Generative Models Based on the Bounded Asymmetric Student's t-Distribution

Ons BOUARADA

A Thesis in The

Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Quality Systems Engineering) at

Concordia University

Montréal, Québec, Canada

March 2023

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Ons BOUARADA**

Entitled: **Generative Models Based on the Bounded Asymmetric Student's t-Distribution**

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

Master of Applied Science (Quality Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair and Internal Examiner
Dr. Manar Amayri

_____ Internal Examiner
Dr. Jamal Bentahar

_____ Supervisor
Dr. Nizar Bouguila

Approved by _____
Dr. Zachary Patterson, Graduate Program Director

_____ 2023

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

Abstract

Generative Models Based on the Bounded Asymmetric Student's t-Distribution

Ons BOUARADA

Gaussian mixture models (GMMs) are a very useful and widely popular approach for clustering, but they have several limitations, such as low outliers tolerance and assumption of data normality. Another problem in relation to finite mixture models in general is the inference of an optimal number of mixture components. An excellent approach to solve this problem is model selection, which is the process of choosing the optimal number of mixture components that ensures the best clustering performance. In this thesis, we attempt to tackle both aforementioned issues: we propose using minimum message length (MML) as a model selection criterion for multivariate bounded asymmetric Student's t-mixture model (BASMM). In fact, BASMM is chosen as an alternative to improve the GMM's limitations, as it provides a better fit for the real-world data irregularities. We formulate the definition of MML and the BASMM, and we test their performance through multiple experiments with different problem settings.

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 typically modeled by elliptically contoured distributions, namely Gaussians or Student's t-distributions. In this context, this thesis proposes BAMMHMM: a novel HMM with Bounded Asymmetric Student's t-Mixture Model (BASMM) emissions. This HMM is destined to sufficiently fit skewed and outlier-heavy observations, which are typical in many fields, such as financial or signal processing-related datasets. We demonstrate the improved robustness of our model by presenting the results of different real-world applications.

Acknowledgments

I would like to express my gratitude to my supervisor *Prof. Nizar Bouguila* for his guidance and support throughout my master's degree. He provided me with academic advice and helped whenever I felt stuck in my work. This master's degree under his supervision is not only an academic improvement in my resumé, but also a life lesson for my professional future.

I would like to also extend my thanks to *Dr. Muhammad Azam* who helped me from the beginning of my research work until my graduation. His valuable advice and assistance in the technical details of my work were crucial to the completion of this thesis. He was also very patient with me and motivated me in moments of doubt. I am also very thankful for *Dr. Manar Amayri* who provided me with important technical advice on many aspects of my work.

Finally, I am deeply and forever grateful for my mother Chadlia and my father Slah for their continuous and unconditional support. I can not do them justice for the amount of sacrifice and hard work they did to get me to where I am today. I owe them everything.

This thesis is dedicated to my dear grandparents who sadly passed away in 2022: my grandmother Mné, and my grandfather Khalifa.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The ever-growing presence of technology and software in our daily lives continues to produce a large abundance of data in all its shapes and forms, and new sets of data-related problems. It is necessary to analyze these data to gain insights and make better decisions, or to train machine learning models on it to make predictions or automate tasks. However, the greater part of these data is often unlabeled, which calls for the importance of unsupervised learning methods to analyze and model it. In this regard, clustering [1] is one of the most popular techniques to discover and classify unlabeled data.

Mixture models [2], a notable range of clustering approaches, provide statistical inference on sub-populations of various random phenomena [3]. Mixture models represent the data as an ensemble of clusters following the same probability distribution, with a different set of parameters per cluster. One important characteristic of mixture model-based clustering is that it is a soft clustering approach: it fits a set of probabilistic models to the data and assigns each data vector a probability of belonging to each component. This gives us a quantification of the probability of each vector belonging to each cluster. Thanks to their high flexibility and their capacity to model complex data distributions, mixture models are the subject of an increasing attention and have considered various distributions [4, 5, 6], such as the Gaussian distribution, which is the basis for GMMs [7].

The choice of the mixture components' probability distribution is a determinant factor in the quality of the clustering. This choice assumes a certain shape and distribution about the data, which might or might not be accurate. As an example, the aforementioned GMMs are very convenient and widely used in multiple clustering tasks and research works [8, 9, 10], but they may not always be the optimal choice. Specifically, many datasets are not naturally distributed in a Gaussian-like way, and GMMs are not a suitable clustering method in this case. Furthermore, the symmetry and the unbounded support of the Gaussian distribution prevents it from fitting optimally to real-world datasets that are generally asymmetric and bounded. Some research works have overcome these issues by using the generalized Gaussian distribution [11] for the mixture model (GGMM) [12, 13, 14] or by introducing asymmetry and/or bounded support to the GMMs [15] and GGMMs [16, 17, 18]. However, even with these added properties, one limitation persists, and that is the low tolerance to outliers. In fact, outliers are often present in datasets generated by real-world applications. In this case, GMMs require excessively big sizes of the available training datasets to capture the outliers and guarantee the dependability of the model fitting procedure.

On a different note, the performance of mixture model clustering can be sensitive to several other factors, such as the number of mixture components and the initial parameter estimates. In fact, tuning the number of clusters/mixture components for the best fit to the data representation is a task as crucial as the clustering itself. Too many components lead to overfitting the data, and too few components result in a very simplistic model that fails to capture the complexity of the data.

1.1.1 Bounded Asymmetric Student's t-Mixture Model (BASMM)

To address the GMM's limitations, the Student's t-distribution is the basis of the mixture model used in this work. By its definition, the Student's t-distribution is more heavily tailed than the Gaussian distribution, thus the Student's t-mixture model (SMM) is more robust to outliers than the Gaussian mixture model (GMM) [19]. In real-world data, the values are usually concentrated within bounds and distributed in a non-symmetric way. Therefore, it is suitable to introduce bounded support and asymmetry [20] to every component of the mixture model, which allows more flexibility and better fitting to the different shapes of the data. The multivariate Student's-t mixture model has a small number of parameters: mean, covariance matrix, degrees of freedom and mixing parameter for

each component of the mixture. When fitting the SMM to a dataset for clustering, these parameters can be estimated in an iterative fashion using the Expectation Maximization (EM) algorithm [21]. In this thesis, we use the multivariate bounded asymmetric Student's t-mixture model (BASMM) as our main clustering approach. We develop the mathematical background for this model and for its related EM algorithm. We also employ BASMM for different experiments and establish a performance comparison with other benchmark model selection approaches.

1.1.2 Model Selection Criterion for BASMM

In unsupervised learning, the number of mixture components is often unknown and needs to be provided before proceeding to the learning phase. Choosing the number of components has a significant effect on the clustering performance. In fact, too many components can result in overfitting, and too few components reduce the model's flexibility and prevent us from learning the real data representation.

Model selection [22] is introduced to overcome this issue, and it is the process of choosing the optimal number of clusters based on the data and the clustering model at hand.Various strategies have been presented to identify the best number of components for mixture models [23]: an important range of these can be Bayesian and/or information theory-based criteria. The Bayesian approaches take into account the prior probability of the model and the likelihood of the data given the model. Very popular examples of these approaches include the Bayesian information criterion (BIC) [24] or the Laplace empirical criterion (LEC) [25]. Other Bayesian criteria that are based on information theory rely on seeking optimal encoding of the data given the mixture model and its likelihood. These methods include Akaike's information criterion (AIC) [24] and the minimum description range of criteria like minimum description length (MDL) criterion [26] or the mixture minimum description length (MMDL) [27].

Within this same range of criteria, we also cite the minimum message length (MML) [28]. This is an approach that finds the model with a minimal length of a message composed of the prior information and the encoding of the model fitted to the data. MML has been used in many recent works involving different variants of multivariate mixture models [29, 30, 31] and produced excellent results. It has also performed a better model selection than AIC and MDL according to [32]. This makes

MML an interesting model selection criterion to explore with BASMM clustering. In this thesis, we propose MML as a primary method to estimate the optimal number of mixture components for the BASMM. We lay out the mathematical definition for the MML and we test its performance in different data experiments.

1.1.3 Hidden Markov Models

HMMs [33] are a simple, yet powerful, tool to represent and predict sequential events [34] and are widely used in many types of data-driven tasks. The concept of HMMs is primarily based on Markov Chains [35, 36] (proposed by Andrey Markov in the early 20^{th} 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 [37, 38]: evaluation, decoding (inference), and learning. The evaluation is the computation of the probability of an observation sequence given an HMM. Decoding 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 [39, 40], climate modeling [41], signal processing [42], stock market prediction [43], speech [44], video recognition [45], and information retrieval systems [46], 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 among our areas of interest in this thesis.

1.1.4 BASMM emissions for HMMs

When modeling continuous data with HMMs, 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 [47]. This has led to some very useful variants of HMMs [48], 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, as discussed earlier, Gaussian-based distributions do not account for the outliers [49], data asymmetry, nor its 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 [50] by introducing a bounded asymmetric Gaussian mixture [16] 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 [51] is an excellent alternative to the Gaussian when fitting skewed or heavy-tailed populations, thus, the multivariate finite Student's t-Mixture Model (SMM) [52] 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 [53, 54, 55], 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 datasets. This motivates us to introduce BASMMHMM: a HMM with BASMM as an observation emission distribution. 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 account for its finite aspect. 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 [56], stock price prediction and human activity recognition [57].

1.2 Contributions

(1) **Model Selection Criterion For Multivariate Asymmetric Student's-t Mixture Model With Bounded Support Data:**

In this contribution, we propose the use minimum message length (MML) as a model selection criterion on top of the multivariate bounded asymmetric Student's t-mixture model (BASMM) clustering. We test our clustering method and model selection with three different experiments. The results of these experiments are discussed and compared with other model selection criteria and clustering algorithms to demonstrate the merits of our contribution. This work has been submitted as a journal paper to Advances in Data Analysis and Classification.

(2) **Hidden Markov Models with Multivariate Bounded Asymmetric Student's t-Mixture Model Emissions:**

In this contribution, We propose BASMMHMM: a new variant of hidden Markov models where the observation emissions are modeled in a multivariate bounded asymmetric Student's t-mixture model. This work allows us to explore the effectiveness and the improved robustness of BASMMs when integrated in an HMM framework. We test our proposed model in multiple experiments from different problem settings. The results of these experiments are discussed and compared with other variants of HMM based on other popular mixture models. This contribution has been submitted as a journal paper to the Journal of Pattern Analysis and Applications.

1.3 Thesis Overview

- In chapter 1, we introduce the general scope and set the motivations of this thesis. We also briefly review the existing related works and we describe our contributions.

- In chapter 2, we present the multivariate bounded asymmetric Student's t-mixture model (BASMM) and we lay out the mathematical background for this model's learning process with the EM algorithm. Then, we derivate of the model selection criterion for BASMM using minimum message length (MML) in detail. We also validate our work using multiple experiments from real-world applications.

- In Chapter 3, we integrate BASMM into the framework of hidden Markov models. We provide the mathematical description of the model in detail, and we test it on different applications: occupancy estimation, human activity recognition, and stock price prediction. We compare the results of these applications with other mixture model-based HMMs.

- In Chapter 4, we briefly summarize the contributions of this thesis and lay out some potential paths of improvement for our work.

Chapter 2

Multivariate Bounded support asymmetric Student's t-Mixture Model with Model Selection using Minimum Message Length

In this chapter, we consider the task of modeling multidimensional data by a multivariate bounded asymmetric Student's t-mixture model (BASMM) without knowing the prior number of clusters. We then propose minimum message length (MML) as a model selection criterion on top of the BASMM clustering. We test our resulting model with three different experiments: household electricity consumption profiles clustering, head pose estimation from drivers' images, and leukemia detection from genetic expression. The results of BASMM and MML in these experiments are discussed and compared with other model selection criteria and clustering algorithms to demonstrate the merits of our contribution.

2.1 Multivariate Bounded Asymmetric Student's-t Mixture Model

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, compared to 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. When ν increases, the t-pdf tends to have thinner tails and becomes closer to the Gaussian pdf (see Fig. 2.1). As a result, the t-distribution provides a heavy-tailed alternative to the Gaussian distribution for potential outliers in the data and therefore, the SMM is a more outlier-tolerant clustering approach than the GMM.

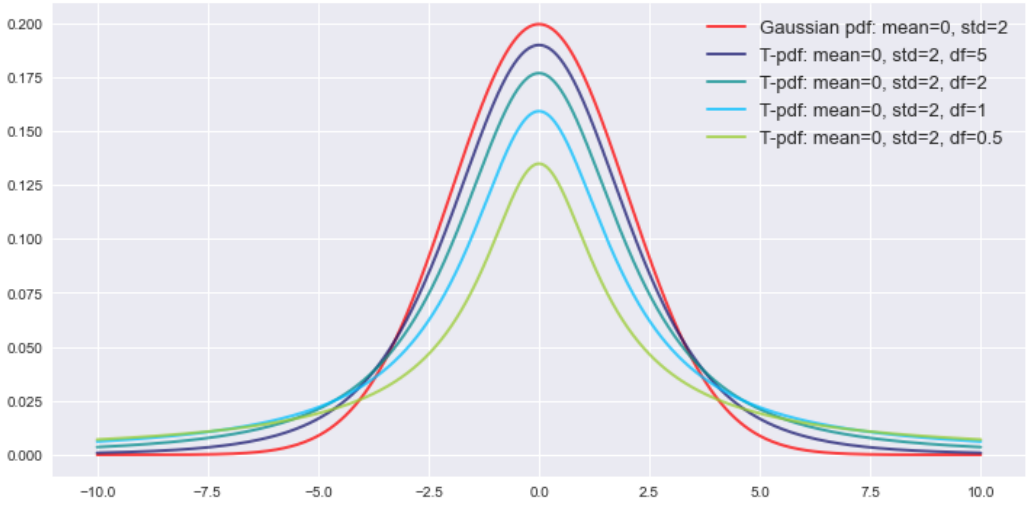

Figure 2.1: PDFs of a Gaussian distribution and four t-distributions with different degrees of freedom

In this section, we lay out the mathematical definitions for the bounded asymmetric Student's t-distribution and mixture model, along with its EM algorithm. It is worth mentioning that throughout this chapter, we consider all algorithms for multivariate data. So, $\vec{X}_i$, which is often the notation of a data point in this chapter, is a vector of two dimensions or more. This choice is explained by the fact that most formats of real-world data are multivariate. Hence, it is more convenient to consider mixture modeling for multivariate data.

2.1.1 Bounded Asymmetric Student's-t Distribution

The BASMM 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. Fig. 2.2 shows a comparison between the density functions of the Gaussian, the Student's t, and the bounded asymmetric Student's t-distribution. The mean and standard deviation are the same for all three pdfs, and the degrees of freedom are the same for the t and the bounded asymmetric t. Note that in Fig. 2.2, the forms of the t-based pdfs are more tailed than the Gaussian, and the bounded asymmetric t-pdf has the most irregular shape. While these pdfs are plotted for univariate populations, they are a good representation of these distributions' behaviour for multivariate data.

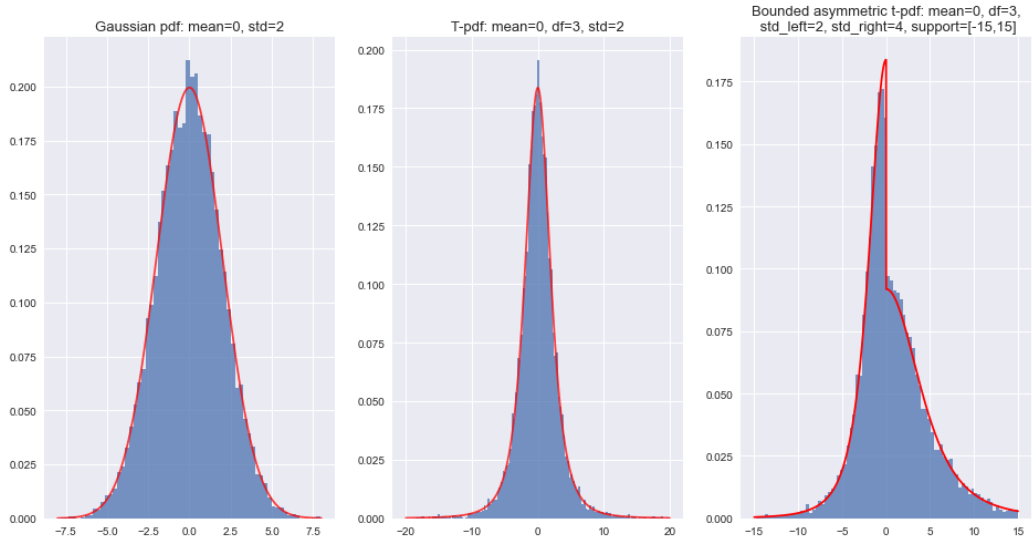

Figure 2.2: PDFs of three different distributions: Gaussian, Student's t, and bounded asymmetric Student's t

Let s be a multivariate Student's-t [58] probability density function with the following parameters: a mean $\vec{\mu} = [\mu_1, \ldots, \mu_d]$, a covariance matrix Σ and ν degrees of freedom. For a random vector $\vec{X} = [x_1, \ldots, x_d]$ of dimension d and given the aforementioned parameters, s can be written as follows:

$$s(\vec{X}|\vec{\mu}, \Sigma, \nu) = \frac{\Gamma(\frac{\nu+d}{2})|\Sigma|^{-1/2}(\nu\pi)^{-d/2}}{\Gamma(\nu/2)[1 + \nu^{-1}\Delta(\vec{X}, \vec{\mu}; \Sigma)]^{(\nu+d)/2}} \tag{1}$$

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

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

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

The asymmetry is added by introducing two different covariance matrices in the parameters of the distribution: covariance on the left Σ_l and covariance on the right Σ_r. So, if we let $\mathcal{S}$ be the multivariate asymmetric Student's t-density function given the parameters $\theta = \{\vec{\mu}, \Sigma_l, \Sigma_r, \nu\}$, $\mathcal{S}$ will be defined as follows:

$$\mathcal{S}(\vec{X}|\theta_k) = \begin{cases} s(\vec{X}|\vec{\mu}, \Sigma_l, \nu) & \text{if } \vec{X} < \vec{\mu} \\ s(\vec{X}|\vec{\mu}, \Sigma_r, \nu) & \text{if } \vec{X} \geq \vec{\mu} \end{cases} \tag{4}$$

Let P be the multivariate asymmetric Student's-t probability density function with bounded support. We define a support region Ω for the distribution, and an indicator function as:

$$\chi(\vec{X}|\Omega) = \begin{cases} 1 & \text{if } \vec{X} \in \Omega \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The bounded support here is defined by multiplying $\mathcal{S}$ by the indicator function χ. The division by the integral of $\mathcal{S}$ over the support region Ω normalizes the bounded function P by the share of $\mathcal{S}(\vec{X}|\vec{\mu}, \Sigma_l, \Sigma_r, \nu)$ that belongs to the support region Ω for the k^{th} distribution [19]. Then, P is defined as follows:

$$P(\vec{X}|\theta) = \frac{\chi(\vec{X}|\Omega) \times \mathcal{S}(\vec{X}|\vec{\mu}, \Sigma_l, \Sigma_r, \nu)}{\int_\Omega \mathcal{S}(\vec{Y}|\vec{\mu}, \Sigma_l, \Sigma_r, \nu)\, d\vec{Y}} \tag{6}$$

where $\theta = \{\vec{\mu}, \Sigma_l, \Sigma_r, \nu, \Omega\}$ denotes the set of distribution parameters.

