FINITE BIVARIATE AND MULTIVARIATE BETA MIXTURE MODELS LEARNING AND APPLICATIONS

Narges Manouchehri

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

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

March 2019
© Narges Manouchehri, 2019

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Narges Manouchehri**

Entitled: **Finite Bivariate and Multivariate Beta Mixture Models Learning and Applications**

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

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. Walter Lucia _____ Chair

Dr. Nizar Bouguila _____ Supervisor

Dr. Fereshteh Mafakheri _____ CIISE Examiner

Dr. Joonhee Lee _____ External Examiner

Approved _____
Dr. Chadi Assi Graduate Program Director

2019.02.28 _____
Dr. Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Finite Bivariate and Multivariate Beta Mixture Models Learning and Applications

Narges Manouchehri

Finite mixture models have been revealed to provide flexibility for data clustering. They have demonstrated high competence and potential to capture hidden structure in data. Modern technological progresses, growing volumes and varieties of generated data, revolutionized computers and other related factors are contributing to produce large scale data. This fact enhances the significance of finding reliable and adaptable models which can analyze bigger, more complex data to identify latent patterns, deliver faster and more accurate results and make decisions with minimal human interaction. Adopting the finest and most accurate distribution that appropriately represents the mixture components is critical. The most widely adopted generative model has been the Gaussian mixture. In numerous real-world applications, however, when the nature and structure of data are non-Gaussian, this modeling fails. One of the other crucial issues when using mixtures is determination of the model complexity or number of mixture components. Minimum message length (MML) is one of the main techniques in frequentist frameworks to tackle this challenging issue.

In this work, we have designed and implemented a finite mixture model, using the bivariate and multivariate Beta distributions for cluster analysis and demonstated its flexibility in describing the intrinsic characteristics of the observed data. In addition, we have applied our estimation and model selection algorithms to synthetic and real datasets. Most importantly, we considered interesting applications such as in image segmentation, software modules defect prediction, spam detection and occupancy estimation in smart buildings.

Acknowledgments

I would like to express my very profound gratitude to my supervisor Prof. Nizar Bouguila. It's hard to put into words exactly to express my thanks to him as a prestigious scientist with such a nice personality. It is one of my greatest honor in whole life to be his student. After several years of real world experience, I decided to continue my education and joined his team in January 2018, without having any background in machine learning. As an extraordinary teacher, he let me have the chance to start my studies and trusted me. As a result of his kind endless supports, patience, encouragement and wise guidance, I could realize one of my biggest dreams which is continuing my education. I will be forever grateful to him for giving me the opportunity to work under his guidance.

I would like to offer my special thanks to one of my best friends Ms. Leila Zare who supported and helped me in different steps of my long journey to Canada and starting my education.

During my studies, it was a great chance to meet Dr. Taoufik Bdiri and Walid Masoudimansour. Thanks for their wise advices. Special thanks to my knowledgeable and nice friend, Jaspreet who is helping me in discovering new field of science. Also, I am so thankful to my friends Muhammad, Omar, Soudeh, Kamal, Basim, Divya, Hieu and other lab mates.

It is noteworthy to mention my gratitude to Ms. Silvie Pasquarelli and Mireille Wahba who supports us with their kind helps and advices, always having beautiful smiles.

Last but not the least, I would like to thank my family as the most valuable gift of my life because of their unconditional love, respect and trust. They always encourage me to achieve my goals, realize my dreams and fly on my own which let me have confidence to discover new worlds. I am truly blessed to have you. Thank you for everything you've ever done for me and ever taught me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Over the past couple of decades, machine learning experienced tremendous growth and advancement. Accurate data analysis, extraction and retrieval of information have been largely studied in the various fields of technology [1]. Technological improvement led to the generation of huge amount of complex data of different types [2]. Various statistical approaches have been suggested in data mining, however data clustering received considerable attention and still is a challenging and open problem [3]. Finite mixture models have been proven to be one of the most strong and flexible tools in data clustering and have seen a real boost in popularity. Multimodal and mixed generated data consist of different components and categories and mixture models proved to be an enhanced statistical approach to discover the latent pattern of data [4, 5]. One of the crucial challenges of modeling and clustering is applying the most appropriate distribution. Most of the literature on finite mixtures concern Gaussian mixture model (GMM) [6]. However, GMM is not a proper tool to express the latent structure of non-Gaussian data. Recently, other distributions which are more flexible have been considered as a powerful alternative [7-33].

1.2 Objectives

The main objective of this thesis is to introduce a novel finite mixture modelling approach by focusing on a capable distribution. We developed a learning framework based on maximum likelihood estimation to infer the optimal parameters of our proposed mixture model and applied it to address following challenging issues:

1. Selecting a flexible mixture density which has demonstrated more efficiency in modeling asymmetric and non-Gaussian data

2. Parameter estimation as one of the crucial and critical challenges when deploying mixture models.

3. Assessment and validation of the feasibility and effectiveness of the proposed model by experimental results involving real datasets and real world applications.

4. Determination of the proper number of clusters by Minimum Message Length (MML).

5. Comparison of he performance of our framework with the widely used Gaussian Mixture Model (GMM).

In this work, we introduce unsupervised learning algorithms for finite mixture models based on bivariate and multivariate Beta distributions which could be applied in various real-world challenging problems. As explained above, our proposed learning framework will deploy deterministic and efficient techniques such as Maximum likelihood (ML), Expectation maximization (EM) and Newton Raphson methods. Furthermore, for model selection, minimum message length (MML) criterion is validated to find the optimal number of clusters inherent within real data sets. We evaluated our clustering approach on different problems.

1.3 Contributions

Our major contributions in this thesis are as follows:

1. Proposing a novel finite mixture model for non Gaussian data. Our proposed framework based on bivariate Beta distribution with three shape parameters and multivariate Beta distribution which to the extend of our knowledge haven't been used before in clustering. We developed all the equations related to its parameters estimation. We have proven that the these two mixtures can be good candidates to cluster data. This contribution has been published in the International Symposium

on Signal, Image, Video and Communications (ISIVC2018) [83].

2. Comparison of our models performance with finite Gaussian mixture model in terms of clustering.

3. Investigating the performance of our framework by testing it on synthetic and real data sets as well as as real-life applications such as spam detection, software modules defect prediction, image segmentation and estimation of occupancy in smart buildings.

1.4 Thesis Overview

In our thesis, we propose bivariate and multivariate Beta mixture models in following chapters:

1. In chapter 2, we present the bivariate Beta mixture model, develop a parameter estimation technique and demonstrate the results of our experiments based on synthetic data, real datasets, image segmentation and estimation of occupancy in smart buildings.

2. Chapter 3 is devoted to the multivariate Beta mixture model and its parameters estimation. MML is validated to find the proper number of clusters. The experimental results of the application of our approach on real data sets and two challenging applications, software defect detection and spam filtering, are compared with those of finite Gaussian mixture.

