# BAYESIAN LEARNING FRAMEWORKS FOR MULTIVARIATE BETA MIXTURE MODELS

Mahsa Amirkhani

A thesis
in
The Department
of
Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science
(Quality Systems Engineering)
Concordia University
Montréal, Québec, Canada

March 2021

# Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Mahsa Amirkhani**

Entitled: **Bayesian Learning Frameworks for Multivariate Beta Mixture Models**

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 commitee:

Dr. Ayda Basyouni _____ Chair

Dr. Mohamed Ouf _____ External Examiner, BCEE

Dr. Mohsen Ghafouri _____ CIISE Examiner

Dr. Nizar Bouguila _____ Supervisor

Approved _____
Dr. Mohammad Mannan, Graduate Program Director

_____ 2021 _____ _____
Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

# Abstract

Bayesian Learning Frameworks for Multivariate Beta Mixture Models

Mahsa Amirkhani

Mixture models have been widely used as a statistical learning paradigm in various unsupervised machine learning applications, where labeling a vast amount of data is impractical and costly. They have shown a significant success and encouraging performance in many real-world problems from different fields such as computer vision, information retrieval and pattern recognition. One of the most widely used distributions in mixture models is Gaussian distribution, due to its characteristics, such as its simplicity and fitting capabilities. However, data obtained from some applications could have different properties like non-Gaussian and asymmetric nature.

In this thesis, we propose multivariate Beta mixture models which offer flexibility, various shapes with promising attributes. These models can be considered as decent alternatives to Gaussian distributions.

We explore multiple Bayesian inference approaches for multivariate Beta mixture models and propose a suitable solution for the problem of estimating parameters using Markov Chain Monte Carlo (MCMC) technique. We exploit Gibbs sampling within Metropolis-Hastings for learning parameters of our finite mixture model. Moreover, a fully Bayesian approach based on birth-death MCMC technique is proposed which simultaneously allows cluster assignments, parameters estimation and the selection of the optimal number of clusters. Finally, we develop a nonparametric Bayesian framework by extending our finite mixture model to infinity using Dirichlet process to tackle the model selection problem. Experimental results obtained from challenging applications (e.g., intrusion detection, medical, etc.) confirm that our proposed frameworks can provide effective solutions comparing to existing alternatives.

# Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Nizar Bouguila for his invaluable advice, continuous support, trust and patience during my study. I will be always thankful for giving me the opportunity to be your student and learning from you.

While working on my thesis, I had the chance to have Narges beside me and I would like to thank her for helpful guidance, support and encouragement. In addition, I like to thank all of my great friends and labmates, specially Ravi, Sadegh, Behnam, Pantea, Yogesh and Kamal that provide me with a supportive and nice environment with a lot of fruitful discussions.

Last but not least, I would like to thank my family for their unconditional support, respect and trust throughout my studies and entire life. Your endless love and care always encourage me.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Data mining and machine learning tools have received much attention recently because of their capability in modeling and analyzing collected data in various fields and applications such as computer vision, information retrieval and pattern recognition [1]. One of the important approaches that has been widely adopted for knowledge discovery and finding the underlying structure of the data is clustering. Clustering is an unsupervised learning approach which involves portioning data into different groups with similar characteristics. The idea is to assign unlabeled data into clusters such that data within a cluster are similar to each other and far from data in other clusters [2]. Among statistical learning techniques, finite mixture models have demonstrated high capability to model complex data sets by considering that each observation has arisen from one of the different groups or components [3, 4]. Furthermore, choosing the most proper probability distribution plays an important role in adapting this model in order to well describe the components. In particular, Gaussian mixture models (GMM) have been widely used for categorization problems and demonstrated satisfactory fitting abilities on different applications and situations [5]. However, the Gaussian assumption under more general circumstances is not realistic and data obtained with different properties from various real-life applications may have non-Gaussian and asymmetric nature [6–8]. Recently, the results of several works have shown that other alternative distributions such as Beta-Liouville [9, 10], Dirichlet [11, 12], inverted Dirichlet [13], generalized Dirichlet [14] could outperform the Gaussian and be a better choice for data clustering in different applications.

In this thesis, we propose multivariate Beta as the main distribution for our mixture model. This distribution is a multivariate generalization of the bivariate Beta distribution. Indeed, we explore multivariate Beta mixture to model directly the data due to its flexibility, various shapes and promising attributes that can be then considered as an alternative to Gaussian distribution [15]. The results of previous works have confirmed the convincing performance of this mixture models in different data mining applications such as object detection, image categorization, outlier detection and medical applications [16–18].

Deploying mixture models involves two challenging aspects. The first one is the learning of model parameters and the second one is the estimation of the model complexity or the selection of the number of components which best describes the data without overfitting or underfitting [3]. To tackle the first problem, several approaches have been developed. One categorization of these approaches can be deterministic and Bayesian methods. In deterministic approaches, the inference is based on the maximum likelihood (ML) estimation, using the well-known expectation maximization (EM) algorithm. This technique has been widely used due to its simplicity, ease of use, and low computational complexity. However, EM algorithm suffers from several drawbacks such as convergence to local maxima instead of global maximum, dependency on initialization and overfitting problems [19,20]. With the improvement of computational methods, Bayesian inference as a particular approach in statistical inference can be suggested to overcome previous drawbacks. Moreover, these computational methods have been recently incorporated in many machine learning applications because of their capabilities and more accurate results compared to EM. The idea behind Bayesian method is to derive properties of probability distribution from data using Bayes' theorem. Indeed, we use our prior beliefs about parameters and update them using knowledge extracted from the observations to obtain posterior probability [21]. Bayesian approaches are based on sampling techniques and Markov Chain Monte Carlo (MCMC) is commonly used as a means of Bayesian inference to draw samples from probability distribution. Thus in this thesis, first we introduce a Bayesian inference framework based on MCMC algorithm, where Gibbs sampling within Metropolis-Hastings is applied for the problem of estimating the parameters of the finite multivariate Beta mixture model. Moreover, we propose a fully Bayesian

approach based on birth-death MCMC, which simultaneously performs the estimation of model parameters and model selection. Finally, a nonparametric Bayesian framework is developed in this thesis by extending our finite model to infinity using a mixture of Dirichlet processes. This method will address the second major challenge in deploying mixture model; i.e., determining the accurate number of clusters.

## 1.1 Contribution

The major contributions of this thesis are as follows:

- ☞ **Bayesian learning of finite multivariate Beta mixture models:**

  We present a Bayesian analysis of finite multivariate Beta mixture model and propose a solution for the problem of estimating parameters using MCMC technique. We exploit Gibbs sampling within Metropolis-Hastings for Monte Carlo simulation. We also obtained prior distribution which is a conjugate for multivariate Beta. The performance of our proposed method is evaluated via challenging applications, including cell image categorization and network intrusion detection. This contribution has been published in *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)* [22].

- ☞ **Birth-death MCMC approach for multivariate Beta mixture models:**

  We propose a Bayesian method based on the birth-death MCMC for multivariate Beta mixture model. It allows automatic and simultaneous estimation of the parameters and model selection by constructing birth and death moves. The effectiveness of the proposed approach is evaluated using real-world medical applications. This work has been submitted to the *34th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems.*

- ☞ **A nonparametric Bayesian framework for multivariate Beta mixture models:**

  We extend the finite multivariate Beta mixture model to the infinite case through a nonparametric Bayesian framework namely Dirichlet process.

Infinite model addresses the model selection problem; i.e., determination of the number of components that best describe data and and allows simultaneous data clustering. This work has been submitted to *Journal of Annals of Mathematics and Artificial Intelligence (Springer)* and is in review process of the revised version.

## 1.2 Thesis Overview

The rest of this thesis is organized as follows:

- In chapter 2, we introduce the multivariate Beta mixture model. In addition, a Bayesian learning framework for our finite mixture model is presented based on Gibbs sampling within Metropolis-Hastings algorithm where we develop the conjugate prior and show the results of our proposed model on real-world applications.

- In Chapter 3, we propose birth-death MCMC algorithm as a fully Bayesian approach for multivariate Beta mixture model with unknown number of components. We show the capability of our model in data clustering and finding the number of components through medical applications.

- Chapter 4 is devoted to infinite mixture model of multivariate Beta distributions by applying nonparametric Bayesian estimation and inference techniques. We demonstrate the effectiveness of the proposed approach via a set of challenging applications and our achieved results are compared to other different methods.

- In chapter 5, we briefly summarize our contributions.

# Chapter 2

# Bayesian Learning of Finite Multivariate Beta Mixture Models

In this chapter, we present a Bayesian approach to analyze finite multivariate Beta mixture models. Bayesian approaches are based on sampling techniques and Markov Chain Monte Carlo (MCMC) is commonly used as a means of Bayesian inference to draw samples from probability distribution. Gibbs sampling and Metropolis-Hastings are specific cases of MCMC class which allow to simulate samples from complex posterior distributions over parameters and their stochastic nature, prevents the problem of local maxima [19, 23]. Hence, we introduce a Bayesian framework based on MCMC algorithm, where Gibbs sampling within Metropolis-Hastings is applied for the problem of estimating the model parameters. We evaluate the performance of our proposed model on two real-world and challenging applications including cell image categorization and network intrusion detection. For each application, the results are compared with Bayesian Gaussian mixture model in terms of data clustering.

## 2.1 Model Specification

In this section, first we introduce multivariate Beta distribution. Then, we explain how finite mixture model is constructed based on this distribution.

### 2.1.1 Multivariate Beta Distribution

The multivariate Beta (MB) distribution is constructed by generalization of the bivariate Beta distribution to $D$ variate distribution. This distribution, as proposed in [24, 25], has the capability to model non-Gaussian data due to its flexibility and various shapes. To describe it, lets assume that a data point, originated from a MB distribution is a $D$-dimensional vector, $\vec{X}_i = \big(x_{i1}, \ldots, x_{id}\big)$, such that $0 < x_{id} < 1$. The shape parameters of this distribution are $\vec{\alpha}_j = \big(\alpha_{j1}, \ldots, \alpha_{jD}\big)$, such that $\alpha_{jd} > 0$ for $d = 1, \ldots, D$ and $|\alpha_j| = \sum_{d=1}^{D} \alpha_{jd}$. The joint density function of this observation $p(\vec{X}_i|\vec{\alpha}_j)$ is:

$$p(\vec{X}_i \mid \vec{\alpha}_j) = c \frac{\prod_{d=1}^{D} x_{id}^{\alpha_{jd}-1}}{\prod_{d=1}^{D}(1 - x_{id})^{(\alpha_{jd}+1)}} \Big[1 + \sum_{d=1}^{D} \frac{x_{id}}{(1 - x_{id})}\Big]^{-|\alpha_j|} \tag{1}$$

where

$$c = \frac{\Gamma(\alpha_{j1} + \cdots + \alpha_{jD})}{\Gamma(\alpha_{j1}) \ldots \Gamma(\alpha_{jD})} = \frac{\Gamma(|\ \alpha_j\ |)}{\prod_{d=1}^{D} \Gamma(\alpha_{jd})} \tag{2}$$

$\Gamma(.)$ denotes the Gamma function. Examples of MB distribution with different shape parameters are shown in Fig. 1.