2.1.2 Relation to the Multivariate Gaussian Distribution

According to [58, 59], the multivariate t-distribution is conditionally related to the normal distribution: if the random multivariate variable $\vec{x}$ follows multivariate t-distribution with a mean $\vec{\mu}$, a covariance matrix Σ, and ν degrees of freedom, then given a precision parameter ϕ, $\vec{X}$ follows a multivariate Gaussian distribution n with mean $\vec{\mu}$ and covariance $\frac{\Sigma}{\phi}$ and where the parameter ϕ is a Gamma-distributed variable with both scale and shape parameters equal to $\frac{\nu}{2}$: $\phi \sim \mathcal{G}(\frac{\nu}{2}, \frac{\nu}{2})$ (See equation (7)).

$$\vec{X} \sim s(\vec{\mu}, \Sigma, \nu) \iff \vec{X}|\phi \sim n(\vec{\mu}, \frac{\Sigma}{\phi}) \text{ and } \phi \sim \mathcal{G}(\frac{\nu}{2}, \frac{\nu}{2}) \tag{7}$$

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 equation (8).

$$s(\vec{X}|\vec{\mu}, \Sigma, \nu) = n\Big(\vec{X}|\vec{\mu}, \frac{\Sigma}{\phi}\Big) \times \mathcal{G}(\phi) \tag{8}$$

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

$$\mathcal{G}(\phi) = \frac{(\frac{\phi\nu}{2})^{\frac{\nu}{2}} \exp{(\frac{-\phi\nu}{2})}}{\phi\Gamma(\frac{\nu}{2})} \tag{9}$$

As for the multivariate Gaussian distribution with a mean vector $\vec{\mu}$ and a covariance matrix Σ, the probability density function is:

$$n(\vec{X}|\vec{\mu}, \Sigma) = \frac{\exp\Big(-\frac{1}{2}(\vec{X} - \vec{\mu})^T\Sigma^{-1}(\vec{X} - \vec{\mu})\Big)}{\sqrt{(2\pi)^k|\Sigma|}} \tag{10}$$

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 χ (see equation (5)) and divide it by the integral over the bounded support region Ω. This yields another definition of the multivariate bounded asymmetric t-distribution P:

$$P(\vec{X}|\theta) = \frac{\mathcal{N}\Big(\vec{X}|\vec{\mu}, \frac{\Sigma_l}{\phi}, \frac{\Sigma_r}{\phi}\Big) \times \mathcal{G}(\phi) \times \chi(\vec{X}, \Omega)}{\int_\Omega \mathcal{N}(\vec{Y}|\vec{\mu}, \frac{\Sigma_l}{\phi}, \frac{\Sigma_r}{\phi}) \times \mathcal{G}(\phi)\, d\vec{Y}} \tag{11}$$

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 section 2.1.1 for the multivariate asymmetric t:

$$\mathcal{N}(\vec{X}|\vec{\mu},\Sigma_l,\Sigma_r) = \begin{cases} n(\vec{X}|\vec{\mu},\Sigma_l) & \text{if } \vec{X} \leq 0 \\ n(\vec{X}|\vec{\mu},\Sigma_r) & \text{otherwise} \end{cases} \tag{12}$$

2.1.3 Bounded Asymmetric Student's-t Mixture Model (BASMM)

In many applications, the data is very complex and does not fit in one simple probability distribution. For instance, it might be multimodal. This means that there are several different modes, or regions of high probability mass, and regions of smaller probability mass in between. In this case, it is rigorous to model the data with a mixture model, i.e., a weighted mixture of components, where each component is a parametric probability distribution (Gaussian, Bernouilli, etc.). For example, Fig. 2.3 shows a histogram plot of what a BASMM-shaped population looks like. This model has three different mixture components and is plotted for univariate data for the ease of display. The behaviour of the BASMM is the same as in Fig. 2.3 for each dimension of a multivariate population.

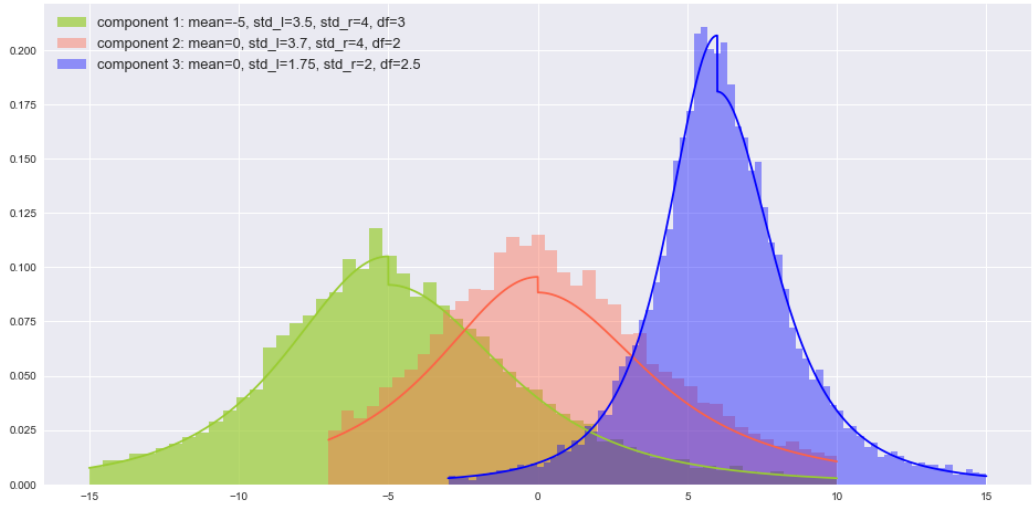

Figure 2.3: Example of a population distributed as a bounded asymmetric t-mixture

This section lays out the mathematical definition for the multivariate bounded asymmetric

Student's-t mixture model, as well as its learning process.

Let $\mathcal{X} = \{\vec{X_1}, \ldots, \vec{X_N}\}$ denotes an observed sample of N multivariate vectors of dimension d each. Modeling $\mathcal{X}$ as Student's-t mixture with K components implies that for every vector $\vec{X_i} = [x_{i1}, \ldots, x_{id}]$, the marginal probability distribution of $\vec{X}_i$ is written as follows:

$$f(\vec{X_i}|\Theta) = \sum_{k=1}^{K} \pi_k P(\vec{X_i}|\theta_k) \tag{13}$$

where π_k and θ_k are respectively the mixing proportion and the set of parameters for the k^{th} mixture component, and finally $\Theta = \{\theta_1, \ldots, \theta_K; \pi_1, \ldots, \pi_K\}$. π_k is the mixing proportion and represents the prior probability that x_i belongs to the k^{th} component, thus satisfies:

$$\pi_k \geq 0 \quad \text{and} \quad \sum_{k=1}^{K} \pi_k = 1 \tag{14}$$

It is worth mentioning that when $\Sigma_{k,l} = \Sigma_{k,r}$ for $k \in \{1, \ldots, K\}$, we get the BSMM, bounded symmetric Student's t-mixture model as a special case of our algorithm. Also if Ω_k is infinite for $k \in \{1, \ldots, K\}$, our model becomes an unbounded asymmetric Student's-t mixture model. With both these conditions combined, we get the regular SMM: Student's t-mixture model [4]. This shows the generality and robustness of BASMM.

2.1.4 Fitting the Mixture Model

Expectation step

Now that we defined the base distribution for the BASMM, we proceed to the expectation step of the EM algorithm [60, 61]. Here, at the iteration t of the EM algorithm, we define by τ_{ik} the posterior probability that the vector $\vec{X_i}$ belongs to the k^{th} component for $i \in \{1, \ldots, N\}$ and $k \in \{1, \ldots, K\}$. These posterior probabilities are calculated as follows:

$$\tau_{ik}^{(t)} = \frac{\pi_k^{(t)} P(\vec{X_i}|\theta_k^{(t)})}{\sum_{j=1}^{K} \pi_j^{(t)} P(\vec{X_i}|\theta_j^{(t)})} \tag{15}$$

The estimation step includes also calculating the log-likelihood of the model $\mathcal{L}(\mathcal{X}|\Theta)$ at the current iteration t:

$$\mathcal{L}(\mathcal{X}|\Theta) = \log\Big(\prod_{i=1}^{N} f(\vec{X}_i|\Theta)\Big) = \sum_{i=1}^{N} \log\Big(\sum_{k=1}^{K} \pi_k P(\vec{X}_i|\theta_k)\Big) \tag{16}$$

where $\Theta = \{\theta_1, \ldots, \theta_K; \pi_1, \ldots, \pi_K\}$ and $\theta_k = \{\vec{\mu}_k, \Sigma_{k,l}, \Sigma_{k,r}, \nu_k\}$ for $1 \leq k \leq K$.

Maximization step

The goal of the maximization step in the EM algorithm is to update the model parameters in a way that maximizes the previously calculated log-likelihood function [62]. As the logarithm is monotonically increasing, it is more suitable to minimize the negative log-likelihood function $\mathcal{J}(\mathcal{X}|\Theta) = -\mathcal{L}(\mathcal{X}|\Theta)$.

By applying the Jensen inequality, we find that at the t^{th} iteration:

$$\mathcal{J}(\mathcal{X}|\Theta) \leq -\sum_{i=1}^{N}\sum_{k=1}^{K} \tau_{ik}^{(t)} \Big[\log \pi_k + \log \mathcal{S}(\vec{X}_i|\theta_k) - \log \int_{\Omega_k} \mathcal{S}(\vec{Y}|\theta_k)\, d\vec{Y}\Big] \tag{17}$$

Thus, minimizing $\mathcal{J}(X|\Theta)$ becomes equivalent to minimizing $\mathcal{E}(X|\Theta)$, where:

$$\mathcal{E}(\mathcal{X}|\Theta) = -\sum_{i=1}^{N}\sum_{k=1}^{K} \tau_{ik}^{(t)} \Big[\log \pi_k + \log \mathcal{S}(\vec{X}_i|\theta_k) - \log \int_{\Omega_k} \mathcal{S}(\vec{Y}|\theta_k)\, d\vec{Y}\Big] \tag{18}$$

In this case, $\mathcal{E}(\mathcal{X}|\Theta)$ is regarded as an error function that needs to be minimized [19] to obtain an optimal fit to the data. Therefore, in each iteration t, the updates for the different parameters of the BASMM are calculated in a way that minimizes $\mathcal{E}(\mathcal{X}|\Theta)$:

$$\Theta^{(t+1)} = \arg\min_{\Theta}(\mathcal{E}(\mathcal{X}|\Theta)) \tag{19}$$

Mean vector estimation The update of the mean vector of the k^{th} mixture component is the solution to the equation:

$$\frac{\partial \mathcal{E}(\mathcal{X}|\Theta)}{\partial \vec{\mu}_k} = 0 \tag{20}$$

This yields the following definition for $\vec{\mu_k}^{(t+1)}$:

$$\vec{\mu}_k^{(t+1)} = \frac{\sum_{i=1}^{N} \tau_{ik}^{(t)} (h(\vec{X}_i|\vec{\mu}_k^{(t)}, \Sigma_{k,l}^{(t)}, \nu_k^{(t)})\vec{X}_i - \mathcal{A}_k)}{\sum_{i=1}^{N} \tau_{ik}^{(t)} h(\vec{X}_i|\vec{\mu}_k^{(t)}, \Sigma_{k,l}^{(t)}, v_k^{(t)})} \tag{21}$$

where:

$$h(\vec{X}|\vec{\mu}, \Sigma, \nu) = \frac{\nu + d}{\nu + (\vec{X} - \vec{\mu})^T \Sigma^{-1} (\vec{X} - \vec{\mu})} \tag{22}$$

and where:

$$\mathcal{A}_k = \frac{\sum_{m=1}^{M} (\vec{y}_{km} - \vec{\mu}_k^{(t)}) \times h(\vec{y}_{km}|\vec{\mu}_k^{(t)}, \Sigma_{k,l}^{(t)}, \nu_k^{(t)}) \times \chi(\vec{y}_{km}|\Omega_k)}{\sum_{m=1}^{M} \vec{y}_{km}|\Omega_k)} \tag{23}$$

where $(\vec{y}_{km})_{m=1}^{M}$ is a generated sample of M vectors from the asymmetric Student's-t distribution S with the parameters $\theta_k = \{\vec{\mu}_k, \Sigma_{k,l}, \Sigma_{k,r}, \nu_k\}$, where M is an integer chosen large enough to approximate the integral of the asymmetric t-density function.

Note that in (21), (23), and (22), either the left or the right covariance matrix can be used, as both of them are among the parameters of two symmetric t-distributions with the same mean $\vec{\mu}_k^{(t)}$ and degrees of freedom $\nu_k^{(t)}$.

Left covariance estimation Following the same logic, the update of the left covariance matrix of the k^{th} mixture component is the solution to the equation:

$$\frac{\partial \mathcal{E}(\mathcal{X}|\Theta)}{\partial \Sigma_{lk}} = 0 \tag{24}$$

After calculations, $\Sigma_{kl}^{(t+1)}$ has the following definition:

$$\Sigma_{kl}^{(t+1)} = \frac{\sum_{i=1}^{N} \tau_{ik}^{(t)} h(\vec{X}_i|\vec{\mu}_k^{(t)}, \Sigma_{k,l}^{(t)}, \nu_k^{(t)})(\vec{X}_i - \vec{\mu}_k^{(t)})(\vec{X}_i - \vec{\mu}_k^{(t)})^T}{\sum_{i=1}^{N} \tau_{ik}^{(t)}} - \mathcal{B}_{kl} \tag{25}$$

where:

$$\begin{aligned} \mathcal{B}_{kl} = \frac{1}{\sum_{m=1}^{M} \chi(\vec{y}_{km}|\Omega_k)} \times \sum_{m=1}^{M} \Big(\Sigma_{kl}^{(t)} - (\vec{y}_{km} - \vec{\mu}_k^{(t)}) \\ \times (\vec{y}_{km} - \vec{\mu}_k^{(t)})^T h(\vec{y}_{km}|\vec{\mu}_k^{(t)}, \Sigma_{kl}^{(t)}, \nu_k^{(t)}) \Big) \chi(\vec{y}_{km}|\Omega_k) \end{aligned} \tag{26}$$

Right covariance estimation Following the same logic, the update of the right covariance matrix of the k^{th} mixture component is the solution to the equation:

$$\frac{\partial \mathcal{E}(\mathcal{X}|\Theta)}{\partial \Sigma_{rk}} = 0 \tag{27}$$

After calculations, $\Sigma_{kr}^{(t+1)}$ has the following definition:

$$\Sigma_{kr}^{(t+1)} = \frac{\sum_{i=1}^{N} \tau_{ik}^{(t)} h(\vec{X}_i|\vec{\mu}_k^{(t)}, \Sigma_{k,r}^{(t)}, \nu_k^{(t)})(\vec{X}_i - \vec{\mu}_k^{(t)})(\vec{X}_i - \vec{\mu}_k^{(t)})^T}{\sum_{i=1}^{N} \tau_{ik}^{(t)}} - \mathcal{B}_{kr} \tag{28}$$

where:

$$\begin{aligned} \mathcal{B}_{kr} = \frac{1}{\sum_{m=1}^{M} \chi(\vec{y}_{km}|\Omega_k)} \times \sum_{m=1}^{M} \Big(\Sigma_{kr}^{(t)} - (\vec{y}_{km} - \vec{\mu}_k^{(t)}) \\ \times (\vec{y}_{km} - \vec{\mu}_k^{(t)})^T h(\vec{y}_{km}|\vec{\mu}_k^{(t)}, \Sigma_{kr}^{(t)}, \nu_k^{(t)}) \Big) \chi(\vec{y}_{km}|\Omega_k) \end{aligned} \tag{29}$$

Degrees of freedom estimation Finally, the update of the degrees of freedom for the k^{th} component $\nu_k^{(t+1)}$ is a solution to the equation:

$$\frac{\partial \mathcal{E}(\mathcal{X}|\Theta)}{\partial \Sigma_{rk}} = 0 \tag{30}$$

which is equivalent to:

$$\begin{aligned} \frac{1}{\sum_{i=1}^{N} \tau_{ik}^{(t)}} \times \sum_{i=1}^{N} \tau_{ik}^{(t)} \delta(\vec{X}_i|\vec{\mu}_k, \Sigma_k, \nu_k) - \psi(\frac{\nu_k}{2}) \\ + \psi(\frac{\nu_k + D}{2}) + \log(\frac{\nu_k}{\nu_k + D}) + 1 - \mathcal{C}_k = 0 \end{aligned} \tag{31}$$

where:

$$\delta(\vec{X}|\vec{\mu}, \Sigma, \nu) = \log(h(\vec{X}|\vec{\mu}, \Sigma, \nu)) - h(\vec{X}|\vec{\mu}, \Sigma, \nu) \tag{32}$$

and where:

$$\mathcal{C}_k = \sum_{m=1}^{M} \Big[\Big(\delta(\vec{y}_{km}|\vec{\mu}_k^{(t)}, \Sigma_k^{(t)}, \nu_k^{(t)}) - \psi(\frac{\nu_k}{2}) + \psi(\frac{\nu_k + D}{2}) + \log(\frac{\nu_k}{\nu_k + D}) + 1 \Big) \times \chi(\vec{y}_{km}|\Omega_k) \Big] \times \frac{1}{\sum_{m=1}^{M} \chi(\vec{y}_{km}|\Omega_k)} \tag{33}$$

The equation (31) has no closed-form solution. In this case, we use the Newton Raphson method to calculate an approximation to $\nu_k^{(t+1)}$.