3. Finally in chapter 4, we conclude our work, highlight some challenges and suggest future works.

Chapter 2

Bivariate Beta Mixture Model

2.1 Bivariate Beta Distribution

Olkin and Liu [34, 35] have proposed a bivariate Beta distribution with two correlated random variables X and Y, both positive real values and less than one. These variables themselves are derived from three independent random variables U, V and W arised from standard Gamma distribution and parametrized by their shape parameters a, b and c, respectively as follows:

$$X = \frac{U}{(U+W)} \tag{1}$$

$$Y = \frac{V}{(V+W)} \tag{2}$$

The moments of X and Y are defined by following equations :

$$E(X) = \frac{a}{(a+c)} \tag{3}$$

$$Var(X) = \frac{ac}{(a+c)^2(a+c+1)} \tag{4}$$

$$E(Y) = \frac{b}{(b+c)} \tag{5}$$

$$Var(Y) = \frac{bc}{(b+c)^2(b+c+1)} \tag{6}$$

The joint density function of this bivariate distribution is expressed as follow:

$$f(X,Y) = \frac{X^{a-1}Y^{b-1}(1-X)^{b+c-1}(1-Y)^{a+c-1}}{B(a,b,c)(1-XY)^{(a+b+c)}} \tag{7}$$

$$B(a,b,c) = \frac{\Gamma(a)\Gamma(b)\Gamma(c)}{\Gamma(a+b+c)}$$

2.2 Mixture model

We assume $\mathcal{X} = \{\vec{X}_1, \vec{X}_2, ..., \vec{X}_N\}$ is a set of N 2-dimensional vectors and each vector $\vec{X}_n = (X_{n1}, X_{n2})$ is generated from a finite but unknown bivariate Beta mixture model $p(\vec{X}|\Theta)$. Considering $\mathcal{X}$ as a composition of M different clusters, we can approximate it by a finite mixture model [36] as described in Eq.8 where $\vec{\alpha}_j = (a_j, b_j, c_j)$. The weight of component j is denoted by p_j. The mixing proportions are positive and sum to one. $\Theta = (p_j, \vec{\alpha}_j)$ represents the set of weight and shape parameters of component j. In our model, Θ includes p_j, a_j, b_j and c_j.

$$p\Big(\vec{X}|\Theta\Big) = \sum_{j=1}^{M} p_j p(\vec{X}|\vec{\alpha}_j) \tag{8}$$

2.3 Model learning

To learn our model, we first apply k-means to initially cluster our data and with the help of means and variances of clusters, Eq.3 to Eq.6, the initial shape parameters can be approximated. This procedure is called method of moments. Afterward, we apply deterministic and efficient techniques such as maximum likelihood (ML), expectation maximization (EM) and Newton Raphson to update the parameters.

2.3.1 Method of moments

Method of moments (MM) is a statistical technique to estimate model's parameters. By the help of the mean and variance of components obtained from k-means phase and Eq.3 to Eq.6, the initial parameters are calculated as follow:

$$a = E(X)\Big(\frac{E(X)}{Var(X)}(1-E(X)) - 1\Big) \tag{9}$$

$$b = E(Y)\Big(\frac{E(Y)}{Var(Y)}(1 - E(Y)) - 1\Big) \tag{10}$$

$$c = \Big(1 - E(Y)\Big)\Big(\frac{E(Y)}{Var(Y)}(1 - E(Y)) - 1\Big) \tag{11}$$

2.3.2 Maximum likelihood and EM algorithm

To tackle the model estimation problem, the parameters which maximize the probability density function of data are determined using ML [37] and EM framework [38]. ML is an estimation procedure to find the mixture model parameters that maximize log-likelihood function [39] which is defined by:

$$L(\Theta, \mathcal{X}) = \log p(\mathcal{X}|\Theta) = \sum_{n=1}^{N} \log \Big(\sum_{j=1}^{M} p_j p(\vec{X}_n|\vec{\alpha}_j)\Big) \tag{12}$$

Each $\vec{X}_n$ is supposed to be arisen from one of the components. Hence, a set of membership vectors is introduced as $\vec{Z}_n = (Z_{n1}, \ldots, Z_{nM})$ where:

$$z_{nj} = \begin{cases} 1 & \text{if } \vec{X}_n \text{ belongs to a component j,} \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

$$\sum_{j=1}^{M} z_{nj} = 1 \tag{14}$$

The complete log-likelihood is given as following:

$$L(\Theta, Z, \mathcal{X}) = \sum_{j=1}^{M} \sum_{n=1}^{N} Z_{nj}\Big(\log p_j + \log p(\vec{X}_n|\vec{\alpha}_j)\Big) \tag{15}$$

In Expectation phase, we assign each vector $\vec{X}_n$ to one of the clusters by its posterior probability given by:

$$\hat{Z}_{nj} = p(j|\vec{X}_n, \vec{\alpha}_j) = \frac{p_j p(\vec{X}_n|\vec{\alpha}_j)}{\sum_{j=1}^{M} p_j p(\vec{X}_n|\vec{\alpha}_j)} \tag{16}$$

The complete log-likelihood is computed as:

$$L(\Theta, Z, \mathcal{X}) = \sum_{j=1}^{M}\sum_{n=1}^{N} \hat{Z}_{nj}(\log p_j + \log p(\vec{X}_n|\vec{\alpha}_j)) = \sum_{j=1}^{M}\sum_{n=1}^{N} \hat{Z}_{nj}(\log p_j +$$

$$(\mathrm{a}_j - 1)\log X_n + (b_j - 1)\log Y_n + (b_j + c_j - 1)\log(1 - X_n) +$$

$$(\mathrm{a}_j + c_j - 1)\log(1 - Y_n) - (a_j + b_j + c_j)\log(1 - X_n Y_n) +$$

$$\log\Gamma(a_j + b_j + c_j) - \log\Gamma(a_j) - \log\Gamma(b_j) - \log\Gamma(c_j)) \quad (17)$$

In maximization step, the gradient of the log-likelihood with respect to parameters is calculated. To solve this optimization problem, we need to find a solution for the following equation:

$$\frac{\partial \log L(\Theta, Z, \mathcal{X})}{\partial \Theta} = 0 \quad (18)$$

However, it doesn't have a closed-form solution and Newton-Raphson as an iterative approach assists to compute the updated parameters by:

$$\hat{\alpha_j}^{new} = \hat{\alpha_j}^{old} - {H_j}^{-1} G_j \quad (19)$$

where G_j is the first derivatives vector described in Eq.20.

$$G_j = (G_{1j}, G_{2j}, G_{3j})^T \quad (20)$$

$$G_{1j} = \frac{\partial L(\mathcal{X}, \Theta)}{\partial a_j} = \sum_{j=1}^{M}\sum_{n=1}^{N} \hat{Z}_{nj}(\log(X_n) + \log(1 - Y_n) - \log(1 - X_n Y_n) +$$
$$\Psi(a_j + b_j + c_j) - \Psi(a_j)) \quad (21)$$

$$G_{2j} = \frac{\partial L(\mathcal{X}, \Theta)}{\partial b_j} = \sum_{j=1}^{M}\sum_{n=1}^{N} \hat{Z}_{nj}(\log(Y_n) + \log(1 - X_n) - \log(1 - X_n Y_n) +$$

$$\Psi(a_j + b_j + c_j) - \Psi(b_j)) \quad (22)$$

$$G_{3j} = \frac{\partial L(\mathcal{X}, \Theta)}{\partial c_j} = \sum_{j=1}^{M} \sum_{n=1}^{N} \hat{Z}_{nj}(\log(1 - X_n) + \log(1 - Y_n) - \log(1 - X_n Y_n) +$$

$$\Psi(a_j + b_j + c_j) - \Psi(c_j))(23)$$

$\Psi(.)$ and $\Psi'(.)$ are digamma and trigamma functions, respectively defined by:

$$\Psi(X) = \frac{\Gamma'(X)}{\Gamma(X)}, \Psi'(X) = \frac{\Gamma''(X)}{\Gamma(X)} - \frac{\Gamma'(X)^2}{\Gamma(X)^2} \tag{24}$$

H_j is Hessian matrix expressed by following equations where $|\vec{\alpha_j}| = a_j + b_j + c_j$.