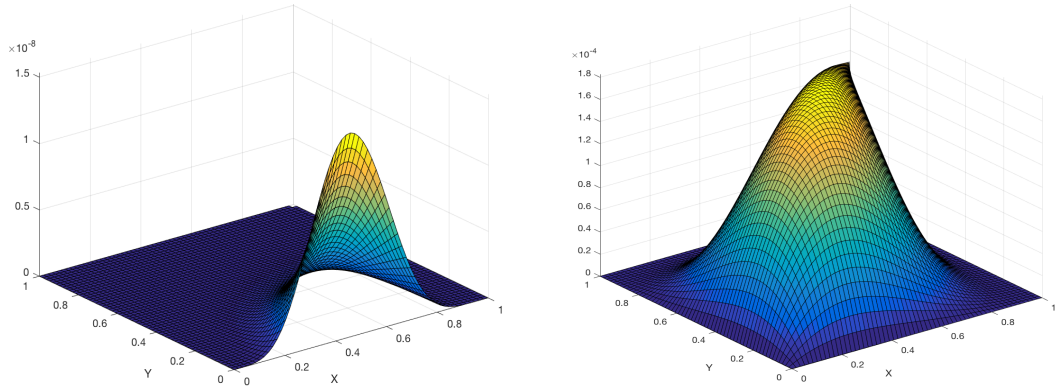



Figure 1: Two examples of multivariate Beta distribution

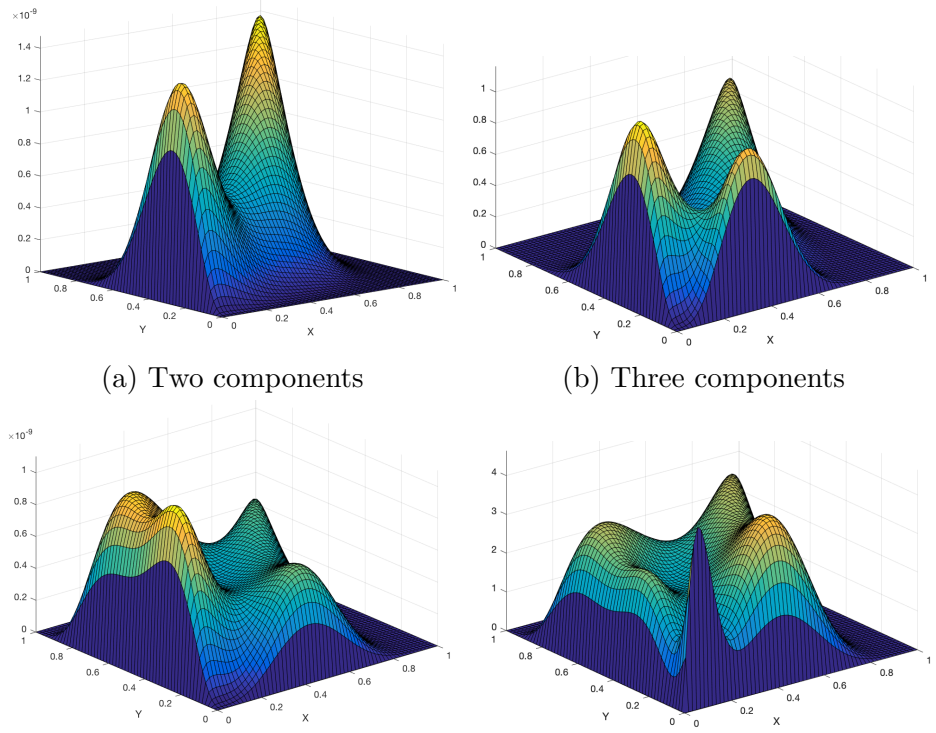
### 2.1.2 Finite Multivariate Beta Mixture Model

Lets consider $\mathcal{X} = \{\vec{X}_1, \ldots, \vec{X}_N\}$ be a data set containing $N$ $D$-dimensional vectors, independent and identically distributed, such that each vector $\vec{X}_i = \big(x_{i1}, \ldots, x_{id}\big)$ is originated from a finite but unknown MB mixture model. The likelihood of a finite

mixture of MB distributions with $M$ components, represented by $\mathcal{X}$ is denoted by:

$$p(\mathcal{X} \mid \vec{P}, \vec{\alpha}) = \prod_{i=1}^{N} \sum_{j=1}^{M} p_j p(\mathcal{X} \mid \vec{\alpha}_j) \tag{3}$$

In this equation, $\vec{\alpha_j} = (\alpha_{j1}, \ldots, \alpha_{jD})$ for $j = 1, \ldots, M$ are the shape parameters for $j^{th}$ component. The complete set of parameters are $\vec{\alpha} = (\vec{\alpha}_1, \ldots, \vec{\alpha}_M)$ as shape parameters and $\vec{P} = (p_1, \ldots, p_M)$ as mixing weights with two following conditions: $\sum_{j=1}^{M} p_j = 1$ and $p_j >= 0$.
Figure 2 illustrates four examples of this mixture model with multiple components.



(a) Two components (b) Three components

(c) Four components (d) Five components

Figure 2: Examples of multivariate Beta mixture models

## 2.2 Bayesian Learning Framework

A challenging issue in the learning phase of mixture models, is the estimation of model parameters. Generally the estimation is based on the EM algorithm and maximization of the likelihood of the data [26] by introducing the latent indicator variable

$\mathcal{Z} = \{\vec{Z}_1, \ldots, \vec{Z}_N\}$. For each observation $\vec{X}_i$, $\vec{Z}_i = (Z_{i1}, \ldots, Z_{iM})$ indicates which component it belongs to [15], such that:

$$Z_{ij} = \begin{cases} 1 & \text{if } \vec{X}_i \text{ belongs to component } j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

In other words, if $\vec{X}_i$ has the highest probability of being in cluster $j$, then $Z_{ij} = 1$ and for other clusters $Z_{ij} = 0$. Thus, by considering the membership vectors for $\mathcal{X}$, we have a complete form of data as $(\mathcal{X}, \mathcal{Z})$ which follows $p(\mathcal{X}, \mathcal{Z} \mid \Theta)$. The symbol $\Theta = (\vec{\alpha}, \vec{P})$ indicates the entire set of parameters. The complete likelihood function is then obtained by combining Eq. 3 and Eq. 4:

$$p\big(\mathcal{X}, \mathcal{Z} \mid \Theta\big) = \prod_{i=1}^{N} \prod_{j=1}^{M} \Big( p\big(\vec{X}_i \mid \vec{\alpha}_j\big) p_j \Big)^{Z_{ij}} \tag{5}$$

Among various deterministic and stochastic approaches for the problem of learning model parameters, we propose a Bayesian framework for MB mixture model due to its remarkable properties and advantages over the likelihood-based approaches. The main difference between Bayesian and likelihood-based methods is that in Bayesian inference, we incorporate our prior knowledge with the likelihood of data, to determine the final posterior probability. The prior belief about parameter $\Theta$, is specified in a prior distribution $p(\Theta)$. This means that here we consider the model parameters as random variables and our final goal is to estimate the distribution over the parameters rather than a single set of parameters. This relationship can be shown using well-known Bayes theorem:

$$p(\Theta \mid \mathcal{X}, \mathcal{Z}) = \frac{p(\mathcal{X}, \mathcal{Z} \mid \Theta) p(\Theta)}{\int p(\mathcal{X}, \mathcal{Z} \mid \Theta) p(\Theta)} \propto p(\mathcal{X}, \mathcal{Z} \mid \Theta) p(\Theta) \tag{6}$$

The proposed Bayesian learning framework is based on estimating the posterior distribution of the mixture model using MCMC techniques. By having the posterior distribution, we are able to simulate $\Theta \sim p(\Theta \mid \mathcal{X}, \mathcal{Z})$, with the help of most commonly used simulation techniques, namely Gibbs sampling [19]. This method updates each parameters in turn from its conditional posterior distribution. Then we combine Gibbs sampler with Metropolis-Hastings algorithm, where it leads to a flexible solution and convincing performance.

Accordingly, by taking a missing multinomial variable $\vec{Z}_i$ into account for each $\vec{X}_i$, such that $\vec{Z}_i \sim \mathcal{M}(1; \hat{Z}_{i1}, \ldots, \hat{Z}_{iM})$, we have:

$$\hat{Z}_{ij} = \frac{p\left(\vec{X}_i|\vec{\alpha_j}\right) p_j}{\sum_{j=1}^{M} p\left(\vec{X}_i|\vec{\alpha_j}\right) p_j} \tag{7}$$

The density function of the mixing weights is independent of $\mathcal{X}$, so it can be defined as [27]:

$$\begin{aligned} p(\vec{P} \mid \mathcal{Z}, \mathcal{X}) &= p(\vec{P} \mid \mathcal{Z}) \\ p(\vec{P} \mid \mathcal{Z}) &\propto p(\mathcal{Z} \mid \vec{P}) p(\vec{P}) \end{aligned} \tag{8}$$

For the mixing weight parameters $p_j$, considering the nature of that ($0 < p_j < 1$ and $\sum_{j=1}^{M} p_j = 1$), the natural choice of the prior is Dirichlet distribution [28], which is defined by Eq. 9, where $\eta = (\eta_1, \ldots, \eta_M)$ is the Dirichlet distribution's parameter vector:

$$p(\vec{P}) = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} p_j^{\eta_j - 1} \tag{9}$$

Moreover we have:

$$\begin{aligned} p(\mathcal{Z} \mid \vec{P}) &= \prod_{i=1}^{N} p(Z_i \mid \vec{P}) = \prod_{i=1}^{N} p_1^{Z_{i1}}, \ldots, p_M^{Z_{iM}} \\ &= \prod_{i=1}^{N} \prod_{j=1}^{M} p_j^{Z_{ij}} = \prod_{j=1}^{M} {p_j}^{n_j} \end{aligned} \tag{10}$$

where $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_{ij}=j}$. Hence, with having the information in Eq. 9 and Eq. 10, the posterior is defined as:

$$\begin{aligned} p(\vec{P} \mid \mathcal{Z}) &= \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} p_j^{\eta_j - 1} \prod_{j=1}^{M} p_j^{n_j} \\ &= \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} P_j^{\eta_j + n_j - 1} \\ &\propto \mathcal{D}(\eta_1 + n_1, \ldots, \eta_M + n_M) \end{aligned} \tag{11}$$

where $\mathcal{D}$ is a Dirichlet distribution with parameters $(\eta_1 + n_1, \ldots, \eta_M + n_M)$.

In order to find a proper prior $p(\vec{\alpha_j})$, for the shape parameter of MB mixture model $(\vec{\alpha_j})$, we consider the fact that MB distribution belongs to the exponential

family. In fact, if a S-parameter density $p$ belongs to the exponential family and we assume $\theta$ as its distribution parameter, then we have the form of [29, 30]:

$$p(\vec{X} \mid \theta) = H(\vec{X}) exp\big[\sum_{l=1}^{S} G_l(\theta) T_l(\vec{X}) + \Phi(\theta)\big] \tag{12}$$

For a vector of parameters, $T(\vec{X})$ is called the natural sufficient statistic. In this case, a conjugate prior on $\theta$ is given by following equation where $\rho = (\rho_l, \ldots, \rho_S) \in \mathbb{R}^S$ and $\kappa > 0$ are hyperparameters:

$$p(\theta) \propto exp\big(\sum_{l=1}^{S} \rho_l G_l(\theta) + \kappa \Phi(\theta)\big) \tag{13}$$

By writing MB in an exponential density form, we have:

$$\begin{aligned} p(\vec{X}_i \mid \vec{\alpha_j}) = exp\Bigg[ & \log\Gamma(\mid \alpha_j \mid) - \sum_{d=1}^{D} \log\Gamma(\alpha_{jd}) + \sum_{d=1}^{D} \alpha_{jd} \log X_{id} \\ & - \sum_{d=1}^{D} \log X_{id} - \sum_{d=1}^{D} \alpha_{jd} \log(1 - X_{id}) \\ & - \sum_{d=1}^{D} \log(1 - X_{id}) - \mid \alpha_j \mid \log(1 + \sum_{d=1}^{D} \frac{X_{id}}{1 - X_{id}}) \Bigg] \end{aligned} \tag{14}$$

Then, by letting

$$\begin{aligned} & \Phi(\vec{\alpha_j}) = \log\Gamma(\mid \alpha_j \mid) - \sum_{d=1}^{D} \log\Gamma(\alpha_{jd}) \\ & H(\vec{X}) = exp\big[ - \sum_{d=1}^{D} \log X_{id} - \sum_{d=1}^{D} \log(1 - X_{id})\big] \\ & G'_d(\vec{\alpha_j}) = \alpha_{jd} \quad d = 1, \ldots, D \quad T'_d(X) = \log X_{id} \\ & G''_d(\vec{\alpha_j}) = -\alpha_{jd} \quad d = 1, \ldots, D \quad T''_d(X) = \log(1 - X_{id}) \\ & G(\vec{\alpha_j}) = - \mid \alpha_j \mid \quad T(X) = \log(1 + \sum_{d=1}^{D} \frac{X_{id}}{1 - X_{id}}) \end{aligned} \tag{15}$$

The prior distribution is thereby as follows, with hyperparameters $(\rho'_d, \rho''_d, \rho, \kappa)$ for