2.2 Model Selection using Minimum Message Length for Bounded Asymmetric Student's-t Mixture Model

As its name suggests, the minimum message length (MML) [29, 30] method is based on compressing a message that contains the data clustered by the evaluated mixture model [3, 63]. The better fit is the model, the greater is its capacity to compress. This approach suggests modeling the observed data $\mathcal{X} = \{\vec{X_1}, \ldots, \vec{X_N}\}$ by a mixture of distributions with different numbers of components K, and calculating the message length (i.e., the amount of measured information after data compression) for each value of K. Then, the mixture that has the optimal number of clusters (K_{opt}) is the one that scores the minimum message length:

$$K_{opt} = \arg\min_{K}(MessLen(K)) \tag{34}$$

where $MessLen$ denotes the message length for a BASMM with K mixture components and a set of parameters Θ_K. It is defined as follows:

$$MessLen(K) \simeq -\log(p(\Theta_K)) - \mathcal{L}(\mathcal{X}|\Theta_K) + \frac{1}{2}\log|F(\Theta_K)| + \frac{N_p}{2}(1 + \log(\frac{1}{12})) \tag{35}$$

where $p(\Theta_K)$ is the prior probability, $\mathcal{L}(\mathcal{X}|\Theta_K)$ is the log-likelihood of the BASMM (defined in the equation (16)), $|F(\Theta_K)|$ is the determinant of the mixture model's Fisher information matrix, and N_p is the number of free parameters in the mixture model. In the case of BASMM, $N_p = K(d^2 + 2d + 2)$. The prior probability and the Fisher information matrix of the BASMM will be

calculated in the following paragraphs.

2.2.1 Fisher Information Matrix Calculation

The Fisher matrix is used in the message length formula to measure the amount of information contained in the evaluated mixture model. For a random multivariate vector $\vec{X}$ that follows a distribution f around a parameter θ, the Fisher information $F(\theta)$ describes how sensitive f is to changes in the parameter θ [63].

The Fisher information matrix is the expected value of the Hessian matrix, and for a multivariate mixture model, calculating it can be a complex task. To get over this difficulty, we approximate $F(\Theta_K)$ by the the block diagonal of the complete data Fisher information matrices of the separate mixture components [28, 3, 30, 64]. Following this approximation, the determinant of the Fisher information matrix $|F(\Theta_K)|$ is the product of all information matrices for all mixture components. This yields the following definition of $|F(\Theta_K)|$:

$$|F(\Theta_K)| \simeq |F(\pi)| \prod_{k=1}^{K} |F(\vec{\mu}_k)||F(\Sigma_{lk})||F(\Sigma_{rk})||F(\nu_k)| \tag{36}$$

where for $k \in \{1, \ldots, K\}$:

$$|F(\pi)| = \frac{N^{K-1}}{\sum_{k=1}^{K} \pi_k} \tag{37}$$

$$F(\vec{\mu}_k) = \frac{\partial^2 \mathcal{L}(\mathcal{X}|\theta_K)}{\partial^2 \vec{\mu}_k} \tag{38}$$

$$F(\Sigma_{kl}) = \frac{\partial^2 \mathcal{L}(\mathcal{X}|\theta_K)}{\partial^2 \Sigma_{kl}} \tag{39}$$

$$F(\Sigma_{kr}) = \frac{\partial^2 \mathcal{L}(\mathcal{X}|\theta_K)}{\partial^2 \Sigma_{kr}} \tag{40}$$

$$F(\nu_k) = \frac{\partial^2 \mathcal{L}(\mathcal{X}|\theta_K)}{\partial^2 \nu_k} \tag{41}$$

The complete derivation of the Fisher information for the different BASMM parameters is detailed in appendix .1.

2.2.2 Prior Probability Calculation

Assuming that all the parameters of the BASMM are independent from each other, the mixture model's prior probability will be in the following format:

$$p(\Theta_K) = p(\vec{\mu})p(\Sigma_l)p(\Sigma_r)p(\nu)p(\pi) \tag{42}$$

where $p(\vec{\mu})$, $p(\Sigma_l)$, $p(\Sigma_r)$, $p(\nu)$, and $p(\pi)$ represent the prior probability weights corresponding to the parameters $\vec{\mu}$, Σ_l, Σ_r, ν and π, respectively.

Starting with the $p(\pi)$, knowing that the mixing weights are defined on the simplex $\{(\pi_1, \ldots, \pi_k) : \sum_{k=1}^{K} \pi_k = 1\}$, we can say that $p(\pi)$ follows a Dirichlet distribution [65]:

$$p(\pi) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \times \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \tag{43}$$

where: $\alpha_1 = \alpha_2 = \cdots = \alpha_K = 1$. Setting these parameters to 1 in the Dirichlet distribution leads to having:

$$p(\pi) = \Gamma(K) = (K-1)! \tag{44}$$

For the mean parameter $\vec{\mu}$, we assume that all prior means follow a uniform distribution within Σ_{all} from the mean of the whole population $\vec{\mu}_{all}$. So, we have:

$$p(\vec{\mu}) = \prod_{k=1}^{K} p(\vec{\mu}_k) = \prod_{k=1}^{K} \prod_{i=1}^{d} \frac{1}{2\sigma_i} = \prod_{i=1}^{d} \frac{1}{(2\sigma_i)^K} \tag{45}$$

For the left and right covariance matrices, we have:

$$p(\Sigma) = \prod_{k=1}^{K} p(\Sigma_k) \tag{46}$$

In the light of lacking prior information about the mixture parameters, we take $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_K = \Sigma_{all}$, where Σ_{all} is the covariance matrix of the whole population. Let $\sigma_1, \sigma_2, \ldots, \sigma_d$ be the respective variances of each dimension of the population. These variances are independent from

each other. Following this, we get:

$$p(\Sigma_l) = \prod_{k=1}^{K} p(\Sigma_{l,all}) = \prod_{k=1}^{K} \prod_{i=1}^{d} \frac{1}{\sigma_{li}} = \prod_{i=1}^{d} \frac{1}{\sigma_{li}^K} \tag{47}$$

$$p(\Sigma_r) = \prod_{k=1}^{K} p(\Sigma_{r,all}) = \prod_{k=1}^{K} \prod_{i=1}^{d} \frac{1}{\sigma_{ri}} = \prod_{i=1}^{d} \frac{1}{\sigma_{ri}^K} \tag{48}$$

Regarding the prior degrees of freedom $p(\nu)$, we assume again the uniform distribution $\mathcal{U}[0, h]$ for these parameters, where h is chosen to be sufficiently large. So, we get the following:

$$p(\nu) = \prod_{k=1}^{K} p(\nu_k) = \prod_{k=1}^{K} \frac{1}{h} = \frac{1}{h^K} \tag{49}$$

Combining the above results, we get the following equation for the prior information $p(\Theta)$:

$$p(\Theta) = \frac{(K-1)!}{2^{dK} h^K} \times \prod_{i=1}^{d} \frac{1}{\sigma_i^{2K}} \tag{50}$$

2.3 Experiments and Results

In this section, we test the BASMM's performance as well as the MML model selection through three different experiments: clustering hourly energy consumption profiles at households, head pose estimation from driving faces images, and leukemia detection from genetic expression data. Every experiment includes a clustering phase and a model selection phase, and the model selection is performed for multiple data scenarios in order to validate the MML's performance. In the following paragraphs, we present the clustering performance metrics and model selection criteria, as well as the benchmark algorithms for comparison. Then, we proceed to the experiments.

2.3.1 Clustering Metrics

In order to evaluate the model selection along with the EM algorithm for our mixture model, we selected a few metrics and reference algorithms to be compared with our BASMM + MML. The evaluation is made in two main steps:

(1) Evaluate the model selection process by comparing MML and other techniques.

(2) For a fixed number of clusters, evaluate the clustering by comparing between BASMM and other algorithms

The reference clustering methods used for comparison are:

- Gaussian mixture model (GMM)
- Bounded asymmetric Gaussian mixture model (BAGMM)
- Student mixture model (SMM)
- Bounded student mixture model (BSMM)

Here is a breakdown of the metrics that will be used for the evaluation:

- **Dunn index:** The Dunn index is an internal clustering validation measure, introduced by J.C. Dunn [66]. Let us denote by d_{min} the minimal distance between points of different clusters and d_{max} the largest distance between 2 points within the same cluster. The Dunn index is defined as the ratio of d_{min} to d_{max}:

$$dunn = \frac{d_{min}}{d_{max}}$$

  For a given assignment of clusters, a higher Dunn index indicates a better clustering.

- **Silhouette score:** This is another technique used to measure the goodness of clustering, its value ranges from -1 to 1. a silhouette score of 1 means clusters are well apart from each other and well distinguished, while a score of 0 indicates that clusters are indifferent, or we can say that the distance between clusters is not significant. -1 Means clusters are assigned in the wrong way. We use the silhouette score to compare the model selection techniques, and find which number of clusters gives the best distinction between them.

- **Calinski-Harabasz score:** Also known as the variance ratio criterion, the Calinski-Harabasz score is intended to measure how dense and well separated clusters are. Mathematically, it is

the ratio of the sum of between-clusters dispersion and inter-cluster dispersion for all clusters. Higher score values indicate better clustering.

- **Specificity score:** This metric quantifies the model's ability to avoid false positives. For experiments that have ground truth labels, the specificity score is easily obtainable from the confusion matrix and, it can be averaged on the different clusters. The specificity score SP is calculated as follows:

$$Specificity = \frac{TN}{TN + FP}$$

where TN is the number of true negatives and FP is the number of false positives. This score ranges from 0 to 1, and the closer it is to 1, the better the model's performance.

- **Sensitivity score:** As opposed to the specificity score, the sensitivity score calculates the model's ability to avoid false negatives. It is defined by the following ratio:

$$Sensitivity = \frac{TP}{TP + FN}$$

where TP is the number of true positives and FN is the number of false negatives. This score ranges from 0 to 1, and the closer it is to 1, the better the model's performance.

- **Davies-Bouldin index score:** This score is defined as the average similarity measure of each cluster with its most similar cluster, where similarity is the ratio of within-cluster distances to between-cluster distances. Thus, clusters that are farther apart and less dispersed will result in a better score. The minimum score is 0, with lower values indicating better clustering.

- **Matthews correlation coefficient (MCC):** MCC [67] takes into account true and false positives and negatives and is generally regarded as a balanced measure that can be used when the data is imbalanced. The MCC is, in essence, a correlation coefficient value between -1 and $+1$. A coefficient of $+1$ represents a perfect prediction, 0 an average random prediction and -1 an inverse prediction. The statistic is also known as the phi coefficient.

2.3.2 Model Selection Criteria

For the model selection, we use MML as well as a list of other popular criteria to estimate the right number of clusters in every experiment. To establish a comparison, we use AIC, BIC, MDL, and MMDL, which are defined as follows, respectively:

$$AIC(K) = -\mathcal{L}(X|\Theta_K) + \frac{N_p}{2} \tag{51}$$

$$BIC(K) = -2\mathcal{L}(X|\Theta_K) + N_p \log N \tag{52}$$

$$MDL(K) = -\mathcal{L}(X|\Theta_K) + \frac{N_p}{2} \log N \tag{53}$$

$$MMDL(K) = -2\mathcal{L}(X|\Theta_K) + \frac{N_p}{2} \log N + \frac{c}{2} \sum_{j=1}^{K} \log \pi_j \tag{54}$$

where in the equation (54), c is the number of parameters in each mixture component, which equals $(d^2 + 2d + 2)$ in the case of our model.

2.3.3 Experiment 1: Clustering Hourly Energy Consumption Profiles at Households

In the field of smart buildings, the discovery and analysis of electricity consumption patterns can provide valuable insights for energy companies and city management organisms. These entities can use the electricity consumption data (ECD) for many purposes, namely to monitor energy consumption, target high demand with a better supply, improve energy utilization efficiency, etc. In this regard, clustering is very useful for detecting electricity consumption profiles (ECP) from ECD, and has been employed in many research works [68, 69, 70]. In this experiment, we attempt to cluster the daily profiles of electricity consumption in households using BASMM. We also perform the model selection with MML criterion to determine the optimal number of ECPs.

Data and Preprocessing

For this experiment, we use UCI's individual household electric power consumption data [71]. It contains 2075259 measurements gathered between December 2006 and November 2010 (47 months). These measurements are taken every minute, which makes the data quite heavy. For

ease of manipulation, we preprocess the data by only keeping the hourly information, so we obtain 24 variables per day for 47 months. We also pivot the dataset by transforming it from a sequence of measures into vectors of a day's hourly measures of electricity. Finally, after performing min-max scaling, the preprocessed data is composed of 1456 vectors of dimension $d = 24$ each. Fig. 2.4 shows a plot of all the hourly measures for every day (vector) in the dataset.

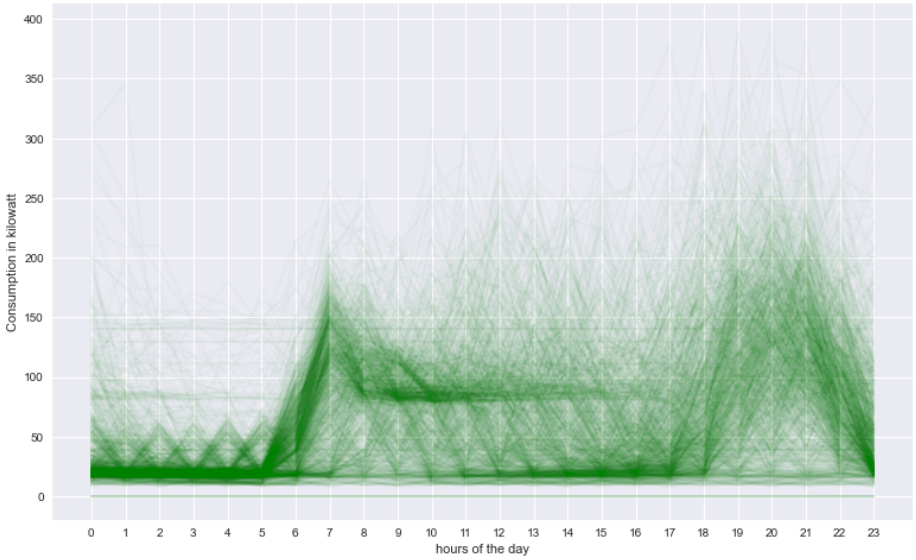

Figure 2.4: Household electricity consumption

Clustering

According to Fig. 2.4, we can notice two distinct patterns/profiles of electricity consumption. One that spikes in the morning hours of the day (peak around 7 am), and another one that presents a peak of consumption in the late night. This provide us an insight about the trends of this household consumption. For example, there are days where more electric appliances are used in the morning (laundry days), while other days present more consumption at prime time (watching television or gaming).

This experiment is purely unsupervised, as we do not have ground truth information or labels beforehand to validate the clustering. Hence, we identify the best clustering and the optimal number of clusters (ECPs) based on different clustering validity indices [72]. We perform the clustering on the preprocessed data by using BASMM with different numbers of mixture components, and based

on the silhouette analysis, $K = 2$ is the optimal number of ECPs (see Fig. 2.5).