$$H = \left\{ \begin{array}{ccc} \frac{\partial G_{1j}}{\partial a} & \frac{\partial G_{1j}}{\partial b} & \frac{\partial G_{1j}}{\partial c} \\ \frac{\partial G_{2j}}{\partial a} & \frac{\partial G_{2j}}{\partial b} & \frac{\partial G_{2j}}{\partial c} \\ \frac{\partial G_{3j}}{\partial a} & \frac{\partial G_{3j}}{\partial b} & \frac{\partial G_{3j}}{\partial c} \end{array} \right\} =$$

$$= \sum_{i=1}^{N} \hat{Z}_{nj} \left\{ \begin{array}{ccc} \Psi'(|\vec{\alpha_j}|) - \Psi'(a_j) & \Psi'(|\vec{\alpha_j}|) & \Psi'(|\vec{\alpha_j}|) \\ \Psi'(|\vec{\alpha_j}|) & \Psi'(|\vec{\alpha_j}|) - \Psi'(b_j) & \Psi'(|\vec{\alpha_j}|) \\ \Psi'(|\vec{\alpha_j}|) & \Psi'(|\vec{\alpha_j}|) & \Psi'(|\vec{\alpha_j}|) - \Psi'(c_j) \end{array} \right\} (25)$$

The estimated values of mixing proportion has a closed-form solution expressed by:

$$p_j = \frac{\sum_{n=1}^{N} p(j|\vec{X}_n, \vec{\alpha}_j)}{N} \tag{26}$$

2.4 Estimation Algorithm

The initialization and estimation framework is described as follows:

1. INPUT: $\mathcal{X}$ and M.
2. Apply the k-means to obtain initial M clusters.
3. Apply the moments method for each component j to obtain $\vec{\alpha_j}$.
4. Expectation step: Compute $\hat{Z}_{nj}$ using Eq.16.
5. Maximization step: Update $\vec{\alpha_j}$ and p_j using Eq.19 and Eq.26, respectively.
6. If $p_j < \epsilon$, discard component j and go to 4.
7. If the convergence criterion passes terminate, else go to 4.

2.5 Experimental Results

In this section to validate the performance of our proposed algorithm, we first test it on four synthetic data sets. Then, this model is evaluated by four real data sets. Moreover, we consider two real-life applications namely occupancy estimation in smart buildings and image segmentation.

2.5.1 Synthetic Data

To investigate the validity of our proposed approach, our framework is applied on four synthetic datasets arised from bivariate Beta mixtures with different parameters. Fig.1 to Fig.4 display four examples of finite bivariate Beta mixtures including two, three, four and five components.

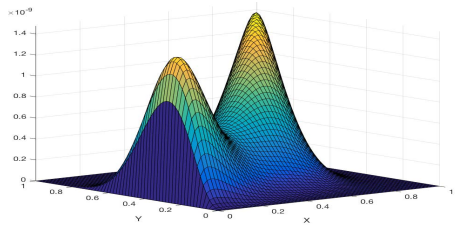

Figure 1: Two-component mixture

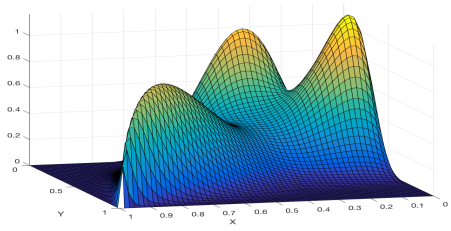

Figure 2: Three-component mixture

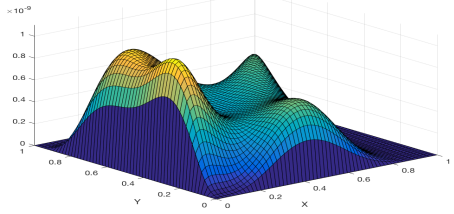

Figure 3: Four-component mixture

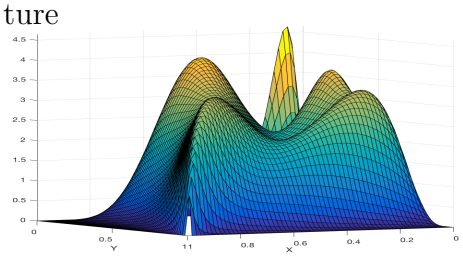

Figure 4: Five-component mixture

Tables 1 to 4 demonstrate the true and estimated parameters of four artificial datasets. According to the results shown in these tables, the model approximate parameters successfully in all four cases.

Table 1: Real and estimated parameters of the two-component mixture model.

Parameter	Real parameter	Estimated Parameter
a_1	1.555	1.5323
b_1	4.15	4.0936
c_1	7	6.9434
p_1	0.8	0.7884
a_2	5	4.3147
b_2	4	3.6988
c_2	2.5	2.3971
p_2	0.2	0.2116

Table 2: Real and estimated parameters of the three-component mixture.

Parameter	Real parameter	Estimated Parameter
a_1	1.41	1.1577
b_1	4.14	3.5623
c_1	7	6.3676
p_1	0.5	0.4642
a_2	5.154	5.0637
b_2	3.923	4.2792
c_2	2.511	2.5030
p_2	0.1572	0.1622
a_3	5.33	5.1638
b_3	1.42	1.3287
c_3	8.631	7.9498
p_3	0.3428	0.3736

Table 3: Real and estimated parameters of the four-component mixture model.

Parameter	Real parameter	Estimated Parameter
a_1	1.515	1.4866
b_1	4.177	3.4077
c_1	7.6491	6.3461
p_1	0.48	0.46
a_2	5.1149	5.7564
b_2	3.1595	3.5689
c_2	2.599	2.7757
p_2	0.15	0.14
a_3	5.2137	6.0211
b_3	1.5444	1.2297
c_3	5.1469	4.8582
p_3	0.16	0.18
a_4	1.112	1.7
b_4	9.192	8.2906
c_4	11.2	11.2
p_4	0.21	0.22

Table 4: Real and estimated parameters of the five-component mixture model.

Parameter	Real parameter	Estimated Parameter
a_1	3.8	3.7447
b_1	4.1	3.6345
c_1	11.58	11.6733
p_1	0.2	0.1982
a_2	8.3	8.84
b_2	1	1.6988
c_2	8.3	7.3868
p_2	0.15	0.1412
a_3	7.111	6.2232
b_3	7	7.1184
c_3	1.1	1.3868
p_3	0.25	0.2568
a_4	1.26	1.1543
b_4	11.1330	9.9503
c_4	8.8233	7.6636
p_4	0.25	0.2296
a_5	11.133	11.1
b_5	3.133	3.6988
c_5	2.483	2.3868
p_5	0.15	0.1742

2.5.2 Real Data

In this section, we estimate the accuracy of our algorithm by four real bivariate data sets. As the first step, we normalize our datasets using the following equation as one of the assumptions of our distribution is that the values of all observations are positive and less than one.

$$X^{'} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{27}$$

To assess the accuracy of the algorithm, the observations are assigned to different clusters based on Bayesian decision rule. Afterward, the accuracy is inferred by confusion matrix. Moreover, we compare bivariate Beta mixture model (BBMM) with Gaussian mixture model (GMM). We hereby introduce the datasets by describing their bivariate attributes and classes.

2.7.2.1 Haberman dataset

The first real dataset is a well-known one called Haberman based on a survival research at the University of Chicago's Billings Hospital between the years 1958 and 1970. It includes 306 instances of patients who had breast cancer and were monitored after having surgery. The dataset has three attributes age of patient at time of operation, patient's year of operation and number of positive axillary nodes detected. The database is labeled based on survival status. The patients who survived 5 years or longer were classified in first class and the ones died within 5 year were assigned to second class [44]. The patient's year of operation and number of positive axillary nodes detected are the two variables used for our assessment. Bivariate Beta mixture model performs with 92% accuracy but this value is 81% for the Gaussian mixture model.

2.7.2.2 Recombinant Bovine Growth Hormone on weight gain in rats dataset

The second data set, called Recombinant Bovine Growth Hormone on Weight Gain in Rats is the result of a research conducted by J.C. Juskevich and C.G. Guyer in 1990 [45]. It contains 60 instances of analyzing weight gains in rats over 85 days period in 6 treatment conditions of recombinant bovine growth hormone (rbGH). The variables are type of treatment and weight gain. The gender of rats is the label of dataset.

Male is 1 and female, 0. Bivariate Beta mixture has a performance of 82.19% while the accuracy is 61.49% for Gaussian mixture model.