$d = 1, \ldots, D$:

$$p(\vec{\alpha_j}) \propto exp\Big[\sum_{d=1}^{D} \rho'_d \alpha_{jd} - \sum_{d=1}^{D} \rho''_d \alpha_{jd} - \rho \mid \alpha_j \mid + \kappa\Big(\log\Gamma(\mid \alpha_j \mid) - \sum_{d=1}^{D} \log\Gamma(\alpha_{jd})\Big)\Big] \tag{16}$$

Having the prior for $\vec{\alpha_j}$, we can determine the posterior distribution as follows:

$$\begin{aligned} p(\vec{\alpha_j} \mid \mathcal{Z}, \mathcal{X}) &\propto p(\vec{\alpha_j}) \prod_{Z_{ij}=1} p(\vec{X_i} \mid \vec{\alpha_j}) \\ &\propto exp\Big[\sum_{d=1}^{D} \big(\rho'_d + \sum_{Z_{ij}=1} \log X_{id}\big)\alpha_{jd} + \sum_{d=1}^{D} \big(\rho''_d + \sum_{Z_{ij}=1} \log(1 - X_{id})\big)\alpha_{jd} \\ &+ \Big(\rho + \sum_{Z_{ij}=1} \log\big[1 + \sum_{d=1}^{D} \frac{X_{id}}{1 - X_{id}}\big]\Big) \mid \alpha_j \mid \\ &+ (\kappa + n_j)\Big(\log\Gamma(\mid \alpha_j \mid) - \sum_{d=1}^{D} \log\Gamma(\alpha_{jd})\Big)\Big] \end{aligned} \tag{17}$$

Since the prior and posterior distributions have the same form, we conclude that $p(\vec{\alpha_j})$ is a conjugate prior on $\vec{\alpha_j}$. We choose the hyperparameters fixed at: $\eta_j = 1, j = 1, \ldots, M$ and $\rho'_d, \rho''_d, \rho = 1$, for $d = 1, \ldots, D$ and $\kappa = 1$.

Having all the posterior probabilities in hand, we can apply Gibbs sampler for our mixture model. It simulates each parameter from its posterior successively, given the previously sampled values. The standard Gibbs sampling is described in the following steps [27]:

1. Initialization
2. Step t: For $t = 1, \ldots$
   (a) Generate $\vec{Z}_i^{(t)} \sim \mathcal{M}(1; \hat{Z}_{i1}^{(t-1)}, \ldots, \hat{Z}_{iM}^{(t-1)})$
   (b) Compute $n_j^{(t)} = \sum_{i=1}^{N} \mathbb{I}_{Z_{ij}^{(t)}=j}$
   (c) Generate $\vec{P}^{(t)}$ from (11)
   (d) Generate $\vec{\alpha_j}^{(t)}$ for $j = (1, \ldots, M)$ from (17) using the Metropolis-Hastings (M-H) algorithm.

where $\mathcal{M}(1; \hat{Z}_{i1}^{(t-1)}, \ldots, \hat{Z}_{iM}^{(t-1)})$ denotes a Multinomial distribution of order one with parameters $(\hat{Z}_{i1}^{(t-1)}, \ldots, \hat{Z}_{iM}^{(t-1)})$. In order to simulate from $\vec{\alpha_j}$ posterior distribution, Metropolis-Hastings method (M-H) is suggested [31]. It is used to avoid direct sampling of mixture parameters. For a specific iteration $t$, the steps of the M-H algorithm are as follows [32]:

1. Generate $\tilde{\alpha}_{jd} \propto q(\vec{\alpha_j} \mid \vec{\alpha_j}^{(t-1)})$ and U $\propto \mathcal{U}[0,1]$
2. Compute $r = \frac{p(\tilde{\alpha_j}|\mathcal{Z},\mathcal{X})q(\vec{\alpha_j}^{(t-1)}|\tilde{\alpha_j})}{p(\vec{\alpha_j}^{(t-1)}|\mathcal{Z},\mathcal{X})q(\tilde{\alpha_j}|\vec{\alpha_j}^{(t-1)})}$
3. If $r < u$ then $\vec{\alpha_j}^{(t)} = \tilde{\alpha_j}$ else $\vec{\alpha_j}^{(t)} = \vec{\alpha_j}^{(t-1)}$

The important issue related to this algorithm is the choice of proposal distribution. Since all $\tilde{\alpha}_{jd} > 0$, $d = 1, \ldots, D$, we considered a random walk M-H with the following proposal distribution:

$$\tilde{\alpha}_{jd} \sim \mathcal{LN}(\log(\alpha_{jd}^{(t-1)}), \sigma^2) \tag{18}$$

where $\mathcal{LN}(\log(\alpha_{jl}^{(t-1)}), \sigma^2)$ is the log-normal distribution with mean $\log(\alpha_{jl}^{(t-1)})$ and variance $\sigma^2$. Note that Eq. 18 is equivalent to:

$$\log(\tilde{\alpha}_{jd}) = \log(\alpha_{jd}^{(t-1)}) + \epsilon_1 \tag{19}$$

and $\epsilon_1 \sim \mathcal{N}(0, \sigma^2)$. In the second step of the M-H algorithm, an acceptance ratio $r$ needs to be calculated in order to make a decision whether the new samples at iteration $t$ should be accepted or rejected for the next iteration. Having the proposal distribution, the random walk M–H algorithm is composed of the following steps:

1. Generate $\tilde{\alpha}_{jd} \sim \mathcal{LN}(\log(a_{jd}^{(t-1)}), \sigma^2)$, $d = 1, \ldots, D$ and U $\propto \mathcal{U}[0,1]$
2. Compute:

$$\begin{aligned} r &= \frac{p(\tilde{\alpha_j} \mid \mathcal{Z}, \mathcal{X}) \prod_{d=1}^{D} \mathcal{LN}((\alpha_{jd}^{(t-1)}) \mid \log(\tilde{\alpha}_{jd}), \sigma^2)}{p(\vec{\alpha_j}^{(t-1)} \mid \mathcal{Z}, \mathcal{X}) \prod_{d=1}^{D} \mathcal{LN}(\tilde{\alpha}_{jd} \mid \log(\alpha_{jd}^{(t-1)}), \sigma^2)} \\ &= \frac{p(\tilde{\alpha_j} \mid \mathcal{Z}, \mathcal{X}) \prod_{d=1}^{D} \tilde{\alpha}_{jd}}{p(\vec{\alpha_j}^{(t-1)} \mid \mathcal{Z}, \mathcal{X}) \prod_{d=1}^{D} \alpha_{jd}^{(t-1)}} \end{aligned}$$

3. If $r < u$ then $\vec{\alpha_j}^{(t)} = \tilde{\alpha_j}$ else $\vec{\alpha_j}^{(t)} = \vec{\alpha_j}^{(t-1)}$

To summarize the complete Bayesian learning method proposed in this chapter, algorithmic version of that based on M-H-within-Gibbs sampler is consolidated in the following algorithm:

**Algorithm 1** M-H-within-Gibbs sampling Algorithm

1. Initialization
   (a) Apply K-means algorithm
   (b) initialize $\vec{\alpha_j}$ for each component $j$

   **Repeat**

2. Gibbs Sampling
   (a) Generate $\vec{Z}_i^{(t)} \sim \mathcal{M}(1; \hat{Z}_{i1}^{(t-1)}, \ldots, \hat{Z}_{iM}^{(t-1)})$
   (b) Compute $n_j^{(t)} = \sum_{i=1}^{N} \mathbb{I}_{Z_{ij}^{(t)}=j}$
   (c) Generate $\vec{P}^{(t)}$ from (11)
3. Metropolis-Hastings
   (a) Generate $\tilde{\alpha}_{jd} \propto q(\vec{\alpha_j} \mid \vec{\alpha_j}^{(t-1)})$ and U $\propto \mathcal{U}[0,1]$
   (b) Compute $r = \frac{p(\tilde{\alpha_j}|\mathcal{Z},\mathcal{X})q(\vec{\alpha_j}^{(t-1)}|\tilde{\alpha_j})}{p(\vec{\alpha_j}^{(t-1)}|\mathcal{Z},\mathcal{X})q(\tilde{\alpha_j}|\vec{\alpha_j}^{(t-1)})}$
   (c) If $r < u$ then $\vec{\alpha_j}^{(t)} = \tilde{\alpha_j}$ else $\vec{\alpha_j}^{(t)} = \vec{\alpha_j}^{(t-1)}$

   **until** Convergence of parameters.

## 2.3 Experimental Results

In this section, we evaluate the performance of our proposed multivariate Beta mixture model with Bayesian approach (BMBMM) by testing it on two real-world applications, namely cell image categorization and network intrusion detection. Moreover, we compare its effectiveness with Bayesian Gaussian mixture model (BGMM). It should be noted that as an important step of the preprocessing in our algorithm, we use min-max normalization (Eq. 20) since one of the assumptions of MB distribution is that the values of all observations are positive and less than one.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{20}$$

In addition, it should be pointed out that our experiments are based on clustering with no training step as entire data is given into the algorithm with no prior knowledge about the labels. For this, we first removed the original labels of datasets and with

the help of our proposed clustering model, we found the predicted labels of each observation. Then, the accuracy is measured by confusion matrix, comparing the predicted labels with true ones. In order to assess our model performance against other method, we use standard metrics based on confusion matrix which are defined as follows:

$$Accuracy = \frac{TP + TN}{Total\ no\ of\ observations} \qquad Precision = \frac{TP}{TP + FP} \tag{21}$$
$$Recall = \frac{TP}{TP+FN} \qquad F1\text{-}score = 2 \times \frac{precision \times recall}{precision + recall}$$

where TP, TN, FP, and FN represent the total number of true positives, true negatives, false positives, and false negatives respectively.

### 2.3.1 Cell Image Categorization

Pattern recognition methods have achieved remarkable attention in medicine and biomedical researches and have been successfully applied across various problem domains, such as cell image classification. These machine learning algorithms provide the ability to differentiate various types of cells based on their biological behaviors and characteristics. These automated methods could be applied in cell biology and screening experiments for the diagnosis and prognosis of diseases from label-free cellular images obtained from an optical microscope.

Malaria is an infectious and life-threatening disorder caused by Plasmodium parasites that are transmitted by the bites of infected types of mosquitoes. Based on WHO Malaria facts, 219 million cases of Malaria and 435,000 deaths associated with this disease were globally reported in 2017 [33]. This is a motivation to make an accurate diagnosis and early detection in order to decrease morbidity and mortality. Malaria parasites can be identified by examining blood smear to find the cells infected with malaria which are identified by the small clot inside the cellular images in contrast with uninfected cells which are clean without any clot. Figure 3 illustrates 8 image samples of cells.

In our experiment, we used NIH Malaria dataset [34]. We examined 159 images including 79 uninfected cells and 80 parasitized cells from the thin blood smear slide images of segmented cells. Scale-Invariant Feature Transform (SIFT) [35] and Bag of Visual Words (BoVW) are used to extract important features from these images.

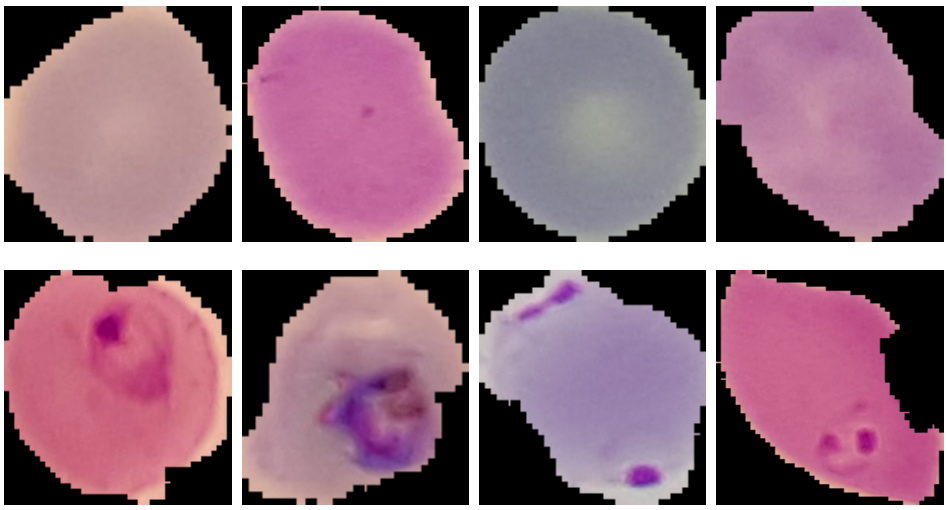


Figure 3: Some random images of Malaria cell database. Top row samples are uninfected and bottom row samples are parasitized blood smear.

