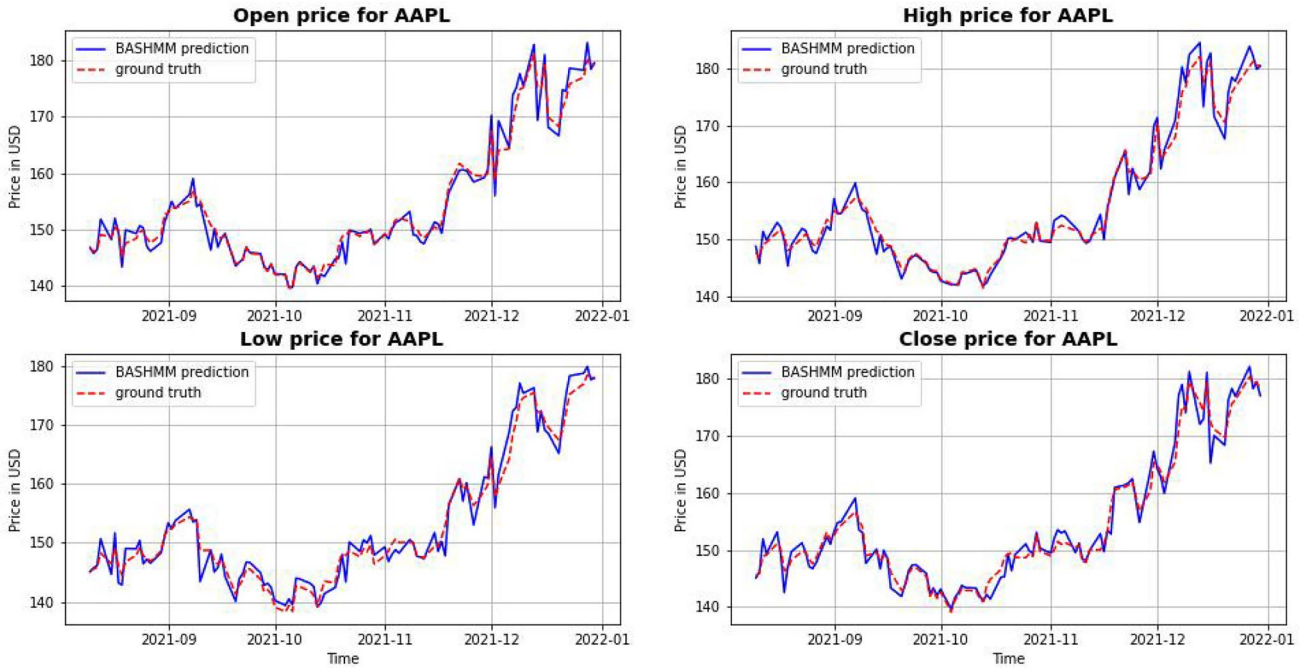

Fig. 10 Apple stock prices: BASMMHMM prediction versus ground truth

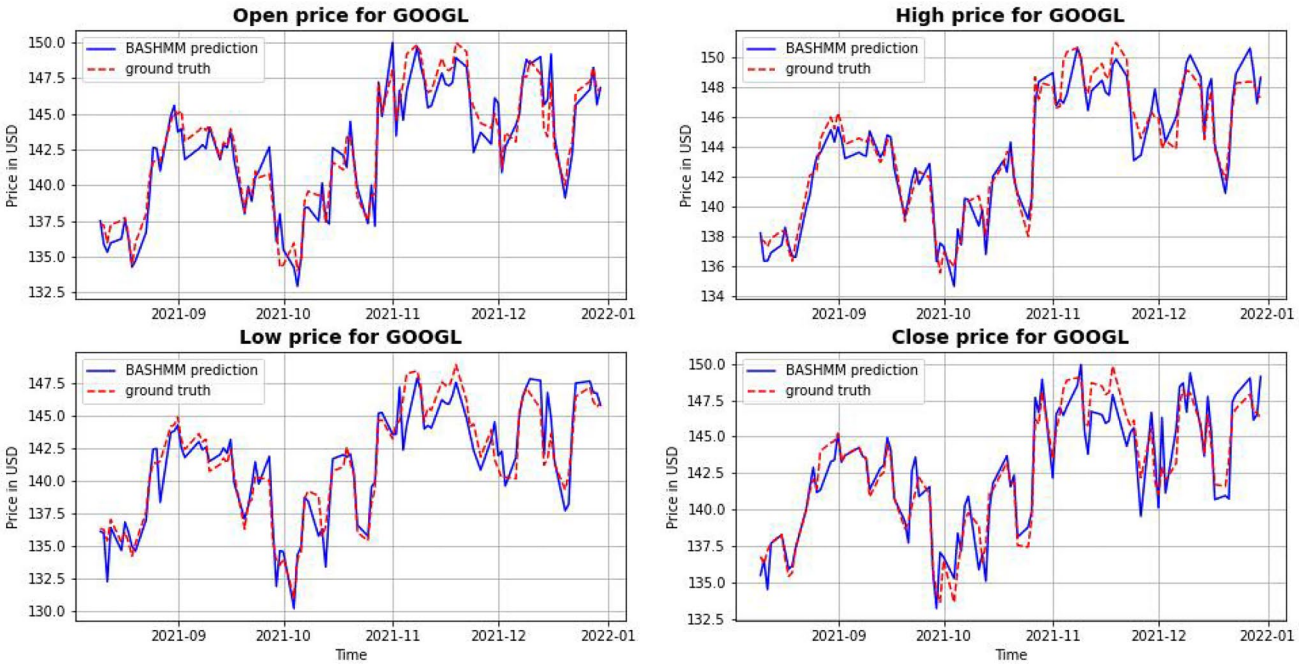

Fig. 11 Google stock prices: BASMMHMM prediction versus ground truth

4.3 Human activity recognition

Human Activity Recognition (HAR) is a popular scientific application that enables machines to recognize human body behaviours. HAR is useful for many real-world tasks, such as fall detection in elderly healthcare monitoring or physical exercise measuring and tracking in sport science. In this experiment, we use the dataset provided by UCI [39], which is popularly used in many research works.

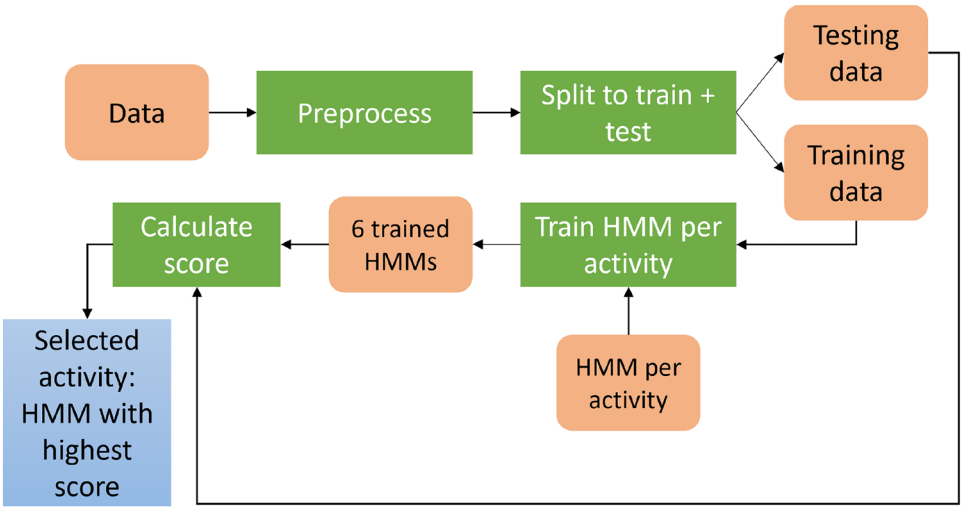

Fig. 12 Human activity recognition: BASMMHMM framework

4.3.1 Dataset and preprocessing

The data at hand consists of 10299 records, each record having 561 features (features are signals received from smartphone sensors). The labels of the data are the different activities performed at the time of recording, and they are mainly six: Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing, and Laying.

The preprocessing consists of MinMax scaling and then reducing the features with the Principal Component Analysis method. We perform the PCA in a way that keeps the variance of the data above 0.95, which gives us 69 principal components.

In this experiment, we use a training sample of 7352 observations and a testing sample of 2947 observations. We create one HMM for every activity, which gives us six HMMs in total. The parameters of each HMM are learned from the corresponding activity's training set with the Baum–Welch algorithm. In the testing phase, for each part of the test set, we calculate all six trained HMMs' likelihood to have generated the observations, and the correspondant activity to the HMM with the highest likelihood is selected as the prediction label. for all six HMMs, we choose 2 hidden states and $K = 2$ mixture components per hidden state. The Fig. 12 summarizes the pipeline of the modeling in this experiment.