2.7.2.3 Theophylline interactions with Heartburn Medication

In the third case of real data, T.J. Sullivan, J.H. Reese, et al studied randomized block design in patients with chronic obstructive pulmonary disease [46]. Each subject received Theophylline along with famotidine (Pepcid), cimetidine (Tagamet) and Placebo and Theophylline clearance (liters/hour) was measured. The variables are subject number and Theophylline clearance. The labels are 1=Placebo, 2=Pepcid and 3=Tagamet. The performance of our mixture model is 88% while Gaussian mixture model has an accuracy of 69%.

2.7.2.4 Occurrence of Nouns in Shakespeare's Plays

We also assess our model by a data set including 68 observations as a result of study about frequency of occurrence of nouns in two of Shakespeare's Plays, "Julius Caesar" and "As You Like It". Two attributes were number of occurrences and number of nouns with this many occurrences. The first class is assigned to "Julius Caesar" and the second one to "As You Like It" [47]. The results of our tests shows that our method outperforms Gaussian mixture model as their accuracies are is 91.1% and 82.3%, respectively.

2.5.3 Smart Building

Due to the increase of energy demands, rising global emissions of greenhouse gases and climate changes, resource consumption management has become one of the most complex and sophisticated technical topics. In response to such growing demands, smart buildings are new research trend and receiving increasing attention by academia and industry as building sector has an essential role in energy consumption. It has been acknowledged in published literatures that analysis of occupant's numbers and behaviors is one of the proposed portfolios that should largely be overlooked in building energy utilization patterns and managent of supplies. However, it is a critical and problematic process. Amayri et al. [48] conducted a research and carried out detailed energy audits to tackle this problem. In their method of research, performed

in Grenoble Institute of Technology, they controlled and measured specified factors by an ambiance sensing network. They found out that motions, CO2 concentration, power consumption, door and window positions, acoustic pressure from microphone were the most important features which could describe occupancy. The real number of occupants and their activities were recorded by two video cameras and considered as data classes. Afterwards, they calculated information gain values of each feature to find the most relevant ones for estimation of the occupancy. In our work, we consider the two most important features motion counter and acoustic pressure (microphone) as our variables. The data contains 717 instances and the value of class labels are 0, 1, 2, 3 and 4 which indicate the number of occupants. We evaluated our unsupervised model on this dataset and compared it with GMM to analyze its performance. The accuracy of our framework is based on confusion matrices illustrated in Tables 5 and 6. As it can be inferred, the accuracy of BBMM is 91.63% which outperformes the GMM with 75.17%.

Table 5: Confusion matrix of BBMM

497	35	8	4	2
5	83	0	0	0
0	0	47	0	0
0	0	5	23	0
0	0	0	0	7

Table 6: Confusion matrix of GMM

490	9	0	1	0
10	33	16	11	2
0	0	5	0	0
0	36	16	4	0
3	40	23	11	7

2.5.4 Color image segmentation

Image segmentation is one of the core research topics and high-level tasks in the field of computer vision. The significance of this application is highlighted by the fact that it nourishes numerous applications progressively. We validated our proposed framework by the well-known publicly available Berkeley segmentation data set [49], [50]. This database is composed of a variety of natural color images generally used as a reliable way to compare image segmentation algorithms. It is noteworthy that the choice of color space is an important problem when dealing with color images and it is highly desirable that the chosen space be robust against varying illumination, concise, discriminatory and noise. We applied $l_1l_2l_3$ color space which is a photometric color invariant for matte and shiny spaces [52] described as below:

$$l_1(R,G,B) = \frac{(R-G)^2}{(R-G)^2 + (R-B)^2 + (G-B)^2} \tag{28}$$

$$l_2(R,G,B) = \frac{(R-B)^2}{(R-G)^2 + (R-B)^2 + (G-B)^2} \tag{29}$$

$$l_3(R,G,B) = \frac{(G-B)^2}{(R-G)^2 + (R-B)^2 + (G-B)^2} \tag{30}$$

Moreover, considering a segmentation approach proposed by Yang and Krishnan [52], we assumed that each pixel $\vec{X}_n \in \mathcal{X}$ has an immediate neighbor $\vec{\hat{X}}_n \in \mathcal{X}$, so-called peer of former one, both arisen from the same cluster. Moreover, the boundary pixels are ignored as they have a minor share in the whole image. Since for each pixel (r, c) there are 4 main neighbors that are likely to be in the same region, we can use one of them as the corresponding peer. According to [52], (r + 1, c) is ideal to be neighbour pixel [53]. Figure 5 shows a comparison between results obtained by our approach and GMM.

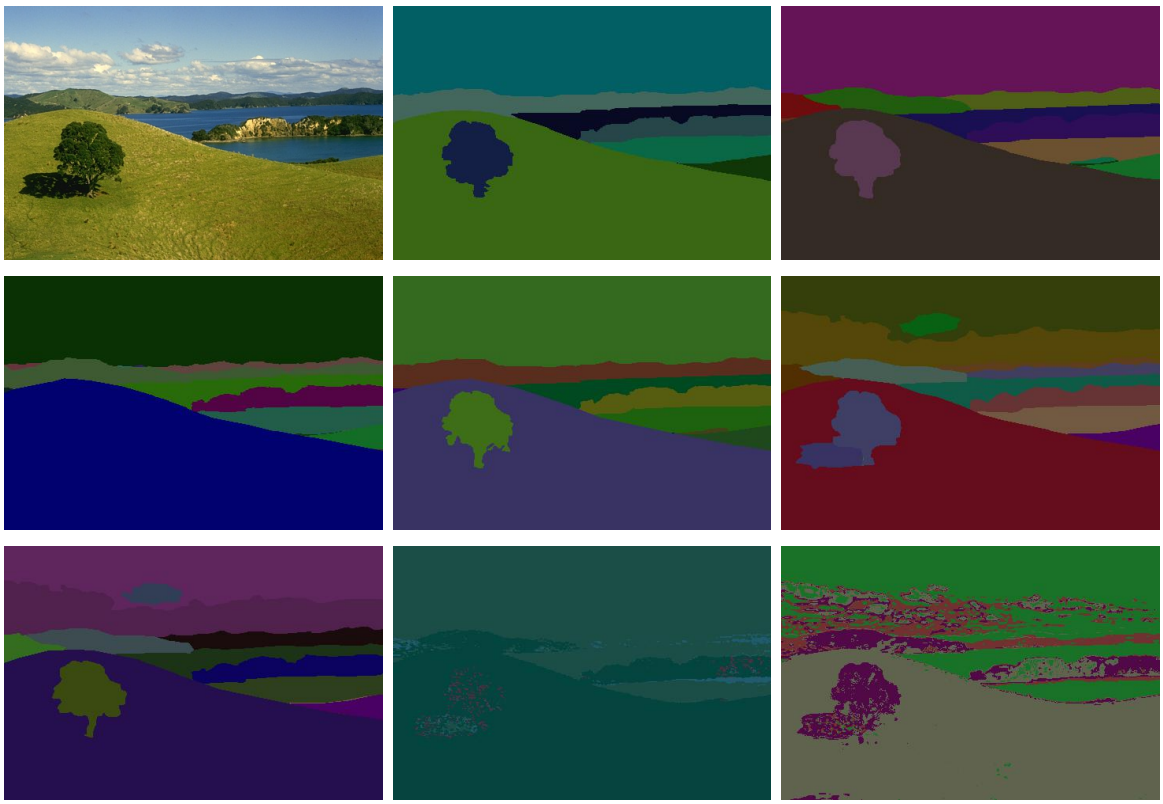

Figure 5: Original image (29030), 6 labeled images, GMM and BBMM outputs

As we need some quantitative measures to compare the segmentation results of BBMM with GMM, we applied six image segmentation evaluation metrics, Adjusted Rand Index (ARI), Adjusted Mutual Information Score (AMIS), Homogeneity Score (HS), Completeness Score (CS), Calinski-Harabaz Index (CHI), Jaccard similarity score (JSS) which their results are presented in Table 7. As it is shown, our model outperforms the GMM according to all metrics.