After normalizing data in the first step, K-means is applied to obtain $M$ clusters. Then for each component $j$, we initialize the parameters $\vec{\alpha_j}$. For evaluating the performance of our cluster analysis, we use standard metrics such as accuracy, precision, recall and F1-score.

The confusion matrices of our proposed Bayesian model and BGMM are shown in Table 1. Actual labels and predicted labels are denoted by (AC) and (P), respectively. Table 2 reveals the comparison between different results of performance metrics for both models which shows that BMBMM outperforms BGMM.

Table 1: Confusion matrices for Malaria cells

**BMBMM**

| | Healthy (P) | Infected (P) |
|---|---|---|
| Healthy (AC) | 70 | 9 |
| Infected (AC) | 11 | 69 |

**BGMM**

| | Healthy (P) | Infected (P) |
|---|---|---|
| Healthy (AC) | 72 | 7 |
| Infected (AC) | 15 | 65 |

Table 2: Model performance results for Malaria images

| Model | Accuracy(%) | Precision(%) | Recall(%) | F1-score(%) |
|---|---|---|---|---|
| BMBMM | 87.42 | 88.46 | 86.25 | 87.34 |
| BGMM | 86.16 | 90.27 | 81.25 | 85.53 |

### 2.3.2 Intrusion Detection

Network security as part of cyber-security systems has recently become an extremely important issue for users and service providers, while a huge amount of devices can connect to the internet and communicate through the network. Nevertheless, a large variety of attacks, mobile threats and intrusion attempts can be occurred in many networking environments and computing facilities. These anonymous and malicious activities may cause network crashes, communication failure and loss of sensitive data. To that end, detecting and preventing such activities should be a mandate in current computer networks. Intrusion Detection Systems (IDSs) are developed to discover and detect any abnormal actions, such as identify unauthorized access, alteration and destruction within the networks by monitoring the network traffics permanently [36].

According to the constant changes in network connections and environments, intelligent and reliable intrusion detection systems should be designed to cover dynamic patterns and behavior of the networks. Machine learning and data mining based solutions can be leveraged to create robust and effective IDS. These systems are able to analyze and learn from the existing traffic patterns to accurately predict the upcoming traffic and the behaviors of users [37]. Supervised machine learning algorithms have been proposed to detect abnormal activities (anomaly) considering the normal network behavior and already known intrusion scenarios. Such supervised algorithms suffer from significant drawbacks, where they are not robust to network traffic changes and newly founded intrusions. Since they are trained on historical data under predefined patterns, they may fail to perform accurately due to overfitting problems. On the other hand, unsupervised machine learning based solutions can provide flexibility and robustness in accurate intrusion detection and adapt themselves with the latest network behavior [38].

In this paper, the *NSL-KDD* data [39], a refined version of famous *KDD-CUP'99* intrusion detection dataset is employed. We used 20% of whole data which contains

25,192 entries. The data has two categories of 13449 normal and 11743 anomaly instances. All various attacks such as *DoS*, *Probe*, *R2L* and *U2R* are grouped together in anomaly class. Each connection has 41 quantitative and qualitative features. In the preprocessing step, we need to convert all categorical attributes to numerical before normalizing the data. The confusion matrices obtained by using our proposed model and BGMM are shown in Tables 3.

Table 3: Confusion matrices for intrusion detection dataset

**BMBMM**

| | Normal (P) | Anomaly (P) |
|---|---|---|
| Normal (AC) | 10617 | 2832 |
| Anomaly (AC) | 2103 | 9640 |

**BGMM**

| | Normal (P) | Anomaly (P) |
|---|---|---|
| Normal (AC) | 10895 | 2554 |
| Anomaly (AC) | 4191 | 7552 |

Four performance measures for BMBMM and BGMM are represented in Table 4.

Table 4: Model performance results for intrusion detection

| Model | Accuracy(%) | Precision(%) | Recall(%) | F1-score(%) |
|---|---|---|---|---|
| BMBMM | 80.41 | 77.29 | 82.09 | 79.62 |
| BGMM | 73.22 | 74.72 | 64.31 | 69.13 |

Statistics in Table 4 reveal the fact that comparing to BGMM, BMBMM provides higher accuracy, precision, recall anf F1-score indicate that BMBMM outperforms BGMM. Additionally, another important metric in binary classification is False Negative Rate ($FNR = \frac{FN}{FN+TP}$). This ratio shows the incorrect identification of anomaly. It means classifying data as normal incorrectly, which is in fact the attack. For our model, FNR is 17.9% which is significantly lower than the result provided by BGMM (35.69%), showing the efficiency and capability of MB mixture model in detecting intrusions.

# Chapter 3

# Birth-Death MCMC Approach for Multivariate Beta Mixture Models

In this chapter, we present birth-death MCMC (BDMCMC) approach for MB mixture model. It is based on continuous time birth-death events which simultaneously performs the estimation of model parameters and model selection. Moreover, we present the experimental results obtained from two real-world medical applications namely heart failure detection and Thyroid disease detection to demonstrate the effectiveness of the proposed algorithm.

## 3.1 Bayesian Inference via BDMCMC

In this section, we propose our Bayesian framework for parameters estimation of our mixture model and determining the proper number of mixture components as two major challenges in mixture modeling. Concerning parameters estimation, we employed Bayesian learning and with the help of the most commonly used MCMC techniques, namely Gibbs sampling and Metropolis–Hastings algorithm, the posterior distribution of the mixture model can be approximated.

Another challenge in the context of the mixture of distributions is model selection, for which several approaches have been suggested within Bayesian inference such as Bayes factors, Bayesian Information Criterion (BIC), reversible jump MCMC (RJMCMC) and birth and death processes [40–43]. For variable dimension problems, a common approach in literature is RJMCMC, where it consists of split/combine and

birth/death moves for changing the number of components. However, the extension of this method to multivariate cases is difficult. Therefore, a birth-death MCMC (BDMCMC) algorithm has been introduced in [40] which we use in this chapter for learning of MB mixture model. It is based on generating an ergodic Markov Chain with the joint posterior distribution of the parameters and the model as its stationary distribution, where the number of components is considered as unknown. It is also proved that RJMCMC converges to BDMCMC under certain conditions [44]. Moreover, the results of several studies based on this algorithm have shown convincing performance in the case of mixture of various distributions such as gamma, Dirichlet and Beta [45–47].

### 3.1.1 Priors and Posteriors

In fully Bayesian framework, the unknown number of components $M$ with the other parameters of the model $(\vec{P}, \vec{\alpha})$ are regarded as random variables drawn from some prior distributions that we have to determine. The joint distribution of all these variables is:

$$\begin{aligned} p(M, \vec{P}, Z, \vec{\alpha}, \mathcal{X}) =& p(M)p(\vec{P} \mid M)p(Z \mid \vec{P}, M) \\ & p(\vec{\alpha} \mid Z, \vec{P}, M)p(\mathcal{X} \mid \vec{\alpha}, Z, \vec{P}, M) \end{aligned} \tag{22}$$

Following [42], by imposing common conditional independencies, the joint distribution can be written as following:

$$p(M, \vec{P}, Z, \vec{\alpha}, \mathcal{X}) = p(M)p(\vec{P} \mid M)p(Z \mid \vec{P}, M)p(\vec{\alpha} \mid M)p(\mathcal{X} \mid \vec{\alpha}, Z) \tag{23}$$

Then, the main goal of the Bayesian inference is to create realizations from the conditional joint density $p(M, \vec{P}, Z, \vec{\alpha} \mid \mathcal{X})$.

One of the important steps in Bayesian learning is the choice of suitable prior distributions for the parameters of mixture model. First for the number of components $M$, we assume a truncated Poisson prior as below:

$$p(M) \propto \frac{\lambda^M}{M!}, \quad (M = 1, \ldots, M_{max}) \tag{24}$$

Assuming that the parameters of the MB are statistically independent, for the shape parameters $\vec{\alpha}$, an appealing choice as a prior is a Gamma distribution denoted by

$\mathcal{G}(.)$ that can be written as follows:

$$p(\vec{\alpha}_j) = \mathcal{G}(\vec{\alpha}_j \mid \vec{u}, \vec{v}) = \prod_{d=1}^{D} \frac{v_d^{u_d}}{\Gamma(u_d)} \alpha_{jd}^{u_d - 1} e^{-v_d \alpha_{jd}} \tag{25}$$

Here $\{u_{jd}\}$ and $\{v_{jd}\}$ are hyperparameters which have constraint such that $u_{jd} > 0$ and $v_{jd} > 0$. Having this prior in hand, the full conditional posterior distribution for $\vec{\alpha_j}$ is:

$$\begin{aligned} p(\vec{\alpha_j} \mid \dots) &\propto p(\vec{\alpha_j} \mid \vec{u}, \vec{v}) \prod_{i=1}^{N} p(\vec{X}_i \mid \Theta_{Z_i}) \\ &\propto \prod_{d=1}^{D} \frac{v_d^{u_d}}{\Gamma(u_d)} \alpha_{jd}^{u_d - 1} e^{-v_d \alpha_{jd}} \\ &\times \left[ \frac{\Gamma(\mid \alpha_j \mid)}{\prod_{d=1}^{D} \Gamma(\alpha_{jd})} \right]^{n_j} \prod_{Z_i = j} \left[ \frac{\prod_{d=1}^{D} x_{id}^{\alpha_{jd} - 1}}{\prod_{d=1}^{D} (1 - x_{id})^{(\alpha_{jd} + 1)}} \Big( 1 + \sum_{d=1}^{D} \frac{x_{id}}{(1 - x_{id})} \Big)^{-|\alpha_j|} \right] \end{aligned} \tag{26}$$

where $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_i = j}$ indicates the number of observations belonging to cluster $j$ and symbol $\mid \dots$ represent conditioning on all other variables. Moreover, for the mixing weight vector $\vec{P}$, we know that it is defined on $(p_1, \dots, p_M) : \sum_{j=1}^{M-1} p_j < 1$, then the typical prior choice is a Dirichlet distribution with parameters $\eta = (\eta_1, \dots, \eta_M)$ as following:

$$p(\vec{P} \mid M, \eta) = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} p_j^{\eta_j - 1} \tag{27}$$

Also, the prior for the latent variable $Z$ is:

$$p(Z \mid \vec{P}, M) = \prod_{j=1}^{M} p_j^{n_j} \tag{28}$$

Then, using Eqs. 27 and 28 we obtain:

$$\begin{aligned} p(\vec{P} \mid \dots) &\propto p(Z \mid \vec{P}, M) p(\vec{P} \mid M, \eta) \\ &\propto \prod_{j=1}^{M} p_j^{n_j} \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} \pi_j^{\eta_j - 1} \\ &\propto \prod_{j=1}^{M} p_j^{n_j + \eta_j - 1} \end{aligned} \tag{29}$$

which indicates a Dirichlet distribution with parameters $(\eta_1+n_1, \ldots, \eta_M+n_M)$. Moreover, recalling $p(Z_i = j) = p_j$ for $(i = 1, \ldots, N; j = 1, \ldots, M)$, we can obtain the posterior for the membership variables as follows:

$$p(Z_i = j \mid \ldots) \propto p_j p(\vec{X}_i \mid \vec{\alpha_j}) \tag{30}$$

### 3.1.2 BDMCMC Methodology

Now we follow the BDMCMC algorithm proposed in [40] to obtain a sample from the posterior distributions of the parameters. The method is designed based on generating a continuous time Markov birth-death process with considering prior for number of mixture components. In this approach, the model parameters can be considered as observations from a marked point process, where each point represents a component of the mixture [48]. In order to create an ergodic Markov chain, the mixture size, $M$, can change, where it can allow new components to be born or existing components to die. Therefore, births and deaths may happen in continuous time, where their happening rates could define the stationary distribution of the process [45].