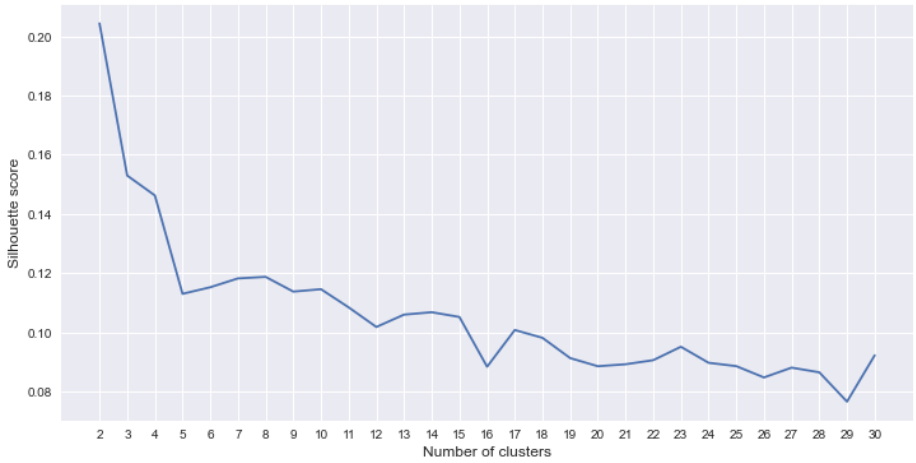

Figure 2.5: Silhouette score for different numbers of mixture components (BASMM clustering)

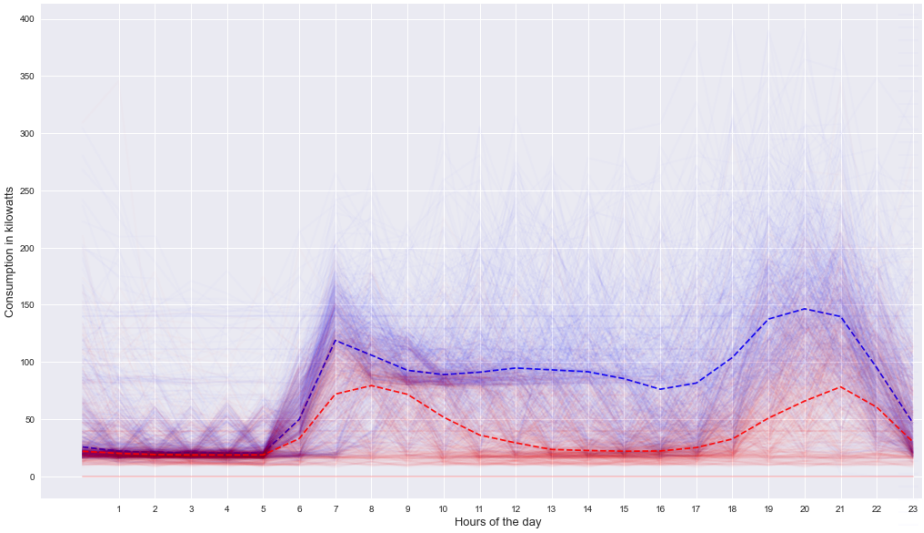

Figure 2.6: Household electricity consumption: BASMM clustering with two ECPs

Results

We run the BASMM, SMM, BAGMM, and GMM algorithms on the data using a k-means initialization and two mixture components. We iterate the EM algorithm for all the models until we reach a stable log-likelihood at less than $\pm 10^{-6}$. Fig. 2.6 shows the two ECPs produced by the BASMM clustering, and Table 2.1 presents the findings in terms of clustering performance metrics for BASMM and the rest of the models.

Table 2.1: Results and comparisons: electricity consumption profiles clustering

Model	Dunn	Silhouette	Davies-Bouldin	Calinski-Harabasz
BASMM	0.089	0.21	1.860	343.762
SMM	0.091	0.161	2.095	293.430
BAGMM	0.061	0.101	2.392	206.796
GMM	0.068	0.09	2.573	183.024

According to Table 2.1, BASMM presents a superior performance compared to the rest of the algorithms. In regards to cluster separation, BASMM scored the best Calinski-Harabasz (343.762) and Davies-Bouldin (1.86) scores among the four mixture models used for the experiment. BASMM has also produced the best silhouette score at 0.2 and the second best Dunn index at 0.089. Overall, the four mixture models generated close clustering results, with Gaussian-based models performing slightly less than Student-based models. As we notice from Fig. 2.4, outliers are present in the form of days with sporadic peaks of electricity consumption (mainly in the afternoon). The higher clustering validity scores produced by BASMM prove a better absorption of these outliers, as despite their presence, the ECPs are better separated and defined. The Dunn index values are low in general (ranging from 0.068 to 0.091). This is explained by the nature of the data, as the daily trends of electricity consumption are similar in value ranges. This translates into a relatively small variance between ECPs.

Model Selection

We perform the model selection on top of BASMM for three different sets of two-folds of the data: A, B, and C, which gives us six different subsets to experiment on: $A1$, $A2$, $B1$, $B2$, $C1$, $C2$. The results are the numbers of ECPs determined by model selection criteria, presented in Table 2.2.

According to Table 2.2, most of the model selection criteria have converged on determining two ECPs (mixture components) for all the data folds, with the exception of some cases where three ECPs were determined. We notice that MML has consistently elected two clusters in all samples, MDL and MMDL made a nearly similar decision, and AIC/BIC have mostly determined three ECPs.

Table 2.2: Number of ECPs determined by different model selection criteria

Data	K[1]	Model selection criteria				
		MML	MDL	MMDL	AIC	BIC
$A1$	2	2	2	2	3	2
$A2$	2	2	2	2	2	3
$B1$	2	2	3	2	2	3
$B2$	2	2	2	2	3	3
$C2$	2	2	2	2	3	2
$C2$	2	2	3	3	2	2

[1]Real number of ECPs

2.3.4 Experiment 2: Head Pose Estimation From Driving Faces Images

Upper body and head pose estimation is an important task for monitoring human behaviour and attention during activities such as driving. In this experiment, we cluster images of faces of people in the driving seat of a car. For this, we use the DrivFace dataset [73] which contains 606 images of four different individuals (two men and two women) in the driver's seat of a car. From the data, we are looking to learn the directions at which the driver is looking and they are mainly three: looking left, looking right, and looking to the front.

Preprocessing

For this experiment, the preprocessing consists of extracting meaningful features from the 2D images that constitute the dataset. We start by converting the images to grayscale and cropping the region of interest which contains the face of the driver in the image to eliminate unnecessary information. We then proceed to extract the keypoint descriptors of the cropped image. SIFT (Scale Invariant Feature Transform) is a popular method that achieves this extraction, but in the case of head pose estimation, it has limited performance. In fact, face images do not have many textures, thus SIFT fails to produce enough descriptors for the detailed landmarks of the face. As an alternative, Dense SIFT (DSIFT) [74] collects more features at each location and scale in an image, increasing recognition accuracy accordingly. Therefore, we use DSIFT to detect the keypoint descriptors of the faces in our dataset. A Gabor filter [75] is also used to detect more features from the images [76]. The fusion between the flattened DSIFT descriptors and the Gabor features [77] produces a

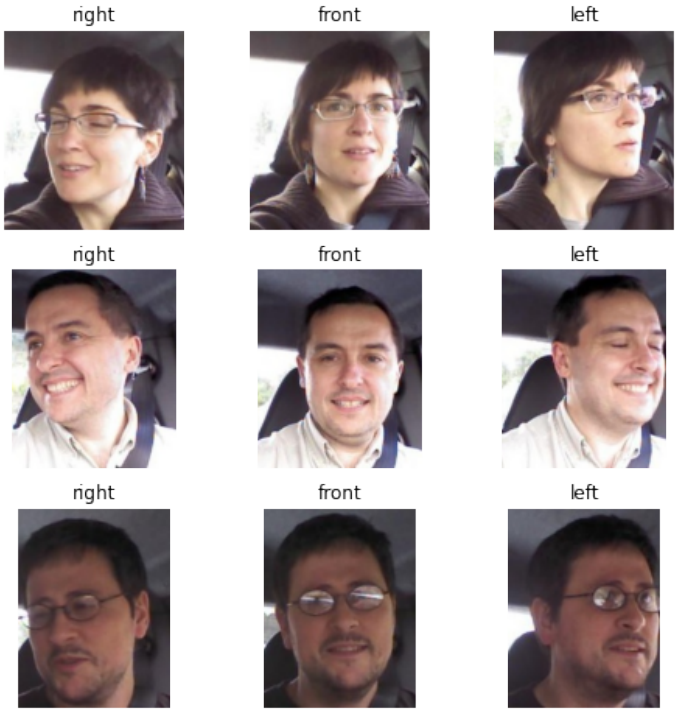

Figure 2.7: Samples of faces looking in different directions

high-dimension feature vector for each image, that is 186368 features.

As the combination of DSIFT keypoint extraction and Gabor filter produces a high-dimensional feature set from the images, we proceed with dimensionality reduction using PCA while keeping a high variance. This gives us a preprocessed dataset of 92 features and 606 instances.

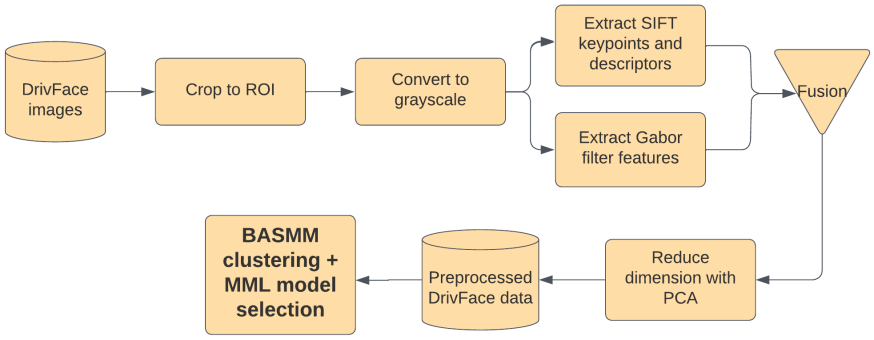

Figure 2.8: Data preprocessing block diagram

Results

We perform the clustering on the preprocessed data using BASMM, SMM, and GMM. We iterate the EM algorithm for all the models until reaching a stable log-likelihood at $\pm 10^{-6}$ or less. The results generated by the different mixture models are presented in Table 2.3

Table 2.3: Results and comparisons: head pose estimation from driving faces

Model	Accuracy	F1 score	Specificity	Sensitivity	MCC
BASMM	0.8679	0.8679	0.9184	0.8679	0.5640
SMM	0.8481	0.8481	0.8724	0.8481	0.5092
GMM	0.83	0.83	0.8418	0.83	0.4550

According to the findings in Table 2.3, all mixture models performed well overall, with accuracies and F1 scores higher than 0.83. However, BASMM stands out and achieves a more accurate estimation than SMM, which in turn performs better than GMM. The robustness of our model is displayed through an accuracy and F1 score of 0.8679, compared to SMM and GMM scoring 0.8481 and 0.83, respectively. We also observe better specificity and sensitivity scored by BASMM (0.9184 and 0.8679, respectively) in comparison to the other two mixture models, and this shows our model's better capacity of finding the optimal boundary between the three classes/clusters in this dataset. This capacity is confirmed by the confusion matrix in Fig. 2.9, where the three head positions were predicted correctly at 81% or more. As for the MCC values, they are overall good and close for all three models, with BASMM having a slight edge at 0.564 compared to SMM and GMM (0.5092 and 0.4550, respectively).

Model Selection

As for the model selection phase, we apply different criteria on top of the BASMM clustering to determine the best number of mixture components. This is performed on four distinct parts of the dataset, containing images of four different drivers:

- $W1$: composed of the images of woman 1
- $W2$: composed of the images of woman 2

- $M1$: composed of the images of man 1
- $M2$: composed of the images of man 2

Table 2.4 illustrates the results of the model selection generated by MML and several other criteria.

Table 2.4: Number of clusters determined by different model selection criteria with different subsets of the dirvers' faces data

Data	K[1]	Model selection criteria				
		MML	MDL	MMDL	AIC	BIC
$W1$	3	3	3	3	3	3
$W2$	3	3	2	3	2	2
$M1$	3	3	3	2	3	3
$M2$	3	3	3	2	2	2

[1]Real number of data clusters

According to Table 2.4, all criteria predicted the number of clusters correctly in most cases. MML is only criterion that selected the exact number of clusters in all datasets, whereas other criteria predicted some incorrect numbers of clusters mostly for $M2$ and $W2$ sets. In all cases of inaccurate model selection, two clusters were determined by these criteria instead of three.

2.3.5 Experiment 3: Leukemia Detection from Genetic Expression

Leukemia is a form of blood cancer that develops when the human body's bone marrow contains too many white blood cells. This medical condition affects adults and is considered a prevalent form of cancer in children. An early diagnosis of Leukemia is very important to start the treatment process for the patient and avoid complications of the disease. In this context, multiple research works [78, 79] focus on analyzing the medical data in its different forms to detect signs of malignant cells that can help doctors get an early diagnosis. In this experiment, we consider the task of clustering the genetic information data to learn the disease that affects each patient, leukemia being one of these diseases. We use the dataset collected from [80] to perform the experiment.

Dataset and Preprocessing

The data at hand is a high dimensional dataset of genetic expression, collected from the bone marrow cells of human patients. It contains 14208 attributes for 12029 records (data points) and is labelled by the type of disease of each patient. The data description is demonstrated in detail in Fig. 2.10.

The diseases present in these records are most predominantly acute myeloid leukemia (AML) with 4573 occurences and acute lymphocytic leukemia with 3764 occurences. The dataset presents also other types of leukemia and diabetes with less prevalence in the data, as there are also 578 healthy records. There are other diseases that represent a very small fraction of the dataset and are not very relevant to our experiment, hence will be dropped in the preprocessing.

The preprocessing here consists of reducing the dimension of the dataset while keeping a high variance, and regrouping the labels (clusters) in a way that helps detect the main diseases of interest in our experiment. First, we reduce the labels by grouping *Diabetes I* and *Diabetes II* together and dropping the data points related to very low-prevalence diseases: *chronic myeloid leukemia (CML)*, *clinically isolated syndrome*, *MDS*, *DS transient myeloproliferative disorder*. Second, we drop the attributes with low variance (below 0.3) and we scale the data using the z-score. Finally, we apply the principal component analysis to further drop the dimensionality. This provides us a preprocessed dataset of 11633 data points and 50 features. The ground truth labels of the preprocessed dataset are presented in Fig. 2.11

Clustering Process and Results

We perform the clustering on the data using the BASMM, the SMM, the BAGMM and the GMM. For seven mixture components, which is the true number of clusters in our experiment, the obtained results for the different algorithms are presented in Table 2.5.

According to the results in Table 2.5, BASMM clustering showed a great performance with an accuracy and an F1 score around the range of 0.92, higher than BAGMM's metrics (0.8833 and 0.8792, respectively). In comparison the SMM and GMM produced a slightly lower accuracy (0.8427 and 0.8289, respectively) and F1 score (0.8334 and 0.8257, respectively). Regarding the

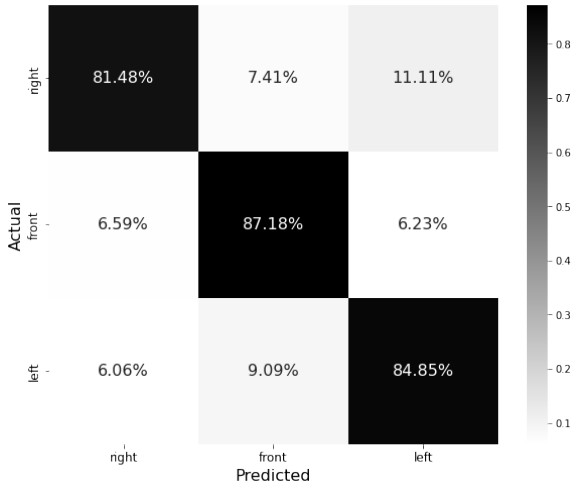

Figure 2.9: Head pose estimation: confusion matrix of BASMM clustering

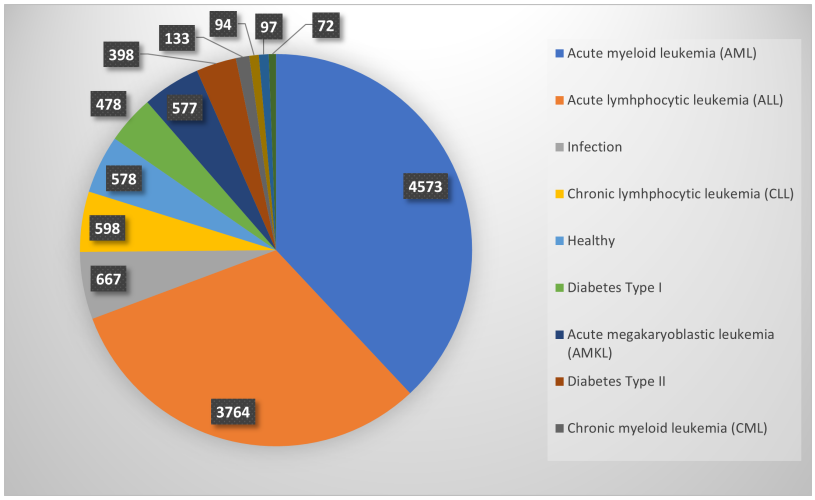

Figure 2.10: Pie chart of different diseases in the dataset

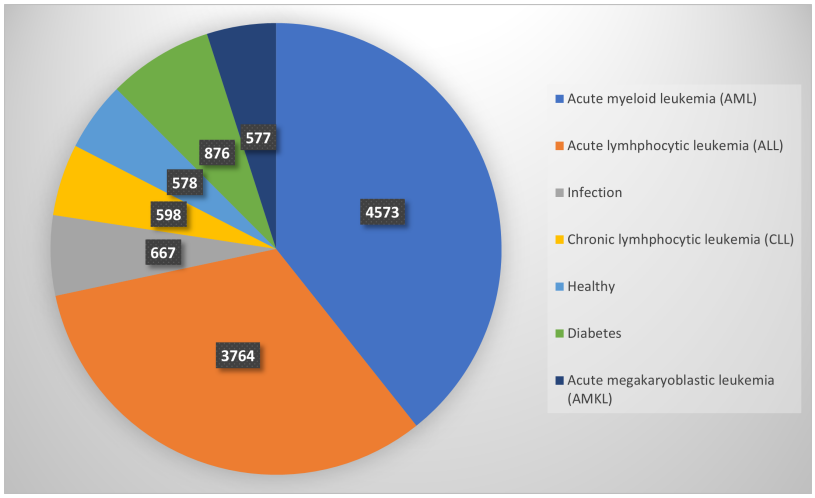

Figure 2.11: Pie chart of different diseases in the dataset after preprocessing

Table 2.5: Results and comparisons: Acute myeloid leukemia detection

Model	Accuracy	F1 score	Specificity	Sensitivity	Davies-Bouldin	MCC
BASMM	0.9255	0.9194	0.9746	0.8823	0.6028	0.8666
BAGMM	0.8833	0.8792	0.9274	0.8367	0.7203	0.8165
SMM	0.8427	0.8334	0.8836	0.7945	0.7569	0.7832
GMM	0.8289	0.8257	0.8647	0.8826	0.7389	0.7102

cluster validity, we observe a slightly smaller Davies-Bouldin index produced by BASMM at around 0.6 in comparison to the rest of the models that generated a Davies-Bouldin index of 0.72 and higher. This indicates a superior cluster separation for BASMM. We also observe that BASMM handled the data imbalance in the best way through the MCC metric. In fact, BASMM also scored the best value, at 0.8666, compared to BAGMM, SMM, AND GMM (which scored 0.8165, 0.7832, and 0.7102, respectively).