Table 7: The results of estimating model performance in image segmentation based on six metrics

	Metrics (K=4)					
Alg.	**ARI**	**NMIS**	**MIS**	**HS**	**VM**	**JSS**
BBMM	0.69	0.62	0.73	0.5	0.61	0.61
	0.69	0.61	0.74	0.48	0.6	0.6
	0.72	0.64	0.72	0.53	0.63	0.63
	0.69	0.63	0.73	0.5	0.61	0.61
	0.57	0.58	0.77	0.42	0.55	0.55
	0.56	0.57	0.74	0.42	0.54	0.54
Mean	**0.65**	**0.6**	**0.74**	**0.47**	**0.59**	**0.59**
GMM	0.58	0.46	0.59	0.41	0.45	0.45
	0.59	0.47	0.61	0.4	0.46	0.46
	0.56	0.44	0.55	0.4	0.44	0.44
	0.58	0.46	0.59	0.41	0.46	0.46
	0.61	0.49	0.72	0.39	0.48	0.48
	0.59	0.48	0.69	0.39	0.47	0.47
Mean	**0.59**	**0.47**	**0.62**	**0.4**	**0.46**	**0.46**

Chapter 3

Multivariate Beta Mixture Model

3.1 Multivariate Beta Distribution

This chapter is devoted to our proposed mixture model based on multivariate Beta distribution. In previous chapter, we introduced the bivariate distribution with three shape parameters and here will describe the multivariate case in detail. The multivariate Beta distribution is constructed by generalization of the bivariate distribution to k variate distribution. Let $U_1, ..., U_k$ and W be independent random variables each having a Gamma distribution and variable X is defined by Eq.31 where $i = 1, ...k$.

$$X_i = \frac{U_i}{(U_i + W)} \tag{31}$$

The joint density function of $X_1, ..., X_k$ after integration over W is expressed by:

$$f(x_1, ..., x_k) = c\frac{\prod_{i=1}^{k} x_i^{a_i - 1}}{\prod_{i=1}^{k}(1 - x_i)^{(a_i+1)}}\Big[1 + \sum_{i=1}^{k}\frac{x_i}{(1 - x_i)}\Big]^{-a} \tag{32}$$

where $0 \leq x_i \leq 1$ and:

$$c = B^{-1}(a_1, ..., a_k) = \frac{\Gamma(a_1 + ... + a_k)}{\Gamma(a_1).....\Gamma(a_k)} = \frac{\Gamma(a)}{\prod_{i=1}^{k}\Gamma(a_i)} \tag{33}$$

a_i is the shape parameter of each variable X_i and:

$$a = \sum_{i=1}^{k} a_i \tag{34}$$

3.2 Mixture model

Let us consider $\mathcal{X} = \{\vec{X}_1, \vec{X}_2, ..., \vec{X}_N\}$ be a set of N k-dimensional vectors such that each vector $\vec{X}_n = (X_{n1}, ..., X_{nk})$ is generated from a finite but unknown multivariate Beta mixture model $p\Big(\vec{X}|\Theta\Big)$. We assume that $\mathcal{X}$ is composed of M different finite clusters and can be approximated by a finite mixture model as described before by:

$$p\Big(\vec{X}|\Theta\Big) = \sum_{j=1}^{M} p_j p(\vec{X}|\vec{\alpha}_j) \tag{35}$$

where $\vec{\alpha}_j = (a_1, ..., a_k)$. $\Theta = (p_j, \vec{\alpha}_j)$ represents the set of weights and shape parameters of component j and the complete model parameters are denoted by $\{p_1, ..., p_M, \vec{\alpha}_1, ..., \vec{\alpha}_M\}$.

3.3 Model learning

3.3.1 Method of moments

The first two moments, sample mean and variance are defined by:

$$E(X) = \bar{x} = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{36}$$

$$Var(X) = \bar{v} = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{x})^2 \tag{37}$$

The shape and scale parameters of multivariate Beta distribution can be estimated using the method of moments by Equation 13 and Equation 14 as follow:

$$\hat{\alpha} = E(X)\Big(\frac{E(X)}{Var(X)}(1 - E(X)) - 1\Big) \tag{38}$$

$$\hat{\beta} = \Big(1 - E(X)\Big)\Big(\frac{E(X)}{Var(X)}(1 - E(X)) - 1\Big) \tag{39}$$

By the help of the mean and variance of components obtained from k-means phase, the initial parameters are approximated.

3.3.2 Maximum Likelihood and EM algorithm

As described in previous chapter, to tackle the problem of finding the parameters of our model, we apply ML approach and EM framework on the compelete likelihood

defined by Eq.40 and the value of a_j is computed by Eq.38 for each component of mixture model. The development of Eq.40 is provided in Appendix 1.

$$\mathcal{L}(\Theta, Z, \mathcal{X}) = \sum_{j=1}^{M}\sum_{n=1}^{N} z_{nj}\Big(\log p_j + \log p(\vec{X}_n|\vec{\alpha}_j)\Big) \tag{40}$$

The first derivatives of Eq. 40 with respect to a_{ji} where $i = 1, ..., k$ are given by:

$$\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial a_{ji}} = \sum_{j=1}^{M}\sum_{n=1}^{N} \hat{Z}_{nj}\Bigg(\log(X_{ni}) - \log(1 - X_{ni}) + \Psi(a_j) - \Psi(a_{ji})$$

$$\text{-log}\left[1 + \textstyle\sum_{i=1}^{k} \frac{X_{ni}}{(1-X_{ni})}\right]\Bigg) \tag{41}$$

As this equation doesn't have a closed form solution, we use Newton-Raphson method and G_j as the first and the Hessian matrix as the second and mixed derivatives of $L(\Theta, Z, \mathcal{X})$ as follow:

$$G_j = (G_{1j}, ..., G_{kj})^T \tag{42}$$

$$H = \begin{Bmatrix} \frac{\partial G_{j1}}{\partial a_{j1}} & \cdots & \frac{\partial G_{1j}}{\partial a_{jk}} \\ \vdots & \ddots & \vdots \\ \frac{\partial G_{jk}}{\partial a_{j1}} & \cdots & \frac{\partial G_{jk}}{\partial a_{jk}} \end{Bmatrix} =$$

$$= \sum_{i=1}^{N} \hat{Z}_{nj} \begin{Bmatrix} \Psi'(|\vec{a}_j|) - \Psi'(a_{j1}) & \cdots & \Psi'(|\vec{a}_j|) \\ \vdots & \ddots & \vdots \\ \Psi'(|\vec{a}_j|) & \cdots & \Psi'(|\vec{a}_j|) - \Psi'(a_{jk}) \end{Bmatrix} \tag{43}$$

where

$$|\vec{a}_j| = a_1 + ... + a_k \tag{44}$$

The estimated values of mixing proportions are expressed by Equation 30 as it has a closed-form solution:

$$p_j = \frac{\sum_{n=1}^{N} p(j|\vec{X}_n, \vec{\alpha}_j)}{N} \tag{45}$$

3.4 Estimation of model complexity with MML

The MML approach is based on evaluating statistical models according to their ability to compress a message containing the data. This technique has been proved to outperform many other model selection methods. The optimal number of clusters of the mixture is that which minimizes the amount of information needed to transmit data $\mathcal{X}$ efficiently from a sender to a receiver.