In this process, birth of new components happen at a constant rate from the prior distribution of $M$, while the death occurs at a rate which is high for less significant components and very low for important components which describe the data well. Whenever the birth event occurs, the weight of new component is calculated from a Beta distribution with parameters $(1, M)$ and the number of components is increased. Then, in order to keep the sum of all the weights equal to unity, the old component weights are scaled down proportionally through multiplying each mixing weight by $(1 - p^*)$, where $p^*$ is the weight of the new component. On the other hand, after eliminating a component, each mixing weight is divided by $(1 - p^*)$, ($p^*$ is the weight of the removed component) [47]. A death event decreases the number of mixture components, where the death rate for each component is computed as a likelihood ratio of the model with and without that component as follows [45]:

$$\Delta_j = \prod_{i=1}^{N} \left( \frac{p\big(\vec{X}_i \mid \Theta\big) - p_j p(\vec{X}_i \mid \vec{\alpha_j})}{(1 - p_j) p\big(\vec{X}_i \mid \Theta\big)} \right), \qquad j = 1, \ldots, M \tag{31}$$

Where $p\big(\vec{X}_i \mid \Theta\big) = \sum_{j=1}^{M} p_j p\big(\vec{X}_i \mid \vec{\alpha}_j\big)$. Then, the total death rate of the process at any time is obtained by the sum of the individual death rates, i.e., $\Delta = \sum_j \Delta_j$,

for $j = 1, \ldots, M$. Since births and deaths are independent Poisson processes, the time between each birth or death occurrence is exponentially distributed with mean $1/(\Delta + \lambda)$. We assume the constant $\lambda$ from prior of $M$ in Eq. 24 for birth rate of mixture component.

The complete BDMCMC algorithm can be summarized in Algorithm 2.

**Algorithm 2** Birth-Death MCMC learning of MBMM

Initialize parameters $M^{(0)}, \vec{P}^{(0)}, \vec{\alpha}^{(0)}$

1. Begin the birth-death process for a virtual fixed time $t_0$ and let the birth rate equivalent to $\lambda$.
   (a) Calculate the death rates for each component, $\Delta_j$ , and the total death rate, $\Delta = \sum_j \Delta_j$.
   (b) Simulate the time to the next jump from an exponential distribution with mean $1/(\Delta + \lambda)$.
   (c) If the run time is lower than $t_0$ continue otherwise jump to step 2.
   (d) Simulate the type of jump: birth or death with probabilities: p(birth) = $\frac{\lambda}{\lambda+\Delta}$ , p(death)= $\frac{\Delta}{\lambda+\Delta}$
   (e) Make the adjustment for mixture components.

   **MCMC steps**

2. Update the allocation $Z^{(i+1)}$.
3. Update the mixing parameters $\vec{P}^{(i+1)}$.
4. Update the parameters $\vec{\alpha_j}^{(i+1)}$
5. Set $i = i + 1$ and iterate

The first step of this algorithm is the birth-death process, while the rests are standard Gibbs sampling moves. In order to sample from $\vec{\alpha_j}$ posterior, we exploit Metropolis-Hastings (M-H) method that we introduced and used in Chapter 2 . It is used to avoid direct sampling of mixture parameters since the full conditional distribution given by Eq. 26 is complex and does not have a well-known form.
For a specific iteration $t$, the steps of the M-H algorithm, to sample $\vec{\alpha_j}$, are as follows [32]:

1. Generate $\tilde{\alpha}_{jd} \sim q\big(\vec{\alpha_j} \mid \vec{\alpha_j}^{(t-1)}\big)$ and $u \sim \mathcal{U}_{[0,1]}$

2. Compute $r = \frac{p(\tilde{\alpha_j}|\dots)q\left(\vec{\alpha_j}^{(t-1)}|\tilde{\alpha_j}\right)}{p(\vec{\alpha_j}^{(t-1)}|\dots)q\left(\tilde{\alpha_j}|\vec{\alpha_j}^{(t-1)}\right)}$

3. If $r < u$ then $\vec{\alpha_j}^{(t)} = \tilde{\alpha_j}$ else $\vec{\alpha_j}^{(t)} = \vec{\alpha_j}^{(t-1)}$

The main issue related to this algorithm is the choice of proposal distribution. Since all $\tilde{\alpha}_{jd} > 0$, $d = 1, \dots, D$, we assumed a random walk M-H with the following proposal $\tilde{\alpha}_{jd} \sim \mathcal{LN}(\log(\alpha_{jd}^{(t-1)}), \sigma^2)$, where $\mathcal{LN}(\log(\alpha_{jl}^{(t-1)}), \sigma^2)$ is the log-normal distribution with mean $\log(\alpha_{jl}^{(t-1)})$ and variance $\sigma^2$.
In this algorithm, by producing samples from posterior distributions over the Markov Chain, the parameters of interest including $M$, can then be estimated by forming the sample path averages after a burn-in period.

## 3.2 Experimental Results

In this section, we validate the performance of our proposed BDMCMC algorithm for MB mixture model (BD-MBMM) on real-world medical tasks. We investigate its ability to estimate the mixture parameters and simultaneously select the proper number of components. We compared our proposed model with similar algorithm for GMM (BD-GMM) and present the results in comparison tables. In our experiments, we consider $M_{max} = 10$ and the effectiveness of the model is evaluated in terms of the accuracy, precision, recall and F1-score based on confusion matrix.

### 3.2.1 Heart Failure Detection

Cardiovascular diseases (CVDs) as serious health issues still remain the leading cause of death globally. In particular, heart failure is a condition caused by CVDs in which the pumping power of the heart is not sufficient to move blood and oxygen in the body. The CVDs can be preventable if the underlying main risk factors, such as high blood pressure, level of cholesterol, diabetes and stress be under control. For this purpose, medical records can be considered as useful resources for designing automatic diagnosis systems using data mining tools [49].

In this experiment, we used real-world data set obtained from the UCI Repository [50], which includes medical records of 299 patients having heart failure to evaluate our proposed model. It consists of 13 features derived from multiple medical tests,

lifestyle and body information. The data set is comprised of two target classes that imply whether patients with heart failure died or survived. The confusion matrices obtained using the models are given in Table 5. Actual labels and predicted labels are denoted by (AC) and (P), respectively. Table 6 represents the comparison between different results of performance metrics for both models which shows that BD-MBMM outperforms BD-GMM.

Table 5: Confusion matrices for heart failure

BD-MBMM

| | Survive (P) | Not Survive (P) |
|---|---|---|
| Survive (AC) | 188 | 15 |
| Not Survive (AC) | 44 | 52 |

BD-GMM

| | Survive (P) | Not Survive (P) |
|---|---|---|
| Survive (AC) | 179 | 24 |
| Not Survive (AC) | 61 | 35 |

Table 6: Model performance results for heart failure

| Model | Accuracy(%) | Precision(%) | Recall(%) | F1-score(%) |
|---|---|---|---|---|
| BD-MBMM | 80.27 | 77.61 | 54.16 | 63.8 |
| BD-GMM | 71.57 | 59.32 | 36.45 | 45.15 |

### 3.2.2 Thyroid Disease Detection

Thyroid as a primary gland produces hormones to regulate the metabolism of the body. The most common thyroid disorders can occur when the thyroid hormones are abnormal. Hyperthyroidism and hypothyroidism are the two main diseases of the thyroid that happen either by releasing too much T4 hormone or by releasing less. Most thyroid problems can be treated by early detection and proper diagnosis. In medical field, data mining has emerged to assist the healthcare experts in early detection, diagnosis and prevention of this disease [51].

In our study, we applied our model on a publicly available data set [52], which includes a sample of 215 patients. The features are the results of the five laboratory tests, namely: RT3U, T4, T3, TSH and DTSH. The data set has three classes which

indicate the diagnosis of thyroid operation as Hypo, Normal, and Hyper. The confusion matrices in Table 7 and the results presented in Table 8 illustrate the potential of our proposed model performance in this application for data clustering and finding the proper number of components.

Table 7: Confusion matrices for Thyroid disease

BD-MBMM

| | Hyper (P) | Normal (P) | Hypo (P) |
|---|---|---|---|
| Hyper (AC) | 22 | 8 | 5 |
| Normal (AC) | 0 | 144 | 6 |
| Hypo (AC) | 2 | 5 | 23 |

BD-GMM

| | Hyper (P) | Normal (P) | Hypo (P) |
|---|---|---|---|
| Hyper (AC) | 20 | 15 | 0 |
| Normal (AC) | 1 | 132 | 17 |
| Hypo (AC) | 4 | 7 | 19 |

Table 8: Model performance results for Thyroid disease

| Model | Accuracy(%) | Precision(%) | Recall(%) | F1-score(%) |
|---|---|---|---|---|
| BD-MBMM | 87.9 | 83.67 | 78.5 | 81.0 |
| BD-GMM | 79.53 | 72.83 | 69.49 | 71.12 |

# Chapter 4

# A Nonparametric Bayesian Framework for Multivariate Beta Mixture Models

In this chapter, we adapt and investigate infinite mixture model based on MB distribution. For this, we develop a nonparametric Bayesian approach by extending finite MB mixture model proposed in chapter 2 to infinity using a mixture of Dirichlet processes. Our approach relies on the estimation of the posterior distribution using Gibbs sampling and Metropolis-Hastings algorithm. Finally, the experimental results are presented to show the effectiveness of our model compare to different other models, which involves four balanced and imbalanced real-world applications namely intrusion detection, Hepatitis diagnosis, software defect categorization and image categorization.

## 4.1 The Infinite Multivariate Beta Mixture Model

Choosing the accurate number of clusters $M$ that can correctly describe the data is an important, yet challenging issue in designing mixture models. Therefore, the ability to define the number of components in advance can be considered as a major limitation in finite mixture approaches [53]. To address the before-mentioned challenge, nonparametric Bayesian approaches have been proposed in the literature, where they can automatically obtain the number of clusters according to the specific choice of

prior for mixing weights [54, 55]. Such infinitely complex models have shown remarkable theoretical and computational progress over the recent years [56]. The number of mixture components in nonparametric Bayesian approaches can be adjusted as new data arrives and is allowed to increase to infinity, where it resolves the problem of model selection. This prominent characteristic of infinite mixture models makes them more realistic approach and adaptive to the given circumstances in contrast with assuming a fixed number of components. In addition, overfitting and underfitting of mixture models can be handled through this assumption. In this context, previous works have provided acceptable and reliable performance in several applications by deploying Dirichlet process mixture model based on various probability distributions [37, 57, 58].

Here we illustrate the main idea behind Dirichlet process mixture model and its capability to create or remove clusters.

### 4.1.1 Conditional Posteriors

In Bayesian inference, one important step is defining the prior distributions. For each $\vec{\alpha_j}$, we consider the same previous approach that we employed in chapter 2 for developing conjugate prior by take into account that MB distribution belongs to exponential family of distribution. The prior distribution is thereby as follows, with hyperparameters $(\rho'_d, \rho''_d, \rho, \kappa)$ for $d = 1, \ldots, D$:

$$p(\vec{\alpha_j}) \propto exp\Bigg[\sum_{d=1}^{D} \rho'_d \alpha_{jd} - \sum_{d=1}^{D} \rho''_d \alpha_{jd} - \rho \mid \alpha_j \mid + \kappa\Bigg(\log\Gamma(\mid \alpha_j \mid) - \sum_{d=1}^{D} \log\Gamma(\alpha_{jd})\Bigg)\Bigg] \tag{32}$$

By having the prior for $\vec{\alpha_j}$, posterior distribution can be determined as follows:

$$p(\vec{\alpha_j} \mid \mathcal{Z}, \mathcal{X}) \propto p(\vec{\alpha_j}) \prod_{Z_{ij}=1} p(\vec{X_i} \mid \vec{\alpha_j}) \tag{33}$$

$$\propto exp\Bigg[ \sum_{d=1}^{D} \big(\rho'_d + \sum_{Z_{ij}=1} \log X_{id}\big)\alpha_{jd} + \sum_{d=1}^{D} \big(\rho''_d + \sum_{Z_{ij}=1} \log(1 - X_{id})\big)\alpha_{jd}$$

$$+ \left(\rho + \sum_{Z_{ij}=1} \log\big[1 + \sum_{d=1}^{D} \frac{X_{id}}{1 - X_{id}}\big]\right) \mid \alpha_j \mid$$

$$+ (\kappa + n_j)\left( \log\Gamma(\mid \alpha_j \mid) - \sum_{d=1}^{D} \log\Gamma(\alpha_{jd})\right)\Bigg]$$