Model Selection Results

To determine the optimal number of clusters/mixture components for the dataset at hand, we take multiple subsets of the original data with different numbers of clusters, thus we obtain the following subsets:

- $S1$: complete dataset (7 clusters)

- $S2$: subset composed of the original dataset excluding healthy patients (6 clusters)
- $S3$: subset composed of the original dataset excluding diabetes and infection patients (5 clusters)
- $S4$: subset composed of data from AML, ALL, CLL, and AMKL patients (4 clusters)
- $S5$: subset composed of data from AML, ALL, and CLL patients (3 clusters)

We apply the MML model selection criterion on the different clustering models. To validate the MML's performance, we use other criteria and we compare their findings with the MML findings. The results presented in Table 2.6 demonstrate the MML's precision in predicting the correct number

Table 2.6: Number of clusters determined by different model selection criteria with different subsets of the data

Data	K[1]	Model selection criteria				
		MML	MDL	MMDL	AIC	BIC
$S1$	7	7	6	7	7	6
$S2$	6	6	7	7	6	6
$S3$	5	5	5	6	5	5
$S4$	4	4	4	4	3	3
$S5$	3	3	4	3	3	3

[1]Real number of data clusters

of clusters for all the different subsets. In comparison, other model selection criteria did determine the correct number of clusters for some subsets and failed to do so for other subsets. For instance, we notice that the AIC criteria performed an accurate model selection for all subsets but $S4$, where it determined 3 clusters instead of 4. BIC criteria gave near-similar results to AIC, with the exception of 6 clusters determined instead of 7 for $S1$. We notice more mixed results for the MMDL and MDL criteria, as the model selection was accurate in some subsets like $S4$, and inaccurate in others like $S2$.

Chapter 3

Hidden Markov Models with Multivariate Bounded Asymmetric Student's t-Mixture Model Emissions

This chapter proposes BASMMHMM: a novel HMM with multivariate bounded asymmetric Student's t-mixture model (BASMM) emissions. Our model 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, the merits of the BASMM (presented in the previous chapter) can add more flexibility to the HMMs when dealing with skewed observations, which are typical in many fields, such as financial or signal processing-related datasets. In this chapter, we present the necessary mathematical background for the BASMMHMM and we apply it in three different experiments: occupancy estimation, stock price prediction, and human activity recognition. The experimental results are discussed and compared with other Gaussian and t-based HMMs.

3.1 Hidden Markov Models

Hidden Markov Models (HMMs) are statistical models that are widely used for modeling temporal or sequential data, where the underlying state of a system is not directly observable but can

be inferred from observed data. HMMs have applications in various fields, including speech recognition, natural language processing, bioinformatics, finance, and many more. They are particularly useful for problems that involve pattern recognition in time-series data, where the underlying state is not directly observable, but can be inferred from the observed data.

An HMM consists of two main components: a hidden state sequence $(q_t)_{t=1}^{t=T}$ and an observable symbol sequence $(y_t)_{t=1}^{t=T}$. The hidden states are generally sampled from a specific range $\{S_1, S_2, \ldots, S_N\}$, and their sequence satisfies the Markov property. In fact, at a timestamp t, given the value of q_{t-1}, the current state q_t is independent of all the states of the sequence prior to the timestamp $t-1$. As for the observations, they are generated from the hidden states at each timestamp according to a probability distribution p over the symbols for each state. This is the emission distribution, and it is the object of our focus for this chapter.

Hidden Markov models (see Fig. 3.1) are fully defined with five elements:

- State space: a set of hidden states that the model can transition between.
- Observation space: the observations generated based on the current hidden state. These observations can be discrete or continuous and can have any number of dimensions.
- Transition probabilities: they determine the probability of transitioning from one hidden state to another. These probabilities are often represented in a transition matrix.
- Emission probabilities: they are the probabilities of generating a particular observation given the current hidden state. These probabilities are often represented in an emission matrix (in discrete HMMs)
- Initial state probabilities: HMMs use initial state probabilities to determine the probability of starting in a particular hidden state.

In this chapter, we focus on continuous HMMs, where the observations are sampled from continuous distributions. The BASMM [81] is the basis of our contribution in this chapter, as we intend to use it for modeling the observation emissions of our proposed HMM. The full mathematical definition of this mixture model are presented in 2.1 in the second chapter. The notations of the mathematical variables used for the rest of this chapter are detailed in Table 3.1.

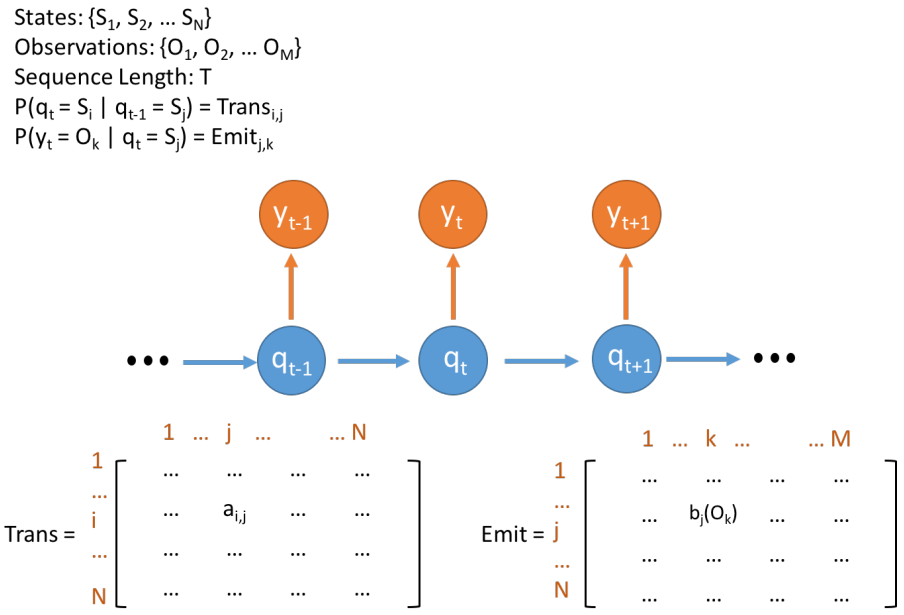

Figure 3.1: concept of a hidden Markov model

3.2 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 [54, 82]. 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:

$$P(y_t|\Theta_i) = \sum_{k=1}^{K} c_{ik} \times \mathcal{S}(y_t|\theta_{ik}) = \sum_{k=1}^{K} c_{ik} \times \mathcal{S}(y_t|\mu_{ik}, \Sigma_{ik}^{l}, \Sigma_{ik}^{r}, \nu_{ik}, \Omega_{ik}) \tag{55}$$

With the definition of the multivariate t given in equation (1), it is hard and computationally

Table 3.1: BASMMHMM notations

Notation	Definition
BASMMHMM	Bounded Asymmetric Student's-t Mixture Model Hidden Markov Model
$\mathcal{M} = \{\lambda_i^{t_0}, \lambda_{ij}, s_j, y_t\}_{i,j,t=1}^{N,N,L}$	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
$s_i = \{\alpha_{ik}, \mu_{ik}, \Sigma_{ik}, \nu_{ik}\}_{k=1}^{K}$	Parameters of the k^{th} component i^{th} hidden state's t-mixture for $k \in \{1, \ldots, K\}$
$\lambda = (\lambda_{ij})_{1 \leq i,j \leq N}$	$N \times N$ matrix, where λ_{ij} is the transition probability from state s_i to s_j
$\lambda^{t0} = (\lambda_i^{t0})_{1 \leq i \leq N}$	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

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 2.1.2. As a result, the probability of emitting the t^{th} observation y_t by the hidden state s_i (which corresponds to the emission mixture model Θ_i with the set of parameters $\Big(\theta_{ik} = \{\mu_{ik}, \Sigma_{lik}, \Sigma_{rik}\, \nu_{ik}, \Omega_{ik}\}\Big)_{k=1}^{K}$) is the following:

$$P(y_t|s_i) = \sum_{k=1}^{K} \frac{c_{ik} \times \mathcal{N}\Big(y_t|\mu_{ik}, \frac{\Sigma_{lik}}{\phi_{ik}}, \frac{\Sigma_{rik}}{\phi_{ik}}\Big) \times \mathcal{G}(\phi_{ik}) \times h(y_t, \Omega_{ik})}{\int_{\Omega_{ik}} \mathcal{T}(y|\mu_{ik}, \Sigma_{lik}, \Sigma_{rik}, \nu_{ik})\, dy} \tag{56}$$

where ϕ_{ik} is a precision parameter and $\phi_{ik} \sim \mathcal{G}(\frac{\nu}{2}, \frac{\nu}{2})$ (see section 2.1.2). We define also the observation indicators $(\delta_{it})_{i=t=1}^{i=I,t=L}$ by:

$$\delta_{it} = \begin{cases} 1 & \text{if the observation } y_t \text{ is emitted from the hidden state } s_i \\ 0 & \text{otherwise} \end{cases} \tag{57}$$

Also, given $\delta_{it} = 1$, we define the state-conditional mixture component indicators $(\eta_{ikt})_{k=1}^{K}$ as

follows:

$$\eta_{ikt} = \begin{cases} 1 & \text{if } y_t \text{ is emitted from the } k^{th} \text{ mixture component of the hidden state } s_i \\ 0 & \text{otherwise} \end{cases} \tag{58}$$

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:

$$\begin{aligned} P(y^c|s_i) = \prod_{k=1}^{K} \Bigg[c_{ik} \times \mathcal{N}\Big(y|\mu_{ik}, \frac{\Sigma_{lik}}{\phi_{ik}}, \frac{\Sigma_{rik}}{\phi_{ik}}\Big) \\ \times \frac{\mathcal{G}(\phi_{ik}) \times h(y, \Omega_{ik})}{\int_{\Omega_{ik}} \mathcal{T}(y|\mu_{ik}, \Sigma_{lik}, \Sigma_{rik}, \nu_{ik})\, dy} \Bigg]^{\eta_{ikt}} \end{aligned} \tag{59}$$

After calculations, the log-likelihood of the emission mixture for the i^{th} hidden state is given by:

$$\begin{aligned} &\log P(y^c|s_i) = \log \Big[\prod_{k=1}^{K} c_{ik} \times \mathcal{S}(y|\theta_{ik})^{\eta_{ikt}} \Big] \\ &= \sum_{k=1}^{K} \eta_{ikt} \times \Bigg[- \log \Gamma(\frac{\nu_{ik}}{2}) + \frac{\nu_{ik}}{2} \Big(\log (\frac{\nu_{ik}}{2}) - \phi_{ik} + \log \phi_{ik} \Big) \\ &- \frac{1}{2} \Big(\log |\Sigma_{ik}| + d \log(2\pi) + \phi_{ik} \Delta(y, \mu_{ik}; \Sigma_{ik}) \Big) \\ &- \log \int_{\Omega_{ik}} \mathcal{T}(y|\mu_{ik}, \Sigma_{lik}, \Sigma_{rik}, \nu_{ik})\, dy \Bigg] \end{aligned} \tag{60}$$

where Σ_{ik} can be the left or the right covariance matrix based on whether $y \leq 0$ or otherwise.

3.2.1 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:

$$E(\mathcal{M}) = \left(\prod_{i=1}^{N} \lambda_i^{\delta_{i1}} \right) \times \left(\prod_{i=1}^{N} \prod_{j=1}^{N} \prod_{t=1}^{L-1} \lambda_{ij}^{\delta_{it} \times \delta_{jt+1}} \right) \times \left(\prod_{j=1}^{N} \prod_{t=1}^{L} P(y_t^c | s_j)^{\delta_{jt}} \right) \tag{61}$$

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

$$\begin{aligned} &\mathcal{L}(\mathcal{M}) = \log\left(E(\mathcal{M})\right) \\ &= \sum_{i=1}^{N} \left(\delta_{i1} \log \lambda_i + \sum_{j=1}^{N} \sum_{t=1}^{L-1} \delta_{it} \delta_{jt+1} \log \lambda_{ij} \right) + \sum_{j=1}^{N} \sum_{t=1}^{L} \delta_{jt} \log P(y_t^c | s_j) \end{aligned} \tag{62}$$

3.2.2 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 $\{\lambda_i, \lambda_{i,j}, s_j\}_{i,j=1}^{N,N}$ that best fits the sequence of observations $Y = (y_t)_{t=1}^{L}$. This is done by maximizing the likelihood (see equation (62)) in an EM algorithm. let ρ_{it} and ρ_{ijt} be the posterior emission probabilities defined as follows:

$$\rho_{it} = P(\delta_{it} = 1 | y_t) \tag{63}$$

$$\rho_{ijt} = P(\delta_{jt+1} = 1, \delta_{it} = 1 | y_t) \tag{64}$$

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 [83].

Baum-Welch: Expectation

(1) Calculate the forward value α, where $\alpha_t(i)$ is the probability of being in the i^{th} 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 i^{th} state at the t^{th} 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 ρ_{ijt}: the probability of being in state i at time t then being in state j at time $t+1$. ρ_{ijt} is calculated using the forward and backward values as follows:

$$\begin{aligned}\rho_{ijt} &= \frac{\alpha_t(i) \times \lambda_{ij} P(y_{t+1}|s_j) \times \beta_{t+1}(j)}{P(Y|\Theta)} \\ &= \frac{\alpha_t(i) \times \lambda_{ij} P(y_{t+1}|s_j) \times \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\alpha_t(i) \times \lambda_{ij} P(y_{t+1}|s_j) \times \beta_{t+1}(j)\right]}\end{aligned} \tag{65}$$

(4) Calculate the posterior emission values ρ_{it}, i.e., the probability of being in the i^{th} state at the time t, given the observations Y and the model Θ. We get the emission posteriors by summing over the ρ_{ijt} values for all states:

$$\rho_{i,t} = \sum_{j=1}^{N} \rho_{ijt} \tag{66}$$

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

$$\begin{aligned}\mathcal{Q}(\mathcal{M}) &= E(\mathcal{L}(\mathcal{M})) \\ &= \sum_{i=1}^{N}\Big(\rho_{i1}\log\lambda_i + \sum_{j=1}^{N}\sum_{t=1}^{L-1}\rho_{ijt}\log\lambda_{ij}\Big) + \sum_{j=1}^{N}\sum_{t=1}^{L}\rho_{jt}E(\log P(y_t^c|s_j))\end{aligned} \tag{67}$$

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 $(\lambda_i^{t_0})_{i=0}^{N}$ by using the γ values:

$$\widehat{\lambda_i^{t_0}} = \rho_{it_0} \quad ; \quad i \in \{1, 2, \ldots, N\} \tag{68}$$

(2) Update the state transition probabilities:

$$\widehat{\lambda_{ij}} = \frac{\text{number of transitions from } s_i \text{ to } s_j}{\text{number of transitions from } s_i} = \frac{\sum_{t=1}^{L-1} \rho_{ijt}}{\sum_{t=1}^{L} \rho_{it}} \tag{69}$$

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

$$\widehat{\mu_{ik}} = \frac{\sum_{t=1}^{L} \xi_{ikt}(u_{ik}(y_t)y_t - A_{ik})}{\sum_{t=1}^{L} \xi_{ikt} u_{ik}(y_t)}; \tag{70}$$

where ξ_{ikt} is the i^{th} state's mixture component membership posterior, i.e., the probability that the observation y_t is emitted from the k^{th} component of the i^{th} hidden state:

$$\xi_{ikt} = \frac{\rho_{it} c_{ik} \mathcal{S}(y_t|s_{ik})}{\sum_{j=1}^{K} c_{ij} \mathcal{S}(y_t|s_{ij})} \tag{71}$$

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

$$A_{ik} = \frac{\sum_{m=1}^{M} (S_m - \mu_{ik}) u_{ik}(S_m) h(S_m, \Omega_{ik})}{\sum_{l=1}^{M} h(S_l, \Omega_{ik})} \tag{72}$$

And u_{ikt} is the precision function for an observation y_t of dimension d:

$$u_{ik}(y_t) = \frac{d + \nu_{ik}}{\nu_{ik} + \Delta(y_t, \mu_{ik}; \Sigma_{ik})} \tag{73}$$

The mixing weights $(c_{ik})_{i=k=1}^{i=N,k=K}$ are updated by dividing the probability of emission from the k^{th} mixture component of the i^{th} hidden state by the total probability of being in that i^{th}

state at any timestamp in the Markov chain:

$$\widehat{c_{ik}} = \frac{\sum_{t=1}^{L} \xi_{ikt}}{\sum_{t=1}^{L} \sum_{l=1}^{K} \xi_{ilt}} = \frac{\sum_{t=1}^{L} \xi_{ikt}}{\sum_{t=1}^{L} \rho_{it}} \tag{74}$$

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

$$\widehat{\Sigma_{ik}} = \frac{\sum_{t=1}^{L} \xi_{ikt} u_{ikt} \times (y_t - \mu_{ik})(y_t - \mu_{ik})^T}{\sum_{t=1}^{L} \xi_{ikt}} - B_{ik} \tag{75}$$

where B_{ik} is given by:

$$B_{ik} = \frac{\sum_{m=1}^{M} (\Sigma_{ik} - (S_m - \mu_{ik})(S_m - \mu_{ik})^T u_{ik}(S_m)) h(S_m, \Omega_{ik})}{\sum_{m=1}^{M} h(S_m, \Omega_{ik})} \tag{76}$$

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

$$\begin{aligned} & g(\nu_{ik}, d) + 1 + \frac{1}{\sum_{t=1}^{L} \xi_{ikt}} \sum_{t=1}^{L} \xi_{ikt} \Big(\log u_{ik}(y_t) - u_{ik}(y_t) \Big) \\ & - \frac{1}{\sum_{m=1}^{M} h(S_m, \Omega_{ik})} \sum_{m=1}^{M} \Big(g(\nu_{ik}, d) + 1 + \log u_{ik}(S_m) - u_{ik}(S_m) \Big) = 0 \end{aligned} \tag{77}$$

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

$$g(\nu, d) = -\psi(\frac{\nu}{2}) + \log(\frac{\nu}{2}) + \psi(\frac{\nu + d}{2}) - \log(\frac{\nu + d}{2}) \tag{78}$$

There is no closed-form solution to the equation (77), so we use the Newton Raphson method [84] 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.