The formula for the message length for a mixture of distributions is given by Eq.46 where $h(\Theta)$ is the prior probability, $p(\mathcal{X}|\Theta)$ is the likelihood, $F(\Theta)$ is the expected Fisher information matrix, and $|F(\Theta)|$ is its determinant. N_p is the number of free parameters to be estimated and is equal to $(M(d+1))-1$. κN_p is the optimal quantization lattice constant for $\mathbb{R}^{N_p}$ [18] and we have $\kappa_1 \simeq \frac{1}{12} \simeq 0.083$ for $N_p = 1$ [54-62].

$$\text{MessLen} \simeq -\log(h(\Theta)) - \log\left(p(\mathcal{X}|\Theta)\right) + \tfrac{1}{2}\log(|F(\Theta)|) + \tfrac{N_p}{2}(1+\log(\kappa N_p)) \quad (46)$$

The Fisher information matrix is the expected value of the Hessian minus the logarithm of the likelihood. We use the complete data Fisher information matrix as proposed in [58]. The determinant of the complete data Fisher information matrix is [57]:

$$|F(\Theta)| \simeq |F(\vec{P})| \prod_{j=1}^{M} |F(\vec{\alpha_j})| \quad (47)$$

where $|F(\vec{P})|$ is the Fisher information with regards to the mixing parameters vector, and $|F(\vec{\alpha}_j)|$ is the Fisher information with regards to the vector $\vec{\alpha}_j$ of a single bivariate Beta distribution. For $|F(\vec{P})|$, mixing parameters satisfy the requirement $\sum_{j=1}^{M} p_j = 1$. Consequently, it is possible to consider the generalized Bernoulli process with a series of trials, each of which has M possible outcomes for M clusters. The determinant of the Fisher information matrix is given by Eq.48 where N is the number of data elements [63].

$$|F(\vec{P})| = \frac{N}{\prod_{j=1}^{M} p_j} \quad (48)$$

The Fisher information for our mixture is given as following:

$$\log|F(\Theta)| = \log(N) - \sum_{j=1}^{M} \log p_j + \sum_{j=1}^{M} \log|F(\vec{\alpha}_j)| \quad (49)$$

To calculate MML, we need to find $h(\Theta)$ which can be represented as follow [62]:

$$h(\Theta) = h(\vec{p})h(\alpha) \tag{50}$$

Considering the nature of the mixing parameters, it can be expressed by a symmetric Dirichlet distribution with parameter as defined in Eq.51 where $\vec{\eta} = (\eta_1, ..., \eta_M)$ is the parameter vector of the Dirichlet distribution:

$$h(\vec{p}) = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} {p_j}^{\eta_j - 1} \tag{51}$$

The choice of $\eta_1 = 1, \ldots, \eta_M = 1$ gives a uniform prior over the space $p_1 + ... + p_M = 1$ [62]. Therefor, the prior is given by:

$$h(\vec{p}) = (M-1)! \tag{52}$$

For $\vec{\alpha}$, we assume that components of α_j are independent:

$$h(\vec{\alpha}) = \prod_{j=1}^{M} h(\alpha_j) = \prod_{j=1}^{M} \prod_{d=1}^{D} h(\alpha_{jd}) \tag{53}$$

We choose to use following simple uniform prior which we experimentally found good results with it [64, 65].

$$h(\alpha_{jd}) = e^{-6} \frac{\alpha_{jd}}{\|\alpha_j\|} \tag{54}$$

The log of prior is given by:

$$\log(h(\Theta)) = \sum_{j=1}^{M-1} log(j) - 6MD - D \sum_{j=1}^{M} log(\|\alpha_j\|) + \sum_{j=1}^{M} \sum_{d=1}^{D} log(\alpha_{jd}) \tag{55}$$

3.5 Estimation Algorithm

The initialization and estimation framework is described as follows:

1. INPUT: $\mathcal{X}$ and M.
2. Apply the k-means to obtain initial M clusters.
3. Apply the moments method for each component j to obtain $\vec{\alpha_j}$.
4. Expectation step: Compute $\hat{Z}_{nj}$ using Eq.16.
5. Maximization step: Update $\vec{\alpha_j}$ and p_j Using Eq.19 and Eq.45, respectively.
6. If $p_j < \epsilon$, discard component j and go to 4.
7. If the convergence criterion passes terminate, else go to 4.
8. Calculate the associated criterion of MML and select the optimal number of components.

3.6 Experimental Results

In this section, we estimate the accuracy of our algorithm by testing it on two real world applications.

3.6.1 Software defect prediction

Software quality assurance and detection of a fault or a defect in a software program have become one of the topics that have received lots of attention in research and technology. Any failure in software may result in high costs for the system [66]. The evaluation of the quality of complex software systems is costly and complicated. Consequently, prediction of software failures and improving reliability is one of the attractive applications for scientists [67-71]. To tackle this problem, it is critical to define the appropriate metrics to express the attributes of the software modules. There are some metrics [72] for assessing software complexity such as the code size, McCabes cyclomatic and Halsteads complexity. 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) [73].

Prediction models [74-75] are applied to improve and optimize the quality which is translated to customer satisfaction as a significant achievement for the companies. Finite mixture models as flexible statistical solutions and clustering techniques are considered as powerful tools in this area [75-76].

Our experiment is performed on three datasets from the PROMISE data repository obtained from NASA software projects and its public MDP (Modular toolkit for Data Processing) which are currently used as benchmark datasets in this area of research [77]. The metrics or features of each dataset are five different lines of code measure, three McCabe metrics, four base Halstead measures, eight derived Halstead measures and a branch-count. The datasets are classified by a binary variable to indicate if the module is defective or not. CM1 as the first dataset is a NASA spacecraft instrument software written in "C". KC1 as the second one, is a "C++" dataset raised from system implementing storage management for receiving and processing ground data. The last case, PC1 is developed using "C" considering functions flight software for earth orbiting satellite. To highlight the basic properties of the datasets, Table 8 is created. As it is shown in Table 9 and Table 10, multivariate Beta mixture model (MBMM) has better performance in all three datasets in comparison with Gaussian mixture model (GMM). For CM1, the accuracy of our model is 98.79% while this value for GMM is 85.94% . KC1 has a more accurate result (94.12%) with MBMM than GMM (88.66%). The performance of the models for PC1 are similar: 94.13% and 91.79% of accuracy for MBMM and GMM, respectively. The precision and recall follow the same behavior as accuracy. The multivariate Beta mixture model is capable to reach 97.44% precision and 99.55% of recall for PC1 and KC1, respectively. While GMM has the best precision and recall in PC1 with 96.06% and 95.23%. Furthermore, MML results validate our approach for model selection as shown in Figure 6.

Table 8: Software modules defect properties

Dataset	Language	Instances	Defects
CM1	C	498	49
KC1	C++	2109	326
PC1	C	1109	77

Table 9: Software modules defect results inferred from the confusion matrix of multivariate Beta mixture model

Dataset	Accuracy	Precision	Recall
CM1	98.79	99.15	99.55
KC1	94.12	94.69	98.31
PC1	94.13	97.44	95.97

Table 10: Software modules defect results according to the confusion matrix of Gaussian mixture model

Dataset	Accuracy	Precision	Recall
CM1	85.94	92.21	90.88
KC1	88.66	93.99	92.69
PC1	91.79	96.06	95.23

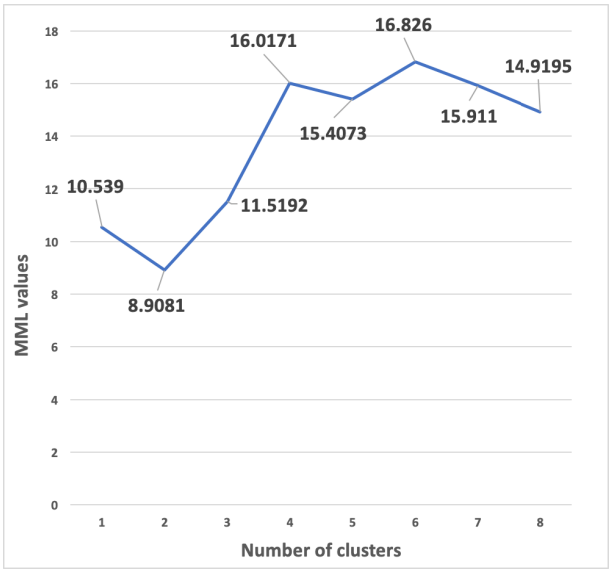

Figure 6: MML results for NASA dataset

3.6.2 Spam detection

Spam filtering as our second real application is one of the major research fields in information systems security. Spams or unsolicited bulk emails pose serious threats. As it was mentioned is some literature up to 75–80% of email messages are spam which resulted in heavy financial losses of 50 and 130 billion dollars in 2005 [78] and 2009 [79], respectively. Considering serious risks and costly consequences, classification and categorization of email [80] have received a lot of attention. Applying machine learning and pattern recognition techniques capability was enhanced compared to hand-made rules [80, 81].