For $\vec{P}$, the mixing weights coefficients, we know that it is defined on $(p_1, \ldots, p_M)$ : $\sum_{j=1}^{M-1} p_j < 1$, then a possible choice as a prior is a symmetric Dirichlet distribution with a concentration parameter $\frac{\eta}{M}$.

$$p(\vec{P} \mid \eta) = \frac{\Gamma(\eta)}{\prod_{j=1}^{M} \Gamma(\frac{\eta}{M})} \prod_{j=1}^{M} p_j^{\frac{\eta}{M}-1} \tag{34}$$

Recalling $\vec{Z_i}$ as a latent variable to show $X_i$ belongs to which cluster, such that $p_j = p(Z_{ij} = 1), j = 1, \ldots, M$, then the inference of $\vec{P}$ can be performed through the inference of $\vec{Z_i}$ [59] as follows:

$$\begin{aligned} p(\mathcal{Z} \mid \vec{P}) &= \prod_{i=1}^{N} p(\vec{Z_i} \mid \vec{P}) = \prod_{i=1}^{N} {p_1}^{Z_{i1}} \ldots {p_M}^{Z_{iM}} \\ &= \prod_{i=1}^{N} \prod_{j=1}^{M} {p_j}^{Z_{ij}} = \prod_{j=1}^{M} {p_j}^{n_j} \end{aligned} \tag{35}$$

where $\mathcal{Z} = \{\vec{Z_1}, \ldots, \vec{Z_N}\}$ and $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_{ij}=1}$ represents the number of elements that are associated with cluster $j$. As the Dirichlet is a conjugate prior to the multinomial, then we can integrate out the mixing proportions $\vec{P}$ to obtain the prior for $\mathcal{Z}$:

$$\begin{aligned} p(\mathcal{Z} \mid \eta) &= \int_{\vec{P}} p(\mathcal{Z} \mid \vec{P}) p(\vec{P} \mid \eta) d\vec{P} = \frac{\Gamma(\eta)}{\prod_{j=1}^{M} \Gamma(\frac{\eta}{M})} \int_{\vec{p}} \prod_{j=1}^{M} p_j^{n_j + \frac{\eta}{M} - 1} d\vec{p} \\ &= \frac{\Gamma(\eta)}{\Gamma(N + \eta)} \prod_{j=1}^{M} \frac{\Gamma(\frac{\eta}{M} + n_j)}{\Gamma(\frac{\eta}{M})} \end{aligned} \tag{36}$$

By combining all Eqs. 34 to 36, Eq. 37 can be written as follows:

$$p(\vec{P} \mid \mathcal{Z}, \eta) = \frac{p(\mathcal{Z} \mid \vec{P}) p(\vec{P} \mid \eta)}{p(\mathcal{Z} \mid \eta)} = \frac{\Gamma(\eta + N)}{\prod_{j=1}^{M} \Gamma(\frac{\eta}{M} + n_j)} \prod_{j=1}^{M} p_j^{n_j + \frac{\eta}{M} - 1} \tag{37}$$

which indicates a Dirichlet distribution with parameter $(n_1 + \frac{\eta}{M}, \ldots, n_M + \frac{\eta}{M})$, then the conditional prior for a single indicator is defined by [55]:

$$p(Z_{ij} = 1 \mid \eta, \mathcal{Z}_{-i}) = \frac{n_{-ij} + \frac{\eta}{M}}{N - 1 + \eta} \tag{38}$$

Where $\mathcal{Z}_{-i} = \{\vec{Z}_1, \ldots, \vec{Z}_{i-1}, \vec{Z}_{i+1}, \ldots, \vec{Z}_N\}$ , $n_{-i,j}$ is the number of observations excluding $\vec{X}_i$ in component $j$. The conditional posterior is then calculated by the product of the prior Eq. 38 and the likelihood of $\vec{X}_i$.

As we have mentioned, an important task in adopting mixture models is the selection of model's complexity. In this section we overcome this problem by considering $M \to \infty$ in Eq. 38 which gives us the following limits [60]:

$$p(Z_{ij} = 1 \mid \eta; \mathcal{Z}_{-i}) = \begin{cases} \frac{n_{-i,j}}{N-1+\eta} & \text{if } n_{-i,j} > 0 \quad (j \in \mathcal{R}) \\ \frac{\eta}{N-1+\eta} & \text{if } n_{-i,j} = 0 \quad (j \in \mathcal{U}) \end{cases} \tag{39}$$

Where $\mathcal{R}$ and $\mathcal{U}$ denote the sets of represented and unrepresented clusters, respectively. Indeed, this equation indicates each observation with a certain probability will be assigned to either the represented or an unrepresented component. In the case of represented component, the conditional prior will depend on the number of observations already allocated to this cluster, while a new component (unrepresented) is only proportional to $\eta$ and $N$ [59, 61]. Having the conditional priors in Eq. 39, then we can determine the conditional posteriors as following [54, 55]:

$$p(Z_{ij} = 1 \mid \vec{\alpha_j}, \eta; \mathcal{Z}_{-i}) = \begin{cases} \frac{n_{-i,j}}{N-1+\eta} p(\vec{X}_i \mid \vec{\alpha_j}) & \text{if } j \in \mathcal{R} \\ \int \frac{\eta p(\vec{X}_i \mid \vec{\alpha_j}) p(\vec{\alpha_j})}{N-1+\eta} d\vec{\alpha_j} & \text{if } j \in \mathcal{U} \end{cases} \tag{40}$$

This equation can be explained as Dirichlet process mixture of MB distributions. If a given observation is assigned to the unrepresented cluster, a new represented cluster will be generated accordingly, which means there is always an empty cluster to justify the infinite mixture model concept. While, if all observations within the represented cluster are assigned to other clusters due to sampling iterations, this cluster will be empty and converted to unrepresented cluster.

The proposed infinite model can be imagined as Chinese restaurant process that follows the analogy described in [62]. In this analogy, a restaurant with countably infinite tables is considered as the mixture components. During the process, the first customer (i.e., data observation) always occupies the first table. The next customer is able to choose between the first unoccupied table or an occupied table with a probability related to the number of people who are already at that table [55, 63]. A graphical model representing our infinite MB mixture is shown in Fig.4.

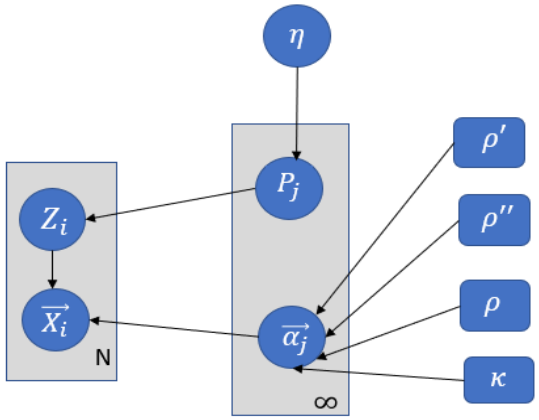



Figure 4: Graphical model representation of proposed Bayesian infinite MBMM. The random variables are in circles, and the Rounded boxes are for the fixed hyperparameters. Boxes show the process of repetition (with the number of repetitions in the lower right) and the arcs describe the conditional dependencies between the variables.

MCMC methods are used to provide estimates of the posterior distribution for infinite MBMM. These statistics-based sampling methods can appropriately generate samples from complicated distributions. Since posterior distributions in mixture models can have intractable forms, two relevant MCMC techniques called Gibbs sampling and Metropolis-Hastings (M-H) are exploited in our work. The complete proposed algorithm can be summarized in Algorithm 3.

In the initialization step, first we assume that all the observations are in the same cluster. Then, the updating of the number of represented components step is based on the previous step which is the generation of the $\vec{Z}_i$. Indeed, the number of clusters $M$, is increased by one when a sample is assigned to an unrepresented cluster, while $M$ is decreased by one if a component becomes empty during the iterations [9].
Note that, for the sampling of the vectors $\vec{Z}_i$, we need to evaluate the integral in Eq. 40, which is analytically intractable. Hence, we used the proposed approach

**Algorithm 3** Nonparametric Bayesian learning of MBMM

1. **Process**
2. initialize assignments and parameters
3. **repeat**
4. Generate $\vec{Z}_i$ from Eq. (40) and then update $n_j$, for $i = 1, \ldots, N$, $j = 1, \ldots, M$.
5. Update the number of represented components denoted by M.
6. Update the mixing parameters for the represented components by $p_j = \frac{n_j}{N+\eta}$, $j = 1, \ldots, M$
7. Update the mixing parameters $p_U = \frac{\eta}{N+\eta}$ of the unrepresented clusters.
8. Generate the mixture parameters $\vec{\alpha_j}$ from Eq. (33) for $j = 1, \ldots, M$ using Metropolis-Hastings
9. **until** Convergence

in [54, 55] for approximating this integral. This approach generates a Monte Carlo estimate by sampling from the prior of $\vec{\alpha_j}$. Moreover, in order to simulate from $\vec{\alpha_j}$ posterior distribution, M-H method is used which we earlier described its algorithm in chapter 2.

## 4.2 Experimental Results

In this section, we evaluate the performance of our infinite multivariate Beta mixture model (IMBMM) by validating it on four real-world applications, namely network intrusion detection, hepatitis diagnosis and software defect categorization and image categorization. We compare the effectiveness of our proposed IMBMM with finite MB mixture models using Bayesian approach (BMBMM), infinite Gaussian mixture model (IGMM) [55] and finite Gaussian mixture model (GMM).

For each of the experiment, we ran the algorithm with varying initial hyper-parameter values and different numbers of iterations. Then, we used the mean of summaries obtained for the last 10% of iterations for parameter estimation of our infinite model. Our specific choice for concentration parameter for each application is

$\eta = (1, 1, 1, 0.8)$, respectively. In order to assess our model performance against other methods, we use standard metrics such as accuracy, precision, recall and F1-score.

### 4.2.1 Intrusion Detection

With massive growth of internet and interconnected devices, the need to secure the networks have currently become a very concerning problem for users and service providers. Every day, many attacks and new threats are created by intruders that may cause crash of the networks and loss of data. In order to tackle this major issue, Network Intrusion Detection Systems (NIDSs) are developed to properly find the attacks by monitoring the network's traffic for any abnormal actions [64]. With the continuous changing patterns in network behavior, it is inevitable to have a dynamic and automated approach to detect and prevent intrusions. Hence, machine learning and data mining techniques can be employed to create robust and effective NIDS.

In this experiment, we apply our model on the *NSL-KDD* data [39], a refined version of *KDD-CUP'99* intrusion detection dataset that we previously described and used in section 2.3.2 of chapter 2. The data has two categories of 13,449 normal and 11,743 anomaly instances with 41 quantitative and qualitative features. The BMBMM, IGMM and GMM are considered for comparison in order to better evaluate the performance of our proposed model.

The confusion matrices of these four models for normal and anomaly data are shown in Table 9. Table 10 demonstrates the overall performance of our proposed model as compared to the other models. According to the results presented in Table 10, we can notice that IMBMM and BMBMM approaches perform comparably while both have higher accuracy than IGMM and GMM. Moreover, our model outperforms the others in terms of F1-score, showing the efficiency and capability of the proposed model in detecting intrusions.