3.3 Experiments and Results

In this section, we select a few popular sequential data-based applications where we attempt to employ the BASMMHMM, 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.

3.3.1 Occupancy Estimation

In the field of smart buildings, occupancy estimation [85, 86, 87] 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 [88, 89]. So in this experiment, we also attempt to estimate the number of occupants in one room using signals from non-intrusive sensors.

Data

The dataset [90] 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. 3.2 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.

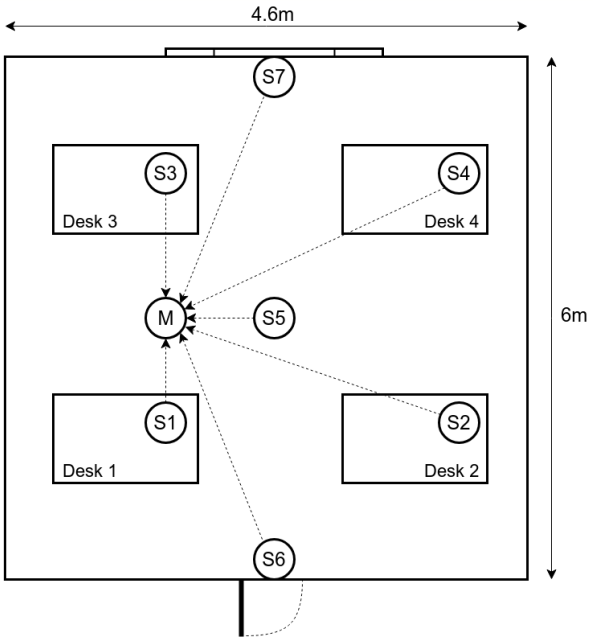

Figure 3.2: Sensors' layout in the room

The obtained data from these nodes spans 21 days (from 22 December 2017 to 11 January 2018) and has been recorded every 30 seconds, 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.

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 [91]. 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. 3.3.

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

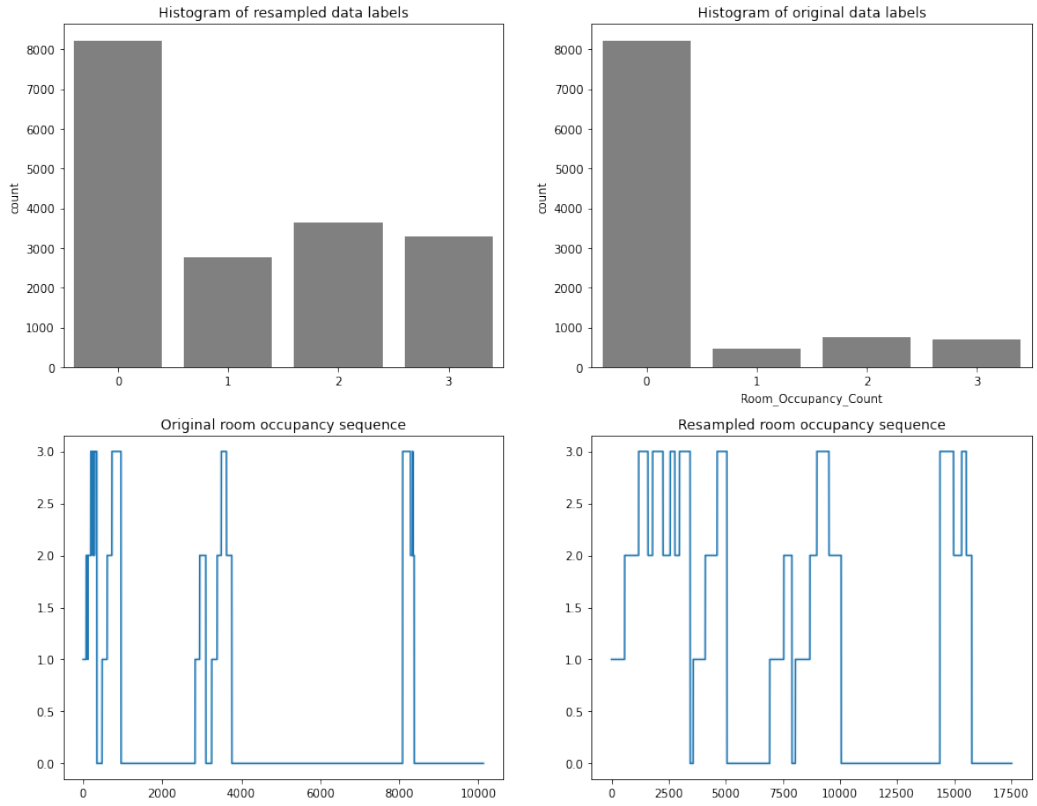

Figure 3.3: Original data versus resampled data

is chosen in a way that keeps the variance of the data above 0.95. Based on Fig. 3.4, we choose eight principal components.

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 3.2.

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

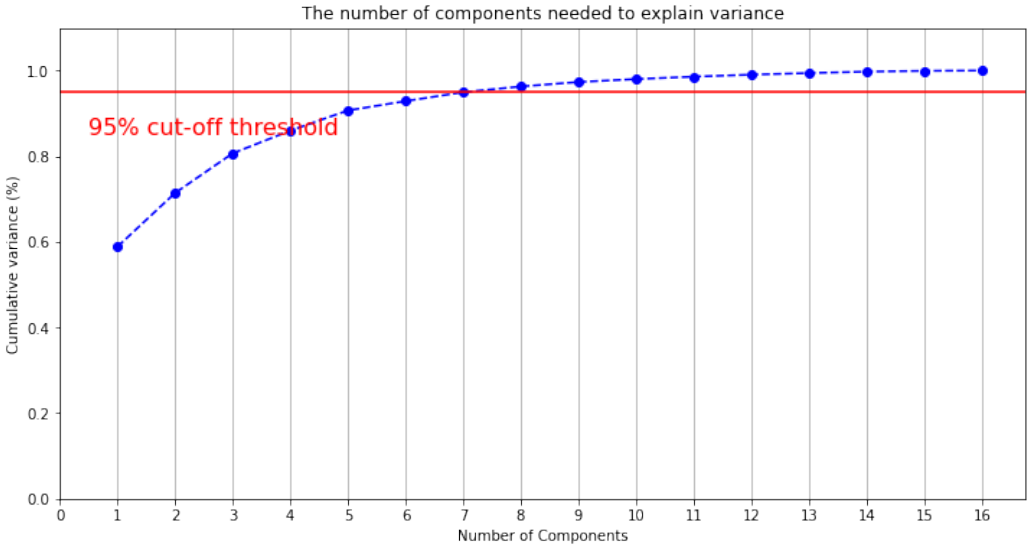

Figure 3.4: Data variance depending on the number of principal components

Table 3.2: 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

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. 3.5 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. 3.6 show a limited prediction of the non-majority classes.

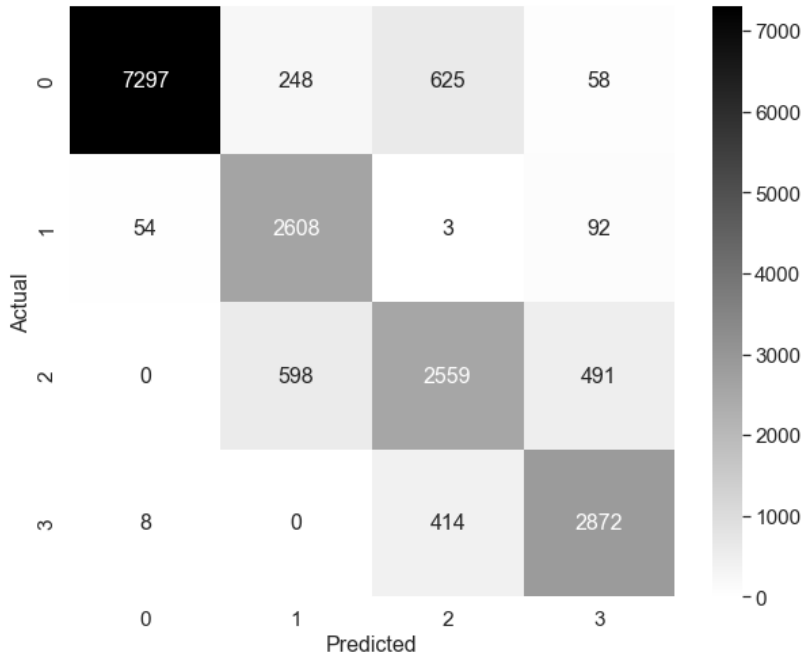

Figure 3.5: Occupancy estimation: confusion matrix of BASMMHMM

3.3.2 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.

Data and Preprocessing

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

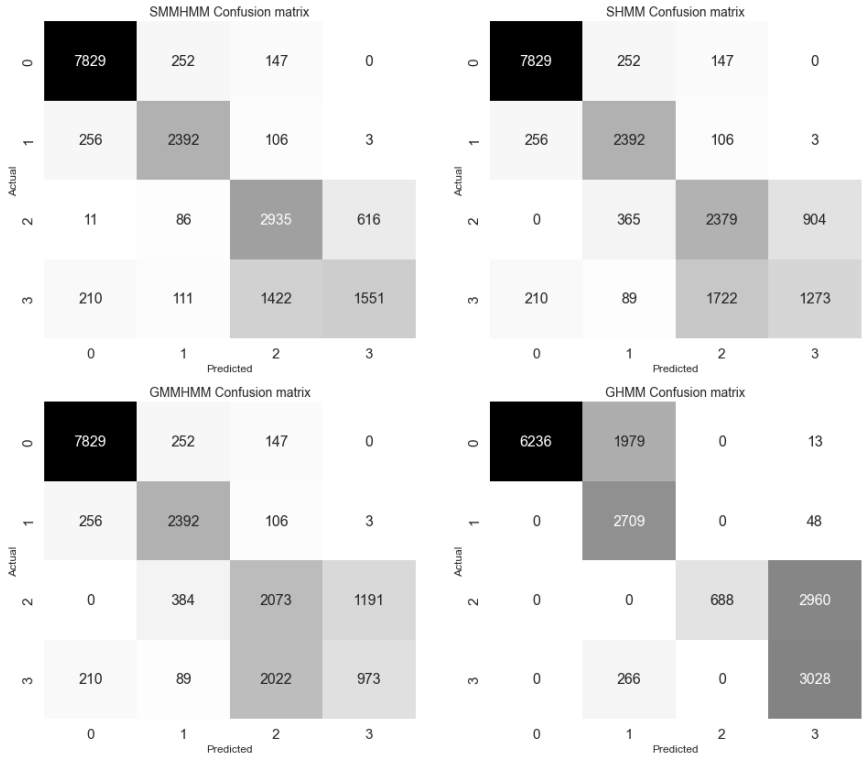

Figure 3.6: Occupancy estimation: confusion matrices of other HMMs

price, low price, and closing price. As for the preprocessing, we perform a MinMax 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.

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. The adopted approach is further explained in Fig. 3.7 and Fig. 3.8 below.

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

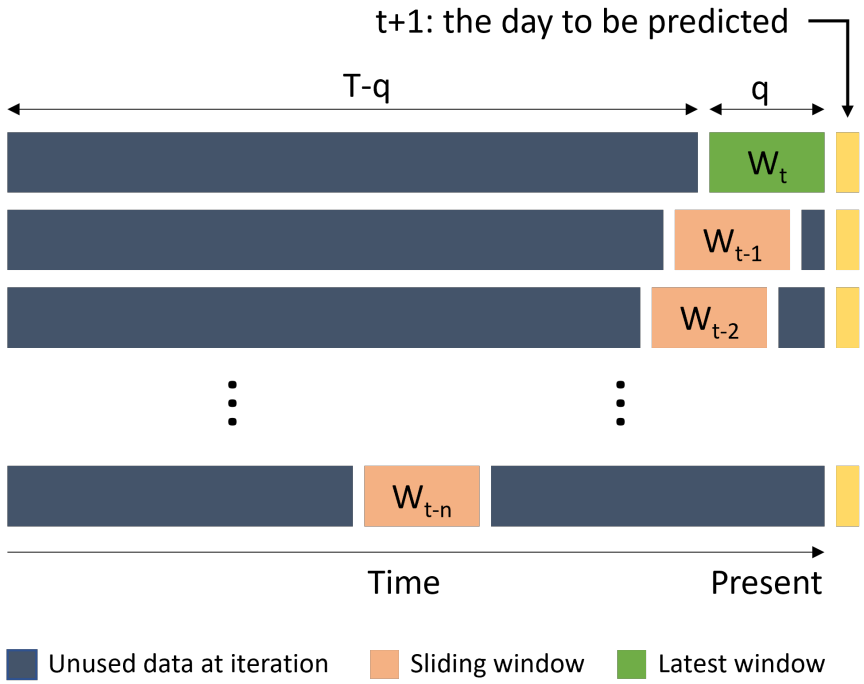

Figure 3.7: Forecasting the t+1 stock prices based on a sliding window of past k days

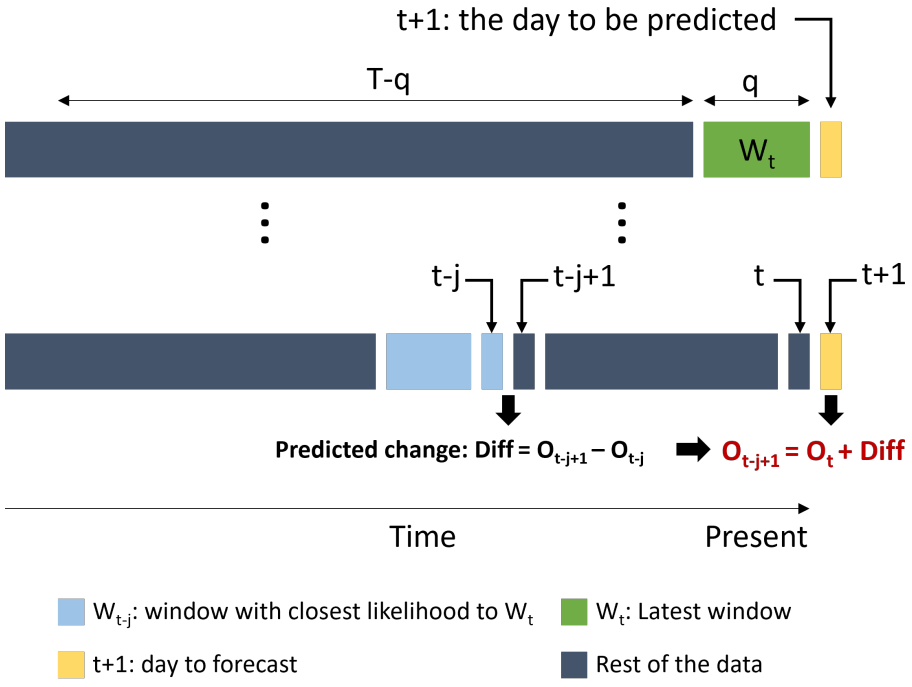

Figure 3.8: Predicted time-series calculation

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:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{y_i - x_i}{x_i} \times 100 \tag{79}$$

  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 equation (79), the RMSE formula is the following:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \tag{80}$$

Tables 3.3, 3.4, 3.5 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 [61]. 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

Table 3.3: AMZN stock price prediction: performance metrics for different models

	BASMMHMM		SMMHMM		GMMHMM	
Parameter	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 3.4: AAPL stock price prediction: performance metrics for different models

	BASMMHMM		SMMHMM		GMMHMM	
Parameter	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

t-mixture emissions (BASMMHMM, SMMHMM) performed better than the GMMHMM, which is based on Gaussian mixture emissions. We can see the graphs in Fig. 3.9, 3.10 and 3.11 a more clear picture of the predicted versus the actual stock prices.

3.3.3 Human Activity Recognition

Human Activity Recognition (HAR) is a popular scientific application that enables machines to recognize human body behaviours. HAR [92] 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 [93], which is popularly used in many research works.