4.3.2 Results

Upon performing the prediction of the human activities, we calculate the weighted averages of the accuracy, precision, recall, and F1 score of the predicted labels. These weighted-averages are calculated by taking the mean of all per-class metrics while considering each class's support. Support refers to the number of actual occurrences of the class in the dataset. The 'weight' essentially refers to the proportion of each class's support relative to the sum of all support values.

Table 7 HAR: Accuracy and F1 score weighted averages for different models

Algorithm	Accuracy	Precision	Recall	Average F1
BASMMHMM	0.79	0.79	0.79	0.79
SMMHMM	0.71	0.71	0.71	0.71
SHMM	0.68	0.69	0.68	0.68
GMMHMM	0.67	0.67	0.67	0.66
GHMM	0.61	0.62	0.61	0.6

The BASMMHMM did a better performance than the rest of the models, as shown in Table 7. The accuracy and the F1 score are close to 0.8, which is an improvement compared to the SMMHMM, which gave about 0.7. It is also worth mentioning that the models with emissions based on the Student's t-mixture and distribution performed slightly better than the ones with emissions based on the Gaussian mixture and distribution.

It is noteworthy that the computational complexity of each iteration of the Baum–Welch algorithm is $\mathcal{O}(LN^2)$ which shows the practicality and scalability of HMMs in general and BASMHMM in particular in real-world applications, especially in scenarios where computationally hungry models are generally avoided (e.g., Federated learning).

5 Conclusion

In this paper, we proposed the use of bounded asymmetric Student's t-mixture models as the observation emission densities of continuous HMMs to offer a more robust methodology for sequential data modeling. We then presented different experiments where we applied BASMMHMM, which

proved an enhanced performance compared to other benchmark HMM-based models. More specifically, the BASM-MHMM can be a strong candidate for solving data outlier and asymmetry problems with its high flexibility.

We can conclude that adding a custom emission to the HMM, such as the Bounded Asymmetric Student's t-Mixture, results in higher adaptability to the model, regardless of its applications. We presented the mathematical formulation of our model, and backed it up by results of different experiments. Applications such as occupancy estimation, stock price prediction and human activity recognition showed a better performance for the BASMMHMM in comparison to other Student's t and Gaussian-based HMMs. The data anomalies are taken into consideration, thus making the BASMMHMM a very useful tool while tackling real world datasets. This also can save us the extra preprocessing that removes the outliers and might often end up altering the data, hence making our modeling "isolated" from the real information/experiment.

Finally, there is room to improve the proposed model and expand the work on many aspects. For instance, the number of emission mixture components is an important parameter to tune for the HMM to ensure optimal fit to the data. Introducing an adequate model selection [40] approach before training the HMM can fulfill this tuning. Furthermore, in the case of high dimensional observations, it is rigorous to implement a feature selection strategy [41] to avoid high computational complexity and to elect the parameters that represent the data in the most efficient way.

Data availability Data could be made available on reasonable request.