### 4.2.2 Hepatitis Prediction

Data mining and pattern recognition techniques have achieved considerable attention in medicine and biomedical researches and have been successfully applied across various problem domains, such as hepatitis diagnosis. Viral hepatitis as a life-threatening

Table 9: Confusion matrices for intrusion detection dataset

| | Normal (P) | Anomaly (P) |
|---|---|---|
| **IMBMM** | | |
| Normal (AC) | 10299 | 3150 |
| Anomaly (AC) | 1492 | 10251 |
| **BMBMM** | | |
| Normal (AC) | 10617 | 2832 |
| Anomaly (AC) | 2103 | 9640 |
| **IGMM** | | |
| Normal (AC) | 10921 | 2528 |
| Anomaly (AC) | 3246 | 8497 |
| **GMM** | | |
| Normal (AC) | 11294 | 2155 |
| Anomaly (AC) | 3652 | 8091 |

Table 10: Model performance results for intrusion detection

| | IMBMM(%) | BMBMM(%) | IGMM(%) | GMM(%) |
|---|---|---|---|---|
| Accuracy | 81.57 | 80.41 | 77.08 | 76.94 |
| Precision | 76.49 | 77.29 | 77.07 | 78.96 |
| Recall | 87.29 | 82.09 | 72.36 | 68.9 |
| F1-Score | 81.54 | 79.62 | 74.64 | 73.59 |

disease is among the most important global health issues in the world. The hepatitis is an inflammation of the liver which is commonly caused by some main viruses, referred to as types A, B, C, D and E [65]. However, its root could be other factors such as infections, autoimmune diseases or toxic substances. The blood test is a main method for identifying this disease. Based on WHO hepatitis facts, 257 million people infected by hepatitis B were globally reported in 2015 [66]. This is a motivation to develop automatic systems that can make an accurate diagnosis and early detection by employing machine learning and statistical tools. That can be the main motivation to design automatic diagnosis systems using machine learning and statistical based techniques in order to make accurate and early detection. These techniques can be employed as a decision support system to assist experts (e.g., doctors and specialists) in analyzing unlabeled patient's data and provide useful hepatitis detection.

In our experiment, we used real-world hepatitis dataset obtained from the UCI Repository [67] to evaluate our proposed model. It contains 155 samples with 19 features which are obtained from different results of medical tests of patients to predict Hepatitis patient survivability. The dataset consists of imbalance data class, with 26% of the patient die and 74% of the patient alive. The confusion matrices of different models are given in Table 11. Table 12 also reveals the performance comparison among different models.

Table 11: Confusion matrices for hepatitis prediction

| | Death (P) | Alive (P) |
|---|---|---|
| **IMBMM** | | |
| Death (AC) | 12 | 14 |
| Alive (AC) | 5 | 110 |
| **BMBMM** | | |
| Death (AC) | 13 | 13 |
| Alive (AC) | 17 | 98 |
| **IGMM** | | |
| Death (AC) | 16 | 10 |
| Alive (AC) | 24 | 91 |
| **GMM** | | |
| Death (AC) | 19 | 7 |
| Alive (AC) | 28 | 88 |

Table 12: Model performance results for hepatitis prediction

| | IMBMM(%) | BMBMM(%) | IGMM(%) | GMM(%) |
|---|---|---|---|---|
| Accuracy | 86.52 | 78.72 | 75.88 | 75.35 |
| Precision | 70.58 | 43.33 | 40.0 | 40.42 |
| Recall | 46.15 | 50.0 | 61.54 | 73.08 |
| F1-Score | 55.81 | 46.43 | 48.48 | 52.05 |

As observed in Table 12, the proposed IMBMM yields superior results compared to other approaches and it provides the highest accuracy rate (86.52%). Moreover, our model provide higher F1-socre which is and important metric in imbalanced data classification.

### 4.2.3 Software Defect Categorization

With rapid advancement in software development tools and activities in the last decades, software quality becomes an important and crucial aspect in user functionality of current software-based systems. The quality assessment in such complex software systems is quite costly and time-consuming. Thus, detection of software defects and failures is a critical process during the product quality assurance phase in order to enhance software reliability, avoid any additional costs, as well as, increasing software security. To address this problem, suitable metrics should be defined that can be representative of the software attributes. To that end, some relevant metrics such as the code size, McCabes cyclomatic complexity and the Halsteads complexity have been considered for evaluating software quality [68]. The Halsteads and McCabes complexity measures are based on the characteristics of the software modules as explained in [69]. The McCabes metric includes essential, cyclomatic and design complexity and the number of lines of code. While the Halsteads metric consists of base and derived measures and line of code (LOC). Software defects can be identified with machine-learning and statistical-based tools. these tools are capable of detecting software issues in a real-time manner and help to reduce manual quality assurance and testing activities.

In this work, we examined our model on two datasets, namely JM1 and KC2 from the PROMISE data repository [70] obtained from NASA software projects which are currently used as benchmark datasets in this area of research. JM1 is written in C and is a real-time predictive ground system, while KC2 is a C++ dataset raised from system implementing storage management for receiving and processing ground data. The quality of software source code is described into McCabe and Halstead features, where these datasets are highly imbalanced with binary classes of defective and normal (non-defective). Some properties of JM1 and KC2 datasets are outlined in Table 13.

Table 13: Software modules defect dataset properties

| | JM1 | KC2 |
|---|---|---|
| Language | C | C++ |
| Modules | 10885 | 522 |
| Defects | 2106 | 105 |

Table 14 summarizes the results of performance metrics applying different models for software defect categorization.

Table 14: Results on software modules defect categorization using different models

| Dataset | Model | Accuracy(%) | Precision(%) | Recall(%) | F1-score(%) |
|---|---|---|---|---|---|
| JM1 | IMBMM | 82.18 | 27.09 | 32.07 | 29.39 |
| | BMBMM | 80.32 | 24.22 | 33.02 | 27.94 |
| | IGMM | 70.46 | 16.05 | 36.79 | 22.35 |
| | GMM | 74.01 | 17.44 | 33.5 | 22.94 |
| KC2 | IMBMM | 79.55 | 51.06 | 68.57 | 58.53 |
| | BMBMM | 74.74 | 44.32 | 78.09 | 56.55 |
| | IGMM | 76.35 | 46.01 | 71.43 | 55.97 |
| | GMM | 71.94 | 41.38 | 80.0 | 54.55 |

Statistics in Table 14 reveal the fact that comparing to the other models, IMBMM provides higher accuracy, where it can also find the proper number of clusters. Since these datasets are highly imbalanced, F1-measure is more representative, which combining both precision and recall into one single score. Therefore, our model can provide higher level of performance due to higher F1-score.

### 4.2.4 Image Categorization

Image categorization becomes an extremely important task according to the vast amount of images generated daily from various resources. Hence, different solutions have been designed to regroup similar images belonging to the same categories. Clustering is among the most suitable approaches for this task as it can alleviate the cost of image labeling. To evaluate the efficiency of our model, Caltech 101 dataset has been used in this experiment [71]. We chose our images from four classes: 90 images of Motorbike, 85 images of Sunflower, 80 images of Watch and 96 images of Leopard class. Sample images from each group are shown in Fig. 5.

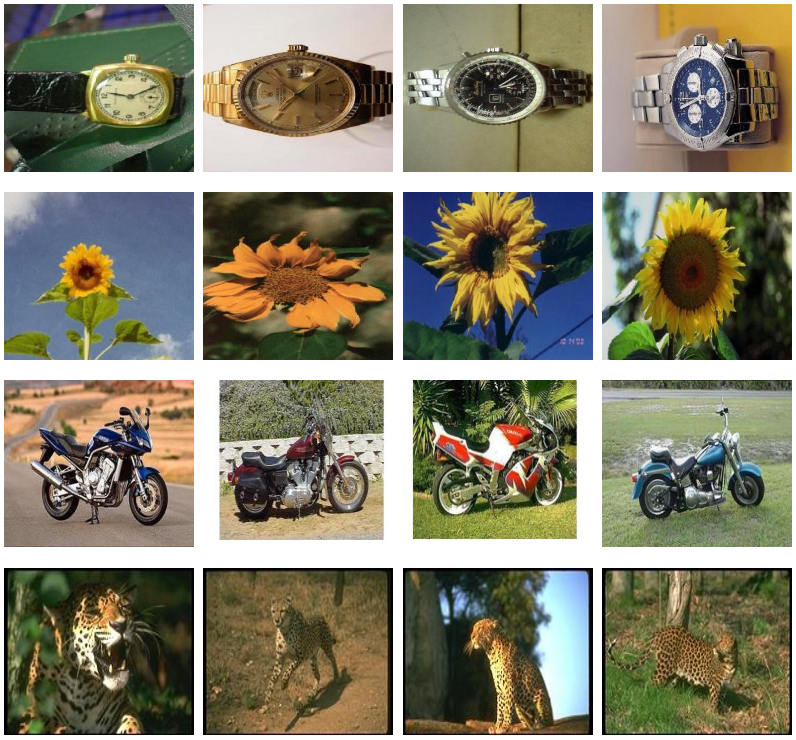


Figure 5: Sample images from four categories of Caltech 101 dataset

Before applying our model, we need to extract representative features from images. Hence, we used one of the most popular feature extractor algorithms, SIFT (scale invariant feature transform) for detection the features and extraction of descriptors of each image [35]. Then the obtained SIFT descriptors are clustered using K-means algorithm to form the bag of visual words (BoVW). In this representation, each image will be characterized with a histogram of visual words. The resulted vectors are then used as an input to our model for clustering. Figure 6 depicts the confusion matrix obtained by IMBMM. The Motorbike class has the highest accuracy compared to the other classes, however as can be observed many images of Leopard have been misclassified as Motorbike. The comparison results with other models are presented in Table 15. We used the macro average of precision, recall and F1-score. According to this table, the proposed IMBMM outperforms the other models with 78.63% accuracy, where it can also find the accurate number of components.

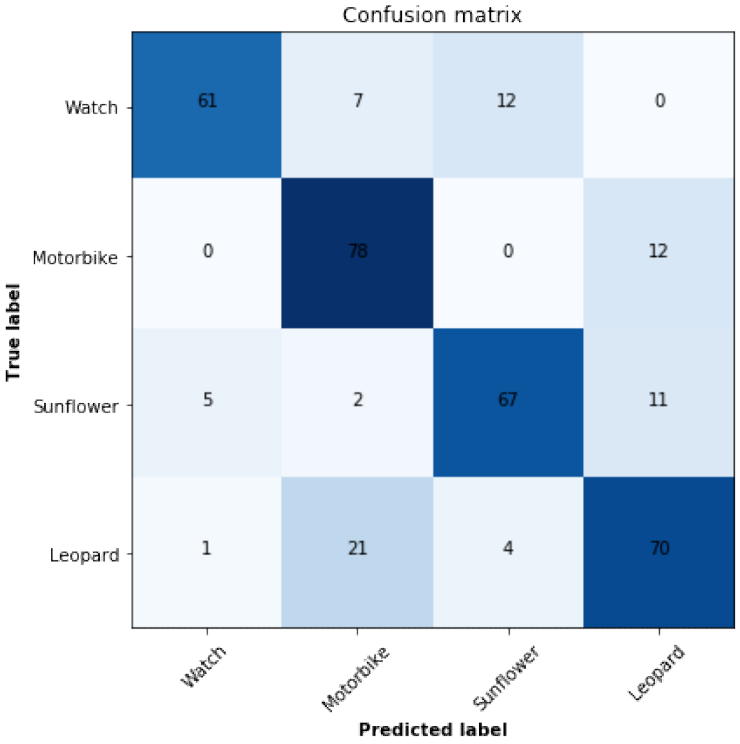



Figure 6: IMBMM Confusion matrix, Caltech dataset

Table 15: Model performance results for Caltech dataset

| | IMBMM(%) | BMBMM(%) | IGMM(%) | GMM(%) |
|---|---|---|---|---|
| Accuracy | 78.63 | 75.21 | 71.22 | 69.23 |
| Precision | 79.81 | 76.78 | 75.88 | 71.11 |
| Recall | 78.66 | 75.20 | 71.20 | 69.5 |
| F1-Score | 79.23 | 75.98 | 73.47 | 70.3 |

# Chapter 5

# Conclusion

Cluster analysis as an unsupervised learning method has been widely adopted for knowledge discovery and finding underlying structure of the data and finds its way into many applications from different domains, such as computer vision, information retrieval and pattern recognition.

Among various techniques, mixture models have been widely used for clustering problems in statistical modeling. In this thesis, we presented different Bayesian frameworks for multivariate Beta mixture models. The consideration of this distribution is due to its flexibility and capability in modeling non-Gaussian data.

First, we designed a Bayesian learning framework for the parameter estimation of our mixture model based on Markov Chain Monte Carlo technique through a hybrid sampling-based Metropolis-Hastings within Gibbs learning algorithm. We developed the proper posteriors in order to simulate parameters. Furthermore, we validated the effectiveness and performance of the proposed Bayesian inference over challenging real-world applications that concern cell image categorization and network intrusion detection. The results have shown that our model outperformed compare to other model.