Our experiment was carried out on a challenging spam data set obtained from UCI machine learning repository, created by Hewlett-Packard Labs [82]. This dataset contains 4601 instances and 58 attributes (57 continuous input attributes and 1 nominal class label target attribute). 39.4% of email (1813 instances) are spam and 60.6% (2788) are legitimate. The attributes are extracted from a commonly used technique called Bag of Words (BoW) as one of the main information representation methods in natural language processing. In this method, each email is presented by its words disregarding grammar. Most of the attributes in spambase dataset indicate whether a particular word or character was frequently occurring in the e-mail. 48 features include the percentage of words in the e-mail that match the word. 6 attributes

are extracted from the percentage of characters in the e-mail that match characters. The rest of the features are the average length of uninterrupted sequences of capital letters, the length of the longest uninterrupted sequence of capital letters and the total number of capital letters in the e-mail. The dataset class denotes whether the e-mail was considered spam or not. To evaluate our framework, first the dataset has been reduced to 3626 instances to have a balanced case. Then, it was normalized by Equation 31 as our assumption is that all observation values are between zero and one. Table 11 shows the results of our model performance in comparison with Gaussian mixture model considering their confusion matrices. As we can realize from table 4, multivariate Beta mixture model is more accurate (79.92%) and has higher value in terms of precision and recall, 80.6% and 82.74%, respectively. MML results are presented in Figure 7 as well.

Table 11: Spam filtering results to compare the performance of MBMM and GMM

Mixture model	Accuracy	Precision	Recall
MBMM	79.92	80.6	82.74
GMM	67.81	78.99	68.29

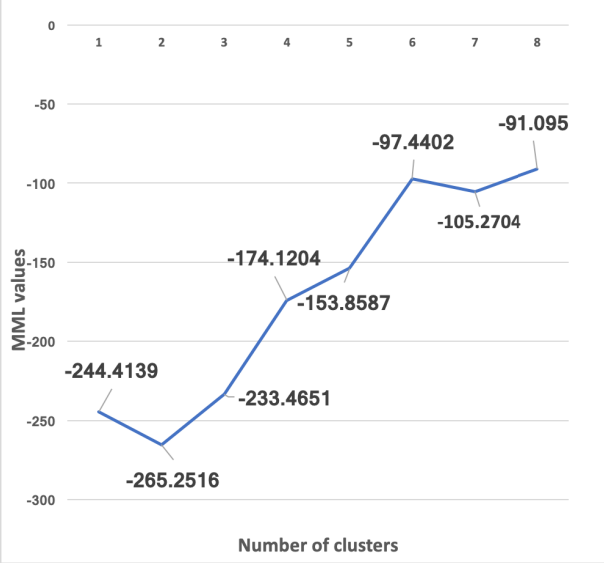

Figure 7: MML results for spam dataset

Chapter 4

Conclusion

In this thesis, we have presented two algorithms based on finite bivariate and multivariate Beta mixture models as novel methods of unsupervised learning, named clustering which is one of the critical challenges in machine learning. The choice of these distributions was motivated by their flexibility for data modelling as compared with the Gaussian distribution. In our work as a model based clustering, we explored deterministic approaches such as maximum likelihood using the expectation maximization algorithm framework to determine the parameters of our mixture models. A model selection technique namely the minimum message length was implemented to determine the number of clusters which describes the model complexity. Indeed, determining the number of components inherent in our dataset is critical in the task of parameter estimation in mixture models. In addition, we evaluated the modeling strengths of our mixture models on various datasets including synthetic and real data. The real and pre-labeled datasets helped to carry tests and validate our model. Moreover, we went further and considered very popular real-world applications. In chapter two we focused first on image segmentation a one of the main image processing techniques which has been receiving considerable attention because of its critical role in numerous applications. The second application in this chapter was estimation of the occupancy in smart buildings. We evaluated our unsupervised model on a real datasets and compared it with GMM to analyze its performance. In third chapter, our model was evaluated on two real world applications. The first one was software defect detection in the context of three NASA datasets. Spam filtering was our second topic of interest using the spam base dataset from the UCI repository.

From the outcomes, we can infer that the bivariate and multivariate Beta mixture models could be competitive modeling approaches. In other words, we can say that our model produces enhanced clustering results largely due to its flexibility [85,86].

Future works will explore more applications especially those dealing with time series data. Moreover, we will explore more efficient optimization techniques for estimating parameter vectors such as variational and Bayesian approaches.

Chapter 5

Appendix 1

Proof of Eq. 40:

$$\mathcal{L}(\Theta, Z, \mathcal{X}) = \sum_{j=1}^{M} \sum_{n=1}^{N} z_{nj}\Big(\log p_j + \log p(\vec{X}_n|\vec{\alpha}_j)\Big) =$$

$$\sum_{j=1}^{M} \sum_{n=1}^{N} \hat{Z}_{nj}\Bigg(\log p_j + \log \Big(\frac{\prod_{i=1}^{k} X_{ni}^{(a_{ji}-1)}}{\prod_{i=1}^{k}(1-X_{ni})^{(a_{ji}+1)}} \times \Big[1+\sum_{i=1}^{k} \frac{X_{ni}}{(1-X_{ni})}\Big]^{-a_j} \times \frac{\Gamma(\sum_{i=1}^{k} a_{ji})}{\prod_{i=1}^{k} \Gamma(a_{ji})}\Big)\Bigg) =$$

$$\sum_{j=1}^{M} \sum_{n=1}^{N} \hat{Z}_{nj}\Bigg(\log p_j + \log \Big(\prod_{i=1}^{k} X_{ni}^{(a_{ji}-1)}\Big) - \log \big(\prod_{i=1}^{k} (1-X_{ni})^{(a_{ji}+1)}\big) + \log \big(\Gamma(a_j)\big)$$

$$\text{-}\log \prod_{i=1}^{k} \Gamma(a_{ji}) + \log \Big[1 + \sum_{i=1}^{k} \frac{X_{ni}}{(1-X_{ni})}\Big]^{-a_j}\Bigg) =$$

$$\sum_{j=1}^{M} \sum_{n=1}^{N} \hat{Z}_{nj}\Bigg(\log p_j + \sum_{i=1}^{k} (a_{ji}-1)\big(\log(X_{ni})\big) - \sum_{i=1}^{k} (a_{ji}+1)\big(\log(1-X_{ni})\big)$$

$$+\log \big(\Gamma(a_j)\big) - \sum_{i=1}^{k} \log \big(\Gamma(a_{ji})\big) - a_j \log \Big(\Big[1 + \sum_{i=1}^{k} \frac{X_{ni}}{(1-X_{ni})}\Big]\Big)\Bigg)$$

Bibliography

[1] J. Han, M. Kamber, and J. Pei, Data mining :concepts and techniques, Elsevier, Morgan Kaufmann, Amsterdam; Boston, 2012.

[2] J. Diaz-Rozo, C. Bielza and P. x Larranaga, , Clustering of Data Streams With Dynamic Gaussian Mixture Models: An IoT Application in Industrial Processes, IEEE Internet of Things Journal, vol.5, pp. 3533, 2018.