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,

Table 3.5: GOOGL stock price prediction: performance metrics for different models

	BASMMHMM		SMMHMM		GMMHMM	
Parameter	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

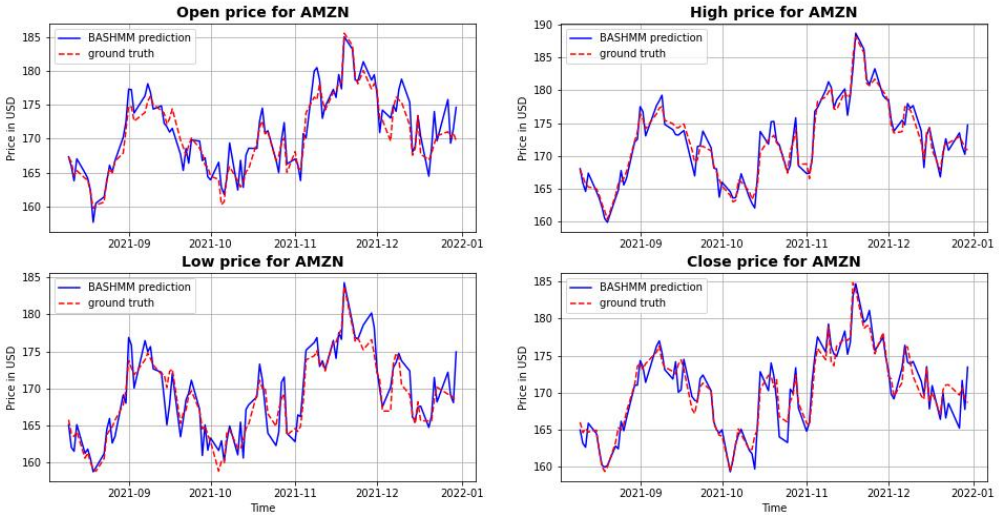

Figure 3.9: Amazon stock prices: BASMMHMM prediction versus ground truth

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

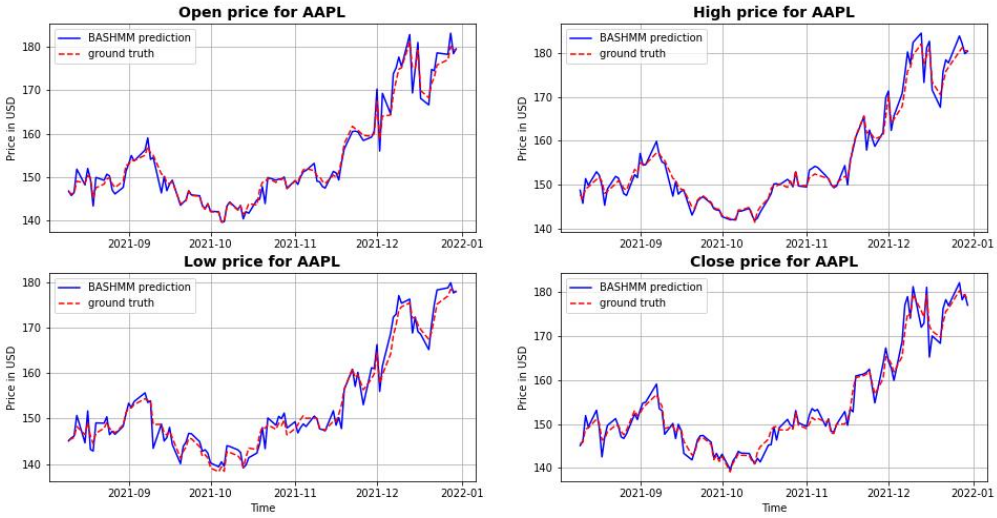

Figure 3.10: Apple stock prices: BASMMHMM prediction versus ground truth

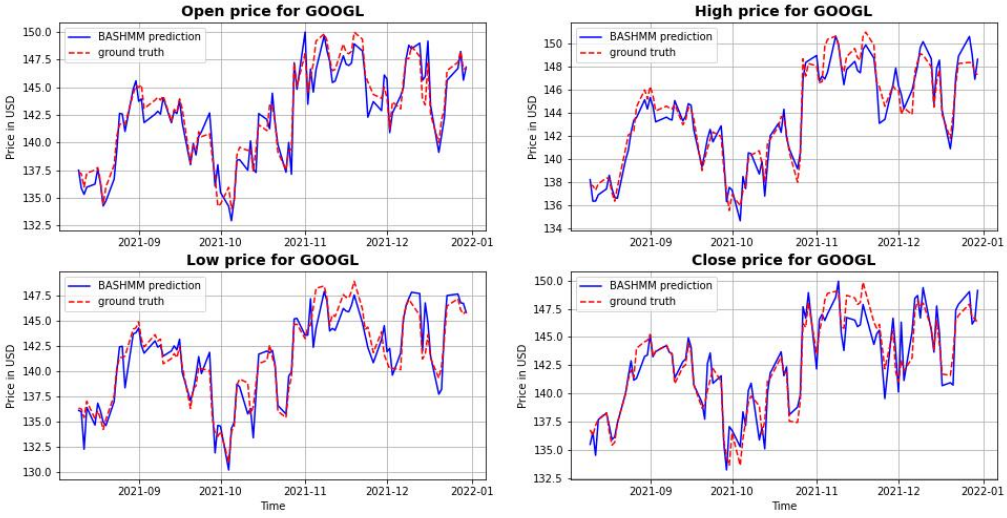

Figure 3.11: Google stock prices: BASMMHMM prediction versus ground truth

states and $K = 2$ mixture components per hidden state. The Fig. 3.12 summarizes the pipeline of the modeling in this experiment.

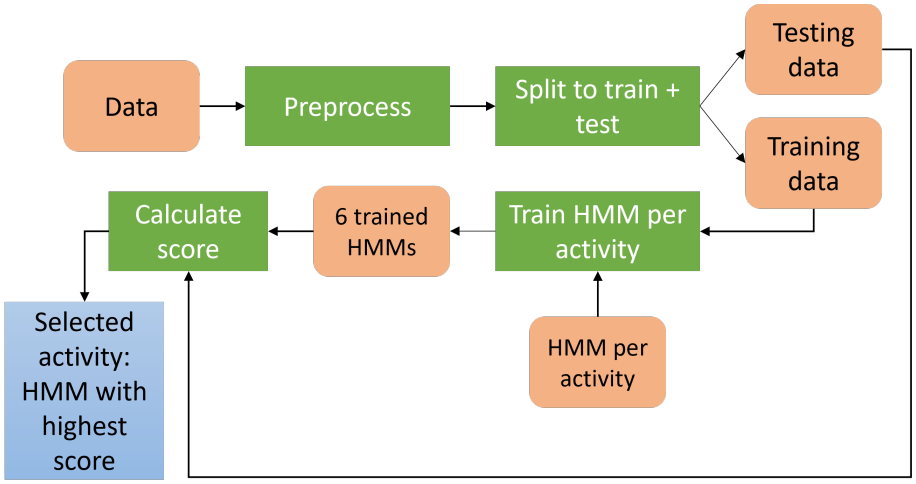

Figure 3.12: Human activity recognition: BASMMHMM framework

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.

The BASMMHMM did a better performance than the rest of the models, as shown in Table 3.6. 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.

Table 3.6: 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

Chapter 4

Conclusion

In this thesis, we proposed the use of minimum message length as a model selection criterion for the multivariate bounded asymmetric Student's t-mixture model (BASMM). This combination of BASMM clustering and MML model selection provides solutions for several limitations that were observed in other well-known mixture models and model selection criteria. On one hand, BASMM produces a better fit for the data with bounded support. Also, thanks to the heavy tails of the Student's t-distribution, BASMM is more robust to natural outliers in real-world datasets than the commonly-used GMM. Moreover, the asymmetry of BASMM offers a more realistic simulation to the data, as it is naturally asymmetric in most cases. On the other hand, the MML criterion is founded on the principle that the best model is the one that provides the most compact and efficient description of the data. This contrasts with other approaches that rely on heuristics or assumptions that may not always hold. In the light of these improvements offered by our model, we developed the detailed mathematical formulation for the EM algorithm and the MML model selection.

We explored the potential of BASMM+MML through three different experiments: electricity consumption profiles clustering, head pose estimation from images of drivers' faces, and acute leukemia detection from genetic expression data. All three experiments were validated by a wide range of performance metrics, and model selection was performed on multiple samples of the datasets. Throughout all experiments, the BASMM+MML demonstrated higher performance than other mixture models, namely GMM and SMM. This was accompanied by an overall more accurate model selection by MML compared to other criteria like MDL, AIC, BIC.

Furthermore, we used BASMMs as the observation emission densities of continuous HMMs to offer a more robust methodology for sequential data modeling. 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 mixture-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 its modeling "isolated" from the real information/experiment.

While this thesis provided robust variants for mixture modeling, model selection strategy, and HMMs, it is important to address any persisting limitations and identify potential areas of further research and improvement. Concerning the first contribution, the MML application on the fitted mixture model to high dimensional data has been computationally costly. In this regard, some optimizations (in the mathematical definition of MML or in the data fed to the algorithm) can be useful for the future. Also, the possibility of training the BASMM on a real-time stream of data has not been explored. Hence, it would be interesting to see how the clustering (BASMM) and the model selection (MML) can adapt to constantly incoming flows of data. In terms of experiments, we can further expand the existent applications in this contribution. For example, the drivers' head pose estimation can be performed from videos rather than images, and the electricity consumption profiles clustering can be extrapolated on different scales (hourly instead of daily consumption profiles and/or a city's consumption instead of one household).

As for the second contribution, there is room to improve the proposed model (BASMMHMM) 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 a model selection [23] approach before training the HMM can fulfill this tuning. This gives us the opportunity to merge the two contributions of this thesis and include MML as a model selection criterion for the BASMMHMM. Furthermore, in the case of high dimensional observations [94, 95], it is rigorous to implement a feature selection strategy [96, 97, 98] to avoid high computational complexity and to elect the parameters that represent the data in the most efficient way.

.1 Appendix: Fisher information calculation

Considering the relation between the multivariate Student's t-distribution and the Gaussian distribution discussed in 2.1.2, and based on the Fisher information for the Gaussian-based mixture models demonstrated in [64, 29], we define the determinant of the Fisher information matrices for the mean, the left and right covariance matrices, and the degrees of freedom in this appendix. For $k \in \{1, \ldots, K\}$, the Fisher information for the mean $\vec{\mu_k}$ of the k^{th} mixture component is defined as follows:

$$
\begin{aligned}
|F(\vec{\mu}_k)| = \prod_{d=1}^{D} \Bigg[& \sum_{\substack{i=1 \\ x_i < \vec{\mu}_k}}^{N} \Big[\Sigma_{lk}^{-1} \Big(\frac{\Sigma_{lk}^{-1} \big(\sum_{m=1}^{M} (l_{km} - \vec{\mu}_k) h(l_{km}|\theta_k) \chi(l_{km}|\Omega_k) \big)^2}{\big(\sum_{m=1}^{M} \chi(l_{km}|\Omega_k) \big)^2} \\
& - \frac{\sum_{m=1}^{M} h(y_{km}|\theta_k) \chi(r_{km}|\Omega_k) (r_{km} - \vec{\mu}_k) \Sigma_{lk}^{-1} (y_{km} - \vec{\mu}_k)^T}{\sum_{m=1}^{M} \chi(l_{km}|\Omega_k)} - 1 \Big) \Big] \\
& + \sum_{\substack{i=1 \\ x_i \geq \vec{\mu}_k}}^{N} \Big[\Sigma_{rk}^{-1} \Big(\frac{\Sigma_{rk}^{-1} \big(\sum_{m=1}^{M} (r_{km} - \vec{\mu}_k) h(r_{km}|\theta_k) \chi(r_{km}|\Omega_k) \big)^2}{\big(\sum_{m=1}^{M} \chi(r_{km}|\Omega_k) \big)^2} \\
& - \frac{\sum_{m=1}^{M} h(r_{km}|\theta_k) \chi(r_{km}|\Omega_k) (r_{km} - \vec{\mu}_k) \Sigma_{rk}^{-1} (r_{km} - \vec{\mu}_k)^T}{\sum_{m=1}^{M} \chi(r_{km}|\Omega_k)} - 1 \Big) \Big] \Bigg]
\end{aligned}
\tag{81}
$$

where $(l_{km})_{m=1}^{M}$ and $(r_{km})_{m=1}^{M}$ are two datasets sampled from the multivariate t-distributions with parameter sets $\{\vec{\mu}_k, \Sigma_{lk}, \nu_k\}$ $\{\vec{\mu}_k, \Sigma_{rk}, \nu_k\}$, respectively.

For $k \in \{1, \ldots, K\}$, the Fisher information for the left covariance matrix Σ_{lk} of the k^{th} mixture

component is defined as follows:

$$
\begin{aligned}
|F(\Sigma_{lk})| = &- \sum_{\substack{i=1 \\ x_i<\vec{\mu}_k}}^{N} 3\Sigma_{lk}^{-1}(x_i-\vec{\mu}_k)\Sigma_{lk}^{-1}(x_i-\vec{\mu}_k)^T h(x_i|\theta_k)^2 \\
&+ \sum_{\substack{i=1 \\ x_i<\vec{\mu}_k}}^{N} 2\Sigma_{lk}^{-1}\Big(\frac{\sum_{m=1}^{M}((l_{km}-\vec{\mu}_k)\Sigma_{rk}^{-1}(l_{km}-\vec{\mu}_k)^T h(l_{km}|\theta_k)^2\chi(l_{km}|\Omega_k)}{\sum_{m=1}^{M}\chi(l_{km}|\Omega_k)}\Big) \\
&- \sum_{\substack{i=1 \\ x_i<\vec{\mu}_k}}^{N} \Sigma_{lk}^{-1}\Big(\frac{\sum_{m=1}^{M}((l_{km}-\vec{\mu}_k)\Sigma_{rk}^{-1}(l_{km}-\vec{\mu}_k)^T)^2 h(l_{km}|\theta_k)^4\chi(l_{km}|\Omega_k)}{\sum_{m=1}^{M}\chi(l_{km}|\Omega_k)}\Big) \\
&+ \sum_{\substack{i=1 \\ x_i<\vec{\mu}_k}}^{N} 3\Big(\frac{\sum_{m=1}^{M}(l_{km}-\vec{\mu}_k)\Sigma_{lk}^{-1}(l_{km}-\vec{\mu}_k)^T h(l_{km}|\theta_k)^2\chi(l_{km}|\Omega_k)}{\sum_{m=1}^{M}\chi(l_{km}|\Omega_k)}\Big) \\
&- \sum_{\substack{i=1 \\ x_i<\vec{\mu}_k}}^{N} \Sigma_{lk}^{-1}\Big(\frac{\sum_{m=1}^{M}(l_{km}-\vec{\mu}_k)\Sigma_{lk}^{-1}(l_{km}-\vec{\mu}_k)^T h(l_{km}|\theta_k)^2\chi(l_{km}|\Omega_k)}{\sum_{m=1}^{M}\chi(l_{km}|\Omega_k)}\Big)^2
\end{aligned}
\tag{82}
$$

For $k \in \{1,\ldots,K\}$, the Fisher information for the right covariance matrix Σ_{rk} of the k^{th} mixture component is defined as follows:

$$
\begin{aligned}
|F(\Sigma_{rk})| = &- \sum_{\substack{i=1 \\ x_i<\vec{\mu}_k}}^{N} 3\Sigma_{rk}^{-1}(x_i-\vec{\mu}_k)\Sigma_{rk}^{-1}(x_i-\vec{\mu}_k)^T h(x_i|\theta_k)^2 \\
&+ \sum_{\substack{i=1 \\ x_i<\vec{\mu}_k}}^{N} 2\Sigma_{rk}^{-1}\Big(\frac{\sum_{m=1}^{M}((r_{km}-\vec{\mu}_k)\Sigma_{rk}^{-1}(r_{km}-\vec{\mu}_k)^T h(r_{km}|\theta_k)^2\chi(r_{km}|\Omega_k)}{\sum_{m=1}^{M}\chi(r_{km}|\Omega_k)}\Big) \\
&- \sum_{\substack{i=1 \\ x_i<\vec{\mu}_k}}^{N} \Sigma_{rk}^{-1}\Big(\frac{\sum_{m=1}^{M}((r_{km}-\vec{\mu}_k)\Sigma_{rk}^{-1}(r_{km}-\vec{\mu}_k)^T)^2 h(r_{km}|\theta_k)^4\chi(r_{km}|\Omega_k)}{\sum_{m=1}^{M}\chi(r_{km}|\Omega_k)}\Big) \\
&+ \sum_{\substack{i=1 \\ x_i<\vec{\mu}_k}}^{N} 3\Big(\frac{\sum_{m=1}^{M}(r_{km}-\vec{\mu}_k)\Sigma_{rk}^{-1}(r_{km}-\vec{\mu}_k)^T h(r_{km}|\theta_k)^2\chi(r_{km}|\Omega_k)}{\sum_{m=1}^{M}\chi(r_{km}|\Omega_k)}\Big) \\
&- \sum_{\substack{i=1 \\ x_i<\vec{\mu}_k}}^{N} \Sigma_{rk}^{-1}\Big(\frac{\sum_{m=1}^{M}(r_{km}-\vec{\mu}_k)\Sigma_{rk}^{-1}(r_{km}-\vec{\mu}_k)^T h(r_{km}|\theta_k)^2\chi(r_{km}|\Omega_k)}{\sum_{m=1}^{M}\chi(r_{km}|\Omega_k)}\Big)^2
\end{aligned}
\tag{83}
$$

Finally, for $k \in \{1,\ldots,K\}$, the Fisher information for the degrees of freedom ν_k of the k^{th}

mixture component is defined as follows:

$$|F(\nu_k)| = \sum_{i=1}^{N} \pi_k \frac{\partial}{\partial \nu_k} \Big(\frac{\frac{\partial}{\partial \nu_k} P(\vec{X}_i|\theta_k)}{\sum_{j=1}^{K} \pi_j P(\vec{X}_i|\theta_j)} \Big) \tag{84}$$

where for $i \in \{1, \ldots, N\}$:

$$\frac{\partial}{\partial \nu_k} P(\vec{X}_i|\theta_k) = \frac{\chi(\vec{X}_i|\Omega_k)}{\int_{\Omega} \mathcal{S}(\vec{Y}|\vec{\mu}, \Sigma_{lk}, \Sigma_{rk}, \nu_k)\, d\vec{Y}} \times \frac{\partial}{\partial \nu_k} \mathcal{S}(\vec{X}_i|\vec{\mu_k}, \Sigma_{lk}, \Sigma_{rk}, \nu_k) \tag{85}$$

Bibliography

[1] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[2] Nizar Bouguila and Wentao Fan. *Mixture models and applications*. Springer, New York, 2020.

[3] Nizar Bouguila and Djemel Ziou. High-dimensional unsupervised selection and estimation of a finite generalized dirichlet mixture model based on minimum message length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1716–1717, 2007.

[4] D. Peel and G. J. Mclachlan. Robust mixture modelling using the t distribution. *Statistics and Computing*, (10):339–348, 2000.