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Yusoff MIM, Mohamed I, Bakar MRA (2014) Hidden Markov models: an insight. In: Proceedings of the 6th international conference on information technology and multimedia, pp 259–264, IEEE
2. Hou W, Fan W, Amayri M, Bouguila N (2022) A novel continuous hidden Markov model for modeling positive sequential data. Hidden Markov models and applications. Springer, New York, pp 199–210
3. Norris JR (1998) Markov chains. Cambridge University Press, Cambridge
4. Chung KL (2012) Markov chains: with stationary transition probabilities. Springer, Heidelberg
5. Blunsom P (2004) Hidden Markov models. Lecture notes, August 15(18-19), p 48
6. Stamp M (2004) A revealing introduction to hidden Markov models. Department of Computer Science San Jose State University, pp 26–56
7. Eddy SR (1996) Hidden Markov models. Curr Opin Struct Biol 6(3):361–365
8. Durbin R, Eddy SR, Krogh A, Mitchison G (1998) Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge University Press, Cambridge
9. Zucchini W, Guttorp P (1991) A hidden Markov model for space-time precipitation. Water Resour Res 27(8):1917–1923
10. Ephraim Y, Merhav N (2002) Hidden Markov processes. IEEE Trans Inf Theory 48(6):1518–1569
11. Nguyen N (2017) An analysis and implementation of the hidden Markov model to technology stock prediction. Risks 5(4):62
12. Ney H, Ortmanns S (1999) Dynamic programming search for continuous speech recognition. IEEE Signal Process Mag 16(5):64–83
13. Juang BH, Rabiner LR (1991) Hidden Markov models for speech recognition. Technometrics 33(3):251–272
14. Miller DR, Leek T, Schwartz RM (1999) A hidden Markov model information retrieval system. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 214–221
15. Volant S, Bérard C, Martin-Magniette M-L, Robin S (2014) Hidden Markov models with mixtures as emission. Stat Comput 24(4):493–504
16. Shaikh SA, Kitagawa H (2014) Efficient distance-based outlier detection on uncertain datasets of gaussian distribution. World Wide Web 17(4):511–538
17. Xian Z, Azam M, Amayri M, Fan W, Bouguila N (2022) Bounded asymmetric gaussian mixture-based hidden Markov models. Hidden Markov models and applications. Springer, New York, pp 33–58
18. Azam M, Alghabashi B, Bouguila N (2020) Multivariate bounded asymmetric gaussian mixture model. Mixture models and applications. Springer, New York, pp 61–80
19. Li R, Nadarajah S (2020) A review of student'st distribution and its generalizations. Empir Econ 58(3):1461–1490
20. Peel D, McLachlan GJ (2000) Robust mixture modelling using the t distribution. Stat Comput 10(4):339–348
21. Chatzis SP, Kosmopoulos DI, Varvarigou TA (2008) Robust sequential data modeling using an outlier tolerant hidden Markov model. IEEE Trans Pattern Anal Mach Intell 31(9):1657–1669
22. Zhang H, Wu QMJ, Nguyen TM (2013) Modified student's t-hidden Markov model for pattern recognition and classification. IET Signal Proc 7(3):219–227
23. Zheng Y, Jeon B, Sun L, Zhang J, Zhang H (2017) Student's t-hidden Markov model for unsupervised learning using localized feature selection. IEEE Trans Circuits Syst Video Technol 28(10):2586–2598
24. Ali S, Bouguila N (2022) A roadmap to hidden markov models and a review of its application in occupancy estimation. In: Hidden Markov Models and Applications, pp 1–31
25. Asghari P, Soleimani E, Nazerfard E (2020) Online human activity recognition employing hierarchical hidden markov models. J Ambient Intell Humaniz Comput 11(3):1141–1152
26. Nguyen TM, Wu QJ (2013) Bounded asymmetrical student's-t mixture model. IEEE Trans Cybern 44(6):857–869
27. Liu C, Rubin DB (1995) Ml estimation of the t distribution using em and its extensions, ecm and ecme. Statistica Sinica, pp 19–39
28. Kibria BG, Joarder AH (2006) A short review of multivariate t-distribution. J Stat Res 40(1):59–72
29. Thom HC (1958) A note on the gamma distribution. Mon Weather Rev 86(4):117–122
30. Oliver JJ, Baxter RA, Wallace CS (1996) Unsupervised learning using mml. ICML. Citeseer, Princeton, pp 364–372
31. Bishop CM (2006) Pattern recognition and machine learning, Springer New York, NY 2006

32. Chen Z, Yang Y (2012) Fault diagnostics of helicopter gearboxes based on multi-sensor mixtured hidden Markov models. J Vib Coust 134(3):031010
33. Collins M (2013) The forward-backward algorithm. Columbia Columbia University, New York
34. Ypma TJ (1995) Historical development of the Newton–Raphson method. SIAM Rev 37(4):531–551
35. Amayri M, Ngo Q.-D, Ploix S, et al. (2017) 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), IEEE, vol. 2, pp 690–695
36. Nasfi R, Amayri M, Bouguila N (2020) A novel approach for modeling positive vectors with inverted dirichlet-based hidden Markov models. Knowl-Based Syst 192:105335
37. Singh AP, Jain V, Chaudhari S, Kraemer FA, Werner S, Garg V (2018) Machine learning-based occupancy estimation using multivariate sensor nodes. In: 2018 IEEE Globecom Workshops (GC Wkshps), IEEE
38. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
39. Reyes-Ortiz J-L, Anguita D, Ghio A, Parra X (2012) Human activity recognition using smartphones data set. UCI Machine Learning Repository; University of California, Irvine, School of Information and Computer Sciences: Irvine, CA, USA
40. Celeux G, Frühwirth-Schnatter S, Robert CP (2019) Model selection for mixture models-perspectives and strategies. Handbook of mixture analysis. Chapman and Hall/CRC, New York, pp 117–154
41. Ali S, Bouguila N (2022) Hidden Markov models: discrete feature selection in activity recognition. Hidden Markov models and applications. Springer, New York, pp 103–155