[3] M. Giordan, R. Wehrens , A comparison of computational approaches for maximum likelihood estimation of the dirichlet parameters on high-dimensional data," SORT-Statistics and Operations Research Transactions, vol. 39, no. 1, pp. 109–126, 2015.

[4] G. J. McLachlan, "Mixture Models in Statistics," International Encyclopedia of the Social Behavioral Sciences, pp. 624-628, 2015.

[5] G.J. McLachlan and D. Peel, Finite Mixture Models. New York: Wiley, (2000).

[6] Z. Luo, W. He, M. Liwang, L. Huang, Y. Zhao, J.Geng , Real-time detection algorithm of abnormal behavior in crowds based on Gaussian mixture model, 12th International Conference on Computer Science and Education (ICCSE), pp. 183, 2017.

[7] T. Klauschies, R. M. Coutinho and U. Gaedke , A beta distribution-based moment closure enhances the reliability of trait-based aggregate models for natural populations and communities,Ecological Modelling, vol. 381, pp. 46-77, 2018.

[8] N. Bouguila, "Hybrid Generative/Discriminative Approaches for Proportional Data Modeling and Classification," IEEE Transactions on Knowledge and Data Engineering, (12), pp. 2184-2202, 2012.

[9] N. Bouguila, "Bayesian hybrid generative discriminative learning based on finite Liouville mixture models," Pattern Recognition, vol. 44, pp. 1183-1200, 2011.

[10] N. Bouguila, "Spatial Color Image Databases Summarization," IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 953-956, 2007.

[11] N. Bouguila and D. Ziou, "Unsupervised selection of a finite Dirichlet mixture model: an MML-based approach," IEEE Transactions on Knowledge and Data Engineering, (8), pp. 993, 2006.

[12] N. Bouguila and D. Ziou, "On fitting finite dirichlet mixture using ECM and MML," Third International Conference on Advances in Pattern Recognition, vol. 3686, pp. 172-182, 2005.

[13] N. Bouguila and D. Ziou, MML-Based Approach for Finite Dirichlet Mixture Estimation and Selection, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), pp.42-51, 2005.

[14] N. Bouguila and O. Amayri, "A discrete mixture-based kernel for SVMs: Application to spam and image categorization," Information Processing and Management, vol. 45, pp. 631-642, 2009.

[15] N. Bouguila and T. Elguebaly, "A fully Bayesian model based on reversible jump MCMC and finite Beta mixtures for clustering," Expert Systems with Applications, vol. 39, pp. 5946-5959, 2012.

[16] N. Bouguila and W. ElGuebaly, "Discrete data clustering using finite mixture models," Pattern Recognition, vol. 42, pp. 33-42, 2009.

[17] N. Bouguila, J. H. Wang and A. B. Hamza, "Software modules categorization through likelihood and bayesian analysis of finite dirichlet mixtures," Journal of Applied Statistics, vol. 37, (2), pp. 235-252, 2010.

[18] N. Bouguila and D. Ziou, "A countably infinite mixture model for clustering and feature selection," Knowledge and Information Systems, vol. 33, (2), pp. 351-370, 2012.

[19] N. Bouguila and D. Ziou, "A Dirichlet process mixture of dirichlet distributions for classification and prediction," EEE Workshop on Machine Learning for Signal Processing, pp. 297-302, 2008.

[20] N. Bouguila and D. Ziou, "Online clustering via finite mixtures of Dirichlet and minimum message length," Engineering Applications of Artificial Intelligence, vol. 19, pp. 371-379, 2006.

[21] N. Bouguila and D. Ziou, "Unsupervised selection of a finite Dirichlet mixture model: an MML-based approach," IEEE Transactions on Knowledge and Data Engineering, (8), pp. 993-1009, 2006.

[22] N. Bouguila and D. Ziou, "Using unsupervised learning of a finite Dirichlet mixture model to improve pattern recognition applications," Pattern Recognition Letters, vol. 26, (12), pp. 1916-1925, 2005.

[23] N. Bouguila and D. Ziou, "A probabilistic approach for shadows modeling and detection," IEEE International Conference on Image Processing, pp. 329-332, 2005.

[24] N. Bouguila and D. Ziou, "Dirichlet-based probability model applied to human skin detection [image skin detection]," IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 521-524, 2004.

[25] N. Bouguila and D. Ziou, "Improving content based image retrieval systems using finite multinomial dirichlet mixture," 14th IEEE Signal Processing Society Workshop Machine Learning for Signal Processing, pp. 23-32, 2004.

[26] N. Bouguila and D. Ziou, "A powerful finite mixture model based on the generalized Dirichlet distribution: unsupervised learning and applications," 17th International Conference on Pattern Recognition, pp. 280-283, 2004.

[27] N. Bouguila, D. Ziou and R. I. Hammoud, "On Bayesian analysis of a finite generalized Dirichlet mixture via a Metropolis-within-Gibbs sampling," Pattern Analysis and Applications, vol. 12, (2), pp. 151-166, 2009.

[28] T. Elguebaly and N. Bouguila, "Finite asymmetric generalized Gaussian mixture models learning for infrared object detection," Computer Vision and Image Understanding, vol. 117, pp. 1659-1671, 2013.

[29] W. Fan and N. Bouguila, "Variational learning of a Dirichlet process of generalized Dirichlet distributions for simultaneous clustering and feature selection," Pattern Recognition, vol. 46, pp. 2754-2769, 2013.

[30] W. Fan and N. Bouguila, "Online variational learning of finite Dirichlet mixture models," Evolving Systems, vol. 3, (3), pp. 153-165, 2012.

[31] W. Fan, N. Bouguila and D. Ziou, "Variational learning of finite Dirichlet mixture models using component splitting," Neurocomputing, vol. 129, pp. 3-16, 2014.

[32] N. Bouguila and D. Ziou, "High-Dimensional Unsupervised Selection and Estimation of a Finite Generalized Dirichlet Mixture Model Based on Minimum Message Length," , Pattern Analysis and Machine Intelligence, IEEE Transactions on, (10), pp. 1716, 2007.

[33] M. S. Allili, N. Bouguila and D. Ziou, "Finite Generalized Gaussian Mixture Modeling and Applications to Image and Video Foreground Segmentation ," Fourth Canadian Conference on Computer and Robot Vision, pp. 183-190, 2007.

[34] I. Olkin and R. Liu, "A bivariate beta distribution," Statistics and Probability Letters, 62, (4), pp. 407-412, 2003.

[35] I. Olkin and T. A. Trikalinos, "Constructions for a bivariate beta distribution," Statistics and Probability Letters, 96, pp. 54-60, 2015.

[36] C. M. Bishop, Pattern Recognition and Machine Learning. 2006.

[37] S. Ganesaligman, "Classification and Mixture Approaches to Clustering via Maximum Likelihood," Applied Statistics, 38, (3), pp. 455-466, 1989.

[38] G. J. McLachlan and T. Krishnan" The EM Algorithm and Extensions," New York: Wiley-Interscience,1997.

[39] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," Journal of the Royal Statistical Society, 39, pp.1–38, 1977.

[40] C. S. Wallace and D. L. Dowe, "Mml clustering of multistate, poisson, von mises circular and gaussian distributions," Statistics and Computing, vol. 10, no. 1, pp. 73–83, 2000.

[41] R. A. Baxter, "Minimum message length inference: Theory and applications," in Monash University, Australia. Citeseer, 1996.

[42] C. S. Wallace and P. R. Freeman, "Estimation and inference by compact coding," Journal of the Royal Statistical Society. Series B (Methodological), pp. 240–265, 1987.

[43] R. A. Baxter and J. J. Oliver, "Finding overlapping components with mml," Statistics and Computing, vol. 10, no. 1, pp. 5–16, 2000