[5] Nizar Bouguila and Djemel Ziou. On fitting finite dirichlet mixture using ECM and MML. In Peng Wang, Maneesha Singh, Chidanand Apté, and Petra Perner, editors, *Pattern Recognition and Data Mining, Third International Conference on Advances in Pattern Recognition, ICAPR 2005, Bath, UK, August 22-25, 2005, Proceedings, Part I*, volume 3686 of *Lecture Notes in Computer Science*, pages 172–182. Springer, 2005.

[6] Bromensele Samuel Oboh and Nizar Bouguila. Unsupervised learning of finite mixtures using scaled dirichlet distribution and its application to software modules categorization. In *2017 IEEE International Conference on Industrial Technology (ICIT)*, pages 1085–1090, 2017.

[7] Douglas A Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

[8] S Mohammad Khansari-Zadeh and Aude Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011.

[9] Bin Hu, Ryan Wen Liu, Kai Wang, Yan Li, Maohan Liang, Huanhuan Li, and Jingxian Liu. Statistical analysis of massive ais trajectories using gaussian mixture models. In *2017 2nd International Conference on Multimedia and Image Processing (ICMIP)*, pages 113–117. IEEE, 2017.

[10] Fatma Najar, Sami Bourouis, Nizar Bouguila, and Safya Belghith. Unsupervised learning of finite full covariance multivariate generalized gaussian mixture models for human activity recognition. *Multim. Tools Appl.*, 78(13):18669–18691, 2019.

[11] Ravi Teja Vemuri, Muhammad Azam, Nizar Bouguila, and Zachary Patterson. A bayesian sampling framework for asymmetric generalized gaussian mixture models learning. *Neural Computing and Applications*, pages 1–12, 2021.

[12] Tarek Elguebaly and Nizar 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.

[13] Mohand Saïd Allili, Djemel Ziou, Nizar Bouguila, and Sabri Boutemedjet. Image and video segmentation by combining unsupervised generalized gaussian mixture modeling and feature selection. *IEEE Trans. Circuits Syst. Video Technol.*, 20(10):1373–1377, 2010.

[14] Tarek Elguebaly and Nizar Bouguila. Bayesian learning of finite generalized gaussian mixture models on images. *Signal Processing*, 91(4):801–820, 2011.

[15] Fatma Najar, Sami Bourouis, Nizar Bouguila, and Safiya Belghith. A comparison between different gaussian-based mixture models. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pages 704–708, 2017.

[16] Muhammad Azam, Basim Alghabashi, and Nizar Bouguila. Multivariate bounded asymmetric gaussian mixture model. In *Mixture Models and Applications*, pages 61–80. Springer, New York, 2020.

[17] Tarek Elguebaly and Nizar Bouguila. Finite asymmetric generalized gaussian mixture models learning for infrared object detection. *Computer Vision and Image Understanding*, 117(12):1659–1671, 2013.

[18] Tarek Elguebaly and Nizar Bouguila. Background subtraction using finite mixtures of asymmetric gaussian distributions and shadow detection. *Mach. Vis. Appl.*, 25(5):1145–1162, 2014.

[19] Tanh Minh Nguyen and Q.M. Jonathan Wu. Multivariate student's-t mixture model for bounded support data. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5548–5552, Windsor, ON, Canada, 2013.

[20] T. M. Nguyen and Q. M. J. Wu. Bounded asymmetrical student's-t mixture model. *IEEE Transactions on Cybernetics*, 44(6):857–869, 2014.

[21] Babak Barazandeh and Meisam Razaviyayn. On the behavior of the expectation-maximization algorithm for mixture models. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 61–65. IEEE, 2018.

[22] Kevin J Grimm, Gina L Mazza, and Pega Davoudzadeh. Model selection in finite mixture models: A k-fold cross-validation approach. *Structural Equation Modeling: A Multidisciplinary Journal*, 24(2):246–256, 2017.

[23] Gilles Celeux, Sylvia Frühwirth-Schnatter, and Christian P Robert. Model selection for mixture models–perspectives and strategies. In *Handbook of mixture analysis*, pages 117–154. Chapman and Hall/CRC, New York, 2019.

[24] Arijit Chakrabarti and Jayanta K Ghosh. Aic, bic and recent advances in model selection. *Philosophy of statistics*, pages 583–605, 2011.

[25] Geoffrey J McLachlan, Sharon X Lee, and Suren I Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6:355–378, 2019.

[26] Jorma Rissanen. *Stochastic complexity in statistical inquiry*, volume 15. World scientific, 1998.

[27] Málrio AT Figueiredo, Jose MN Leitao, and Anil K Jain. On fitting mixture models. In *Energy Minimization Methods in Computer Vision and Pattern Recognition: Second International Workshop, EMMCVPR'99 York, UK, July 26–29, 1999 Proceedings 2*, pages 54–69. Springer, 1999.

[28] Mario A.T. Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

[29] Muhammad Azam and Nizar Bouguila. Multivariate-bounded gaussian mixture model with minimum message length criterion for model selection. *Expert Systems*, 38(5):e12688, 2021.

[30] Muhammad Azam and Nizar Bouguila. Multivariate bounded support asymmetric generalized gaussian mixture model with model selection using minimum message length. *Expert Systems with Applications*, page 117516, 2022.

[31] Can Hu, Wentao Fan, Ji-Xiang Du, and Nizar Bouguila. A novel statistical approach for clustering positive data based on finite inverted beta-liouville mixture models. *Neurocomputing*, 333:110–123, 2019.

[32] Rohan A Baxter and Jonathan J Oliver. Finding overlapping components with mml. *Statistics and Computing*, 10(1):5–16, 2000.

[33] Mohd Izhan Mohd Yusoff, Ibrahim Mohamed, and Mohd Rizam Abu Bakar. Hidden markov models: an insight. In *Proceedings of the 6th International Conference on Information Technology and Multimedia*, pages 259–264. IEEE, 2014.

[34] Wenjuan Hou, Wentao Fan, Manar Amayri, and Nizar Bouguila. A novel continuous hidden markov model for modeling positive sequential data. In *Hidden Markov Models and Applications*, pages 199–210. Springer, New York, 2022.

[35] James R Norris. *Markov chains*. Number 2. Cambridge university press, Cambridge, 1998.

[36] Kai Lai Chung. *Markov Chains: With Stationary Transition Probabilities*, volume 104. Springer Science & Business Media, Heidelberg, 2012.

[37] Phil Blunsom. Hidden markov models. *Lecture notes, August*, 15(18-19):48, 2004.

[38] Mark Stamp. A revealing introduction to hidden markov models. *Department of Computer Science San Jose State University*, pages 26–56, 2004.

[39] Sean R Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996.

[40] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, Cambridge, 1998.

[41] Walter Zucchini and Peter Guttorp. A hidden markov model for space-time precipitation. *Water Resources Research*, 27(8):1917–1923, 1991.

[42] Yariv Ephraim and Neri Merhav. Hidden markov processes. *IEEE Transactions on information theory*, 48(6):1518–1569, 2002.

[43] Nguyet Nguyen. An analysis and implementation of the hidden markov model to technology stock prediction. *Risks*, 5(4):62, 2017.

[44] Hermann Ney and Stefan Ortmanns. Dynamic programming search for continuous speech recognition. *IEEE Signal Processing Magazine*, 16(5):64–83, 1999.

[45] Biing Hwang Juang and Laurence R Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991.

[46] David RH Miller, Tim Leek, and Richard M Schwartz. A hidden markov model information retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 214–221, 1999.

[47] Stevenn Volant, Caroline Bérard, Marie-Laure Martin-Magniette, and Stéphane Robin. Hidden markov models with mixtures as emission. *Statistics and Computing*, 24(4):493–504, 2014.

[48] Elise Epaillard and Nizar Bouguila. Hidden markov models based on generalized dirichlet mixtures for proportional data modeling. In Neamat El Gayar, Friedhelm Schwenker, and

Cheng Suen, editors, *Artificial Neural Networks in Pattern Recognition - 6th IAPR TC 3 International Workshop, ANNPR 2014, Montreal, QC, Canada, October 6-8, 2014. Proceedings*, volume 8774 of *Lecture Notes in Computer Science*, pages 71–82. Springer, 2014.

[49] Salman A Shaikh and Hiroyuki Kitagawa. Efficient distance-based outlier detection on uncertain datasets of gaussian distribution. *World Wide Web*, 17(4):511–538, 2014.

[50] Zixiang Xian, Muhammad Azam, Manar Amayri, Wentao Fan, and Nizar Bouguila. Bounded asymmetric gaussian mixture-based hidden markov models. In *Hidden Markov Models and Applications*, pages 33–58. Springer, New York, 2022.

[51] Rui Li and Saralees Nadarajah. A review of student'st distribution and its generalizations. *Empirical Economics*, 58(3):1461–1490, 2020.

[52] David Peel and Geoffrey J McLachlan. Robust mixture modelling using the t distribution. *Statistics and computing*, 10(4):339–348, 2000.

[53] Sotirios P Chatzis, Dimitrios I Kosmopoulos, and Theodora A Varvarigou. Robust sequential data modeling using an outlier tolerant hidden markov model. *IEEE transactions on pattern analysis and machine intelligence*, 31(9):1657–1669, 2008.

[54] Hui Zhang, Qing Ming Jonathan Wu, and Thanh Minh Nguyen. Modified student's t-hidden markov model for pattern recognition and classification. *IET Signal Processing*, 7(3):219–227, 2013.

[55] Yuhui Zheng, Byeungwoo Jeon, Le Sun, Jianwei Zhang, and Hui Zhang. Student's t-hidden markov model for unsupervised learning using localized feature selection. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2586–2598, 2017.

[56] Samr Ali and Nizar Bouguila. A roadmap to hidden markov models and a review of its application in occupancy estimation. *Hidden Markov Models and Applications*, pages 1–31, 2022.

[57] Parviz Asghari, Elnaz Soleimani, and Ehsan Nazerfard. Online human activity recognition employing hierarchical hidden markov models. *Journal of Ambient Intelligence and Humanized Computing*, 11(3):1141–1152, 2020.

[58] Chuanhai Liu and Donald B Rubin. Ml estimation of the t distribution using em and its extensions, ecm and ecme. *Statistica Sinica*, pages 19–39, 1995.

[59] BM Golam Kibria and Anwar H Joarder. A short review of multivariate t-distribution. *Journal of Statistical research*, 40(1):59–72, 2006.

[60] Sivaraman Balakrishnan, Martin J Wainwright, and Bin Yu. Statistical guarantees for the em algorithm: From population to sample-based analysis. 2017.

[61] Jonathan J Oliver, Rohan A Baxter, and Chris S Wallace. Unsupervised learning using mml. In *ICML*, pages 364–372. Citeseer, 1996.

[62] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[63] M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain. Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1154–1166, 2004.

[64] Zixiang Xian, Muhammad Azam, Manar Amayri, and Nizar Bouguila. Model selection criterion for multivariate bounded asymmetric gaussian mixture model. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 1436–1440. IEEE, 2021.

[65] Mohand Said Allili, Nizar Bouguila, and Djemel Ziou. Finite general gaussian mixture modeling and application to image and video foreground segmentation. *Journal of Electronic Imaging*, 17(1):013005, 2008.

[66] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[67] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):1–13, 2020.

[68] Kehua Li, Zhenjun Ma, Duane Robinson, and Jun Ma. Identification of typical building daily electricity usage profiles using gaussian mixture model-based clustering and hierarchical clustering. *Applied Energy*, 231:331–342, 2018.

[69] Lulu Wen, Kaile Zhou, and Shanlin Yang. A shape-based clustering method for pattern recognition of residential electricity consumption. *Journal of cleaner production*, 212:475–488, 2019.

[70] Yi Wang, Qixin Chen, Chongqing Kang, and Qing Xia. Clustering of electricity consumption behavior dynamics toward big data applications. *IEEE transactions on smart grid*, 7(5):2437–2447, 2016.

[71] Georges Hebrail and Alice Berard. UCI machine learning repository, 2012.

[72] Félix Iglesias, Tanja Zseby, and Arthur Zimek. Absolute cluster validity. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2096–2112, 2019.

[73] Antonio M. López Katerine Diaz-Chito, Aura Hernández-Sabaté. A reduced feature set for driver head pose estimation. *Applied Soft Computing*, 45:98–107, 2016.

[74] Donghyun Kim and Hyeyoung Park. An efficient face recognition through combining local features and statistical feature extraction. In *PRICAI 2010: Trends in Artificial Intelligence: 11th Pacific Rim International Conference on Artificial Intelligence, Daegu, Korea, August 30–September 2, 2010. Proceedings 11*, pages 456–466. Springer, 2010.

[75] Yucheng Wei, Ludovic Fradet, and Tieniu Tan. Head pose estimation using gabor eigenspace modeling. In *Proceedings. International Conference on Image Processing*, volume 1, pages I–I. IEEE, 2002.

[76] Joni-Kristian Kamarainen. Gabor features in image analysis. In *2012 3rd international conference on image processing theory, tools and applications (IPTA)*, pages 13–14. IEEE, 2012.

[77] Jian-Gang Wang, Jun Li, Chong Yee Lee, and Wei-Yun Yau. Dense sift and gabor descriptors-based face representation with applications to gender recognition. In *2010 11th International Conference on Control Automation Robotics & Vision*, pages 1860–1864. IEEE, 2010.

[78] Raheel Baig, Abdur Rehman, Abdullah Almuhaimeed, Abdulkareem Alzahrani, and Hafiz Tayyab Rauf. Detecting malignant leukemia cells using microscopic blood smear images: A deep learning approach. *Applied Sciences*, 12(13):6317, 2022.

[79] Sara A Monaghan, Jeng-Lin Li, Yen-Chun Liu, Ming-Ya Ko, Michael Boyiadzis, Ting-Yu Chang, Yu-Fen Wang, Chi-Chun Lee, Steven H Swerdlow, and Bor-Sheng Ko. A machine learning approach to the classification of acute leukemias and distinction from nonneoplastic cytopenias using flow cytometry data. *American journal of clinical pathology*, 157(4):546–553, 2022.

[80] Stefanie Warnat-Herresthal, Konstantinos Perrakis, Bernd Taschler, Matthias Becker, Kevin Baßler, Marc Beyer, Patrick Günther, Jonas Schulte-Schrepping, Lea Seep, Kathrin Klee, et al. Scalable prediction of acute myeloid leukemia using high-dimensional machine learning and blood transcriptomics. *Iscience*, 23(1):100780, 2020.

[81] Thanh Minh Nguyen and QM Jonathan Wu. Bounded asymmetrical student's-t mixture model. *IEEE transactions on cybernetics*, 44(6):857–869, 2013.

[82] Zhongsheng Chen and Yongmin Yang. Fault diagnostics of helicopter gearboxes based on multi-sensor mixtured hidden markov models. *Journal of vibration and acoustics*, 134(3), 2012.

[83] Michael Collins. The forward-backward algorithm. *Columbia Columbia Univ*, 2013.

[84] Tjalling J Ypma. Historical development of the newton–raphson method. *SIAM review*, 37(4):531–551, 1995.

[85] Manar Amayri, Stephane Ploix, Nizar Bouguila, and Frederic Wurtz. Estimating occupancy using interactive learning with a sensor environment: Real-time experiments. *IEEE Access*, 7:53932–53944, 2019.

[86] Nuha Zamzami, Manar Amayri, Nizar Bouguila, and Stephane Ploix. Online clustering for estimating occupancy in an office setting. In *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, pages 2195–2200, 2019.

[87] Muhammad Azam, Marion Blayo, Jean-Simon Venne, and Michel Allegue-Martinez. Occupancy estimation using wifi motion detection via supervised machine learning algorithms. In *2019 ieee global conference on signal and information processing (GlobalSIP)*, pages 1–5. IEEE, 2019.

[88] Manar Amayri, Quoc-Dung Ngo, Stephane 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)*, volume 2, pages 690–695. IEEE, 2017.

[89] Rim Nasfi, Manar Amayri, and Nizar Bouguila. A novel approach for modeling positive vectors with inverted dirichlet-based hidden markov models. *Knowledge-Based Systems*, 192:105335, 2020.

[90] Adarsh Pal Singh, Vivek Jain, Sachin Chaudhari, Frank Alexander Kraemer, Stefan Werner, and Vishal Garg. Machine learning-based occupancy estimation using multivariate sensor nodes. in 2018 ieee globecom workshops (gc wkshps), 2018.

[91] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[92] Zixiang Xian, Muhammad Azam, and Nizar Bouguila. Statistical modeling using bounded asymmetric gaussian mixtures: Application to human action and gender recognition. In *2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI)*, pages 41–48. IEEE, 2021.

[93] Jorge-Luis Reyes-Ortiz, Davide Anguita, Alessandro Ghio, and X Parra. Human activity recognition using smartphones data set. *UCI Machine Learning Repository; University of California, Irvine, School of Information and Computer Sciences: Irvine, CA, USA*, 2012.

[94] Nizar Bouguila, Khaled Almakadmeh, and Sabri Boutemedjet. A finite mixture model for simultaneous high-dimensional clustering, localized feature selection and outlier rejection. *Expert Syst. Appl.*, 39(7):6641–6656, 2012.

[95] Tarek Elguebaly and Nizar Bouguila. Simultaneous high-dimensional clustering and feature selection using asymmetric gaussian mixture models. *Image Vis. Comput.*, 34:27–41, 2015.

[96] Samr Ali and Nizar Bouguila. Hidden markov models: Discrete feature selection in activity recognition. In *Hidden Markov Models and Applications*, pages 103–155. Springer, New York, 2022.

[97] Srikanth Amudala, Samr Ali, and Nizar Bouguila. Variational inference of infinite generalized gaussian mixture models with feature selection. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 120–127, 2020.

[98] Wentao Fan, Ru Wang, and Nizar Bouguila. Simultaneous positive sequential vectors modeling and unsupervised feature selection via continuous hidden markov models. *Pattern Recognit.*, 119:108073, 2021.