Second, we introduced a fully Bayesian approach for finite mixtures of MB with an unknown number of components. To perform Bayesian analysis for our model, we adopted a Birth-Death MCMC algorithm with birth and death moves for simultanously updating the number of components and learning parameters by defining prior on number of components. The effectiveness of the proposed framework was evaluated using real-world medical applications, where the experimental results have

revealed that the performance of the proposed approach is convincing.

Third, a nonparametric Bayesian framework was developed to addressed the problem of learning infinite MB mixture models. Then, we estimated the posterior distributions through sampling-based MCMC technique. The efficiency of the proposed framework was demonstrated through different real-world applications including both balanced and imbalanced datasets, such as network intrusion detection, hepatitis diagnosis, software defect categorization and image categorization.

In conclusion, experimental results have shown that our Bayesian frameworks based on MB mixtures can outperform other standards methods which are mostly based on Gaussian assumption with effective estimation and selection of model parameters resulting in better performance. As potential future works, feature selection could be incorporated to our proposed frameworks. Moreover, other applications can be explored to further assess the performance of our Bayesian models.

# Bibliography


[1] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[2] Weifu Chen and Guocan Feng. Spectral clustering with discriminant cuts. *Knowledge-Based Systems*, 28:27–37, 2012.

[3] Geoffrey J McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

[4] W. Fan, N. Bouguila, and D. Ziou. Unsupervised hybrid feature extraction selection for high-dimensional non-gaussian data clustering with variational inference. *IEEE Transactions on Knowledge and Data Engineering*, 25(7):1670–1685, 2013.

[5] Ines Channoufi, Sami Bourouis, Nizar Bouguila, and Kamel Hamrouni. Image and video denoising by combining unsupervised bounded generalized gaussian mixture modeling and spatial information. *Multimedia Tools and Applications*, 77(19):25591–25606, 2018.

[6] Chi Liu, Heng-Chao Li, Kun Fu, Fan Zhang, Mihai Datcu, and William J Emery. Bayesian estimation of generalized gamma mixture model based on variational em algorithm. *Pattern Recognition*, 87:269–284, 2019.

[7] Sabri Boutemedjet, Djemel Ziou, and Nizar Bouguila. Model-based subspace clustering of non-gaussian data. *Neurocomputing*, 73(10-12):1730–1739, 2010.

[8] Sami Bourouis, Faisal R. Al-Osaimi, Nizar Bouguila, Hassen Sallay, Fahd M. Aldosari, and Mohamed Al Mashrgy. Bayesian inference by reversible jump MCMC for clustering based on finite generalized inverted dirichlet mixtures. *Soft Comput.*, 23(14):5799–5813, 2019.

[9] Nizar Bouguila. Infinite liouville mixture models with application to text and texture categorization. *Pattern Recognition Letters*, 33(2):103–110, 2012.

[10] Wentao Fan and Nizar Bouguila. Learning finite beta-liouville mixture models via variational bayes for proportional data clustering. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[11] Nizar Bouguila and Djemel Ziou. Dirichlet-based probability model applied to human skin detection [image skin detection]. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages V–521. IEEE, 2004.

[12] Nizar Bouguila and Djemel Ziou. Mml-based approach for finite dirichlet mixture estimation and selection. In Petra Perner and Atsushi Imiya, editors, *Machine Learning and Data Mining in Pattern Recognition, 4th International Conference, MLDM 2005, Leipzig, Germany, July 9-11, 2005, Proceedings*, volume 3587 of *Lecture Notes in Computer Science*, pages 42–51. Springer, 2005.

[13] Taoufik Bdiri and Nizar Bouguila. Positive vectors clustering using inverted dirichlet finite mixture models. *Expert Systems with Applications*, 39(2):1869–1882, 2012.

[14] 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):1716–1731, 2007.

[15] Nizar Bouguila, Djemel Ziou, and Ernest Monga. Practical bayesian estimation of a finite beta mixture through gibbs sampling and its applications. *Statistics and Computing*, 16(2):215–225, 2006.

[16] Narges Manouchehri, Hieu Nguyen, and Nizar Bouguila. Component splitting-based approach for multivariate beta mixture models learning. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1–5. IEEE, 2019.

[17] Mohamed Bouguessa. A mixture model-based combination approach for outlier detection. *International Journal on Artificial Intelligence Tools*, 23(04):1460021, 2014.

[18] Narges Manouchehri and Nizar Bouguila. A frequentist inference method based on finite bivariate and multivariate beta mixture models. In *Mixture Models and Applications*, pages 179–208. Springer, 2020.

[19] Nizar Bouguila, Djemel Ziou, and Riad I Hammoud. On bayesian analysis of a finite generalized dirichlet mixture via a metropolis-within-gibbs sampling. *Pattern Analysis and Applications*, 12(2):151–166, 2009.

[20] Min Yi, Ping Wei, Xian-Ci Xiao, and Heng-Ming Tai. Efficient em initialisation method for time delay estimation. *Electronics Letters*, 39(12):935–936, 2003.

[21] William M Bolstad and James M Curran. *Introduction to Bayesian statistics*. John Wiley & Sons, 2016.

[22] Mahsa Amirkhani, Narges Manouchehri, and Nizar Bouguila. Fully bayesian learning of multivariate beta mixture models. In *2020 IEEE 21th international conference on information reuse and integration for data science (IRI)*, pages 120–128. IEEE, 2020.

[23] Radford M Neal. Bayesian mixture modeling. In *Maximum Entropy and Bayesian Methods*, pages 197–211. Springer, 1992.

[24] Ingram Olkin and Ruixue Liu. A bivariate beta distribution. *Statistics & Probability Letters*, 62(4):407–412, 2003.

[25] Ingram Olkin and Thomas A Trikalinos. Constructions for a bivariate beta distribution. *Statistics & Probability Letters*, 96:54–60, 2015.

[26] 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. IEEE, 2017.

[27] Jean-Michel Marin, Kerrie Mengersen, and Christian P Robert. Bayesian modelling and inference on mixtures of distributions. *Handbook of statistics*, 25:459–507, 2005.

[28] Tarek Elguebaly and Nizar Bouguila. Simultaneous bayesian clustering and feature selection using rjmcmc-based learning of finite generalized dirichlet mixture models. *Signal Processing*, 93(6):1531–1546, 2013.

[29] PM Lee. Bayesian statistics: An introduction, 344 pp. *Edward Arnold, London*, 1997.

[30] Christian Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.

[31] Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[32] Taoufik Bdiri and Nizar Bouguila. Bayesian learning of inverted dirichlet mixtures for svm kernels generation. *Neural Computing and Applications*, 23(5):1443–1458, 2013.

[33] *Malaria statistics*. https://www.who.int/malaria/en/.

[34] Nih malaria dataset. available at https://ceb.nlm.nih.gov/repositories/malaria-datasets/.

[35] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[36] Anna L Buczak and Erhan Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, 18(2):1153–1176, 2015.

[37] Wajdi Alhakami, Abdullah ALharbi, Sami Bourouis, Roobaea Alroobaea, and Nizar Bouguila. Network anomaly intrusion detection using a nonparametric bayesian approach and feature selection. *IEEE Access*, 7:52181–52190, 2019.

[38] Wentao Fan, Nizar Bouguila, and Djemel Ziou. Unsupervised anomaly intrusion detection via localized bayesian feature selection. In *2011 IEEE 11th International Conference on Data Mining*, pages 1032–1037. IEEE, 2011.

[39] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. A detailed analysis of the kdd cup 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications*, pages 1–6. IEEE, 2009.

[40] Matthew Stephens. Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods. *Annals of statistics*, pages 40–74, 2000.

[41] Nizar Bouguila, Jian Han Wang, and A Ben Hamza. Software modules categorization through likelihood and bayesian analysis of finite dirichlet mixtures. *Journal of Applied Statistics*, 37(2):235–252, 2010.

[42] Sylvia Richardson and Peter J Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: series B (statistical methodology)*, 59(4):731–792, 1997.

[43] Nizar Bouguila and Tarek Elguebaly. A fully bayesian model based on reversible jump MCMC and finite beta mixtures for clustering. *Expert Syst. Appl.*, 39(5):5946–5959, 2012.

[44] Olivier Cappé, Christian P Robert, and Tobias Rydén. *Reversible jump MCMC converging to birth-and-death MCMC and more general continuous time samplers*. INSEE, 2001.

[45] Amir Mohammadi, MR Salehi-Rad, and EC Wit. Using mixture of gamma distributions for bayesian analysis in an m/g/1 queue with optional second service. *Computational Statistics*, 28(2):683–700, 2013.

[46] Tarek Elguebaly and Nizar Bouguila. Medical image classification using birth-and-death mcmc. In *2012 IEEE International Symposium on Circuits and Systems*, pages 2075–2078. IEEE, 2012.

[47] Tarek Elguebaly and Nizar Bouguila. A bayesian approach for the classification of mammographic masses. In *2013 Sixth International Conference on Developments in eSystems Engineering*, pages 99–104. IEEE, 2013.

[48] Olivier Cappé, Christian P Robert, and Tobias Rydén. Reversible jump, birth-and-death and more general continuous time markov chain monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(3):679–700, 2003.

[49] Davide Chicco and Giuseppe Jurman. Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. *BMC medical informatics and decision making*, 20(1):16, 2020.

[50] Heart failure clinical records data set. available at https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records, 2020.

[51] Ankita Tyagi, Ritika Mehra, and Aditya Saxena. Interactive thyroid disease prediction system using machine learning technique. In *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pages 689–693. IEEE, 2018.

[52] Thyroid disease. available at http://http://archive.ics.uci.edu/ml/datasets/thyroid+disease, 1992.

[53] 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.

[54] Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

[55] Carl Edward Rasmussen. The infinite gaussian mixture model. In *Advances in neural information processing systems*, pages 554–560, 2000.

[56] Jayanta K Ghosh and RV Ramamoorthi. *Bayesian nonparametrics*. Springer Science & Business Media, 2003.

[57] Ziyang Song, Samr Ali, and Nizar Bouguila. Bayesian learning of infinite asymmetric gaussian mixture models for background subtraction. In *International Conference on Image Analysis and Recognition*, pages 264–274. Springer, 2019.

[58] Tarek Elguebaly and Nizar Bouguila. A nonparametric bayesian approach for enhanced pedestrian detection and foreground segmentation. In *CVPR 2011 WORKSHOPS*, pages 21–26. IEEE, 2011.

[59] Nizar Bouguila and Djemel Ziou. A dirichlet process mixture of generalized dirichlet distributions for proportional data modeling. *IEEE Transactions on Neural Networks*, 21(1):107–122, 2009.

[60] Nizar Bouguila and Djemel Ziou. A dirichlet process mixture of dirichlet distributions for classification and prediction. In *2008 IEEE workshop on machine learning for signal processing*, pages 297–302. IEEE, 2008.

[61] Tarek Elguebaly and Nizar Bouguila. Infinite generalized gaussian mixture modeling and applications. In *International Conference Image Analysis and Recognition*, pages 201–210. Springer, 2011.

[62] David Blackwell, James B MacQueen, et al. Ferguson distributions via pólya urn schemes. *The annals of statistics*, 1(2):353–355, 1973.

[63] Nizar Bouguila and Djemel Ziou. A countably infinite mixture model for clustering and feature selection. *Knowledge and information systems*, 33(2):351–370, 2012.

[64] Wentao Fan, Nizar Bouguila, and Hassen Sallay. Anomaly intrusion detection using incremental learning of an infinite mixture model with feature selection. In *International Conference on Rough Sets and Knowledge Technology*, pages 364–373. Springer, 2013.

[65] Hepatitis. available at https://www.who.int/features/qa/76/en/e, 2013.

[66] Who hepatitis b fact sheet. available at https://www.who.int/news-room/fact-sheets/detail/hepatitis-b, 2020.

[67] Hepatitis. available at https://archive.ics.uci.edu/, 1998.

[68] Saiqa Aleem, Luiz Fernando Capretz, and Faheem Ahmed. Benchmarking machine learning technologies for software defect detection. *arXiv preprint arXiv:1506.07563*, 2015.

[69] Thomas J McCabe. A complexity measure. *IEEE Transactions on software Engineering*, (4):308–320, 1976.

[70] J. Sayyad Shirabad and T.J. Menzies. The PROMISE Repository of Software Engineering Databases. School of Information Technology and Engineering, University of Ottawa, Canada, 2005.

[71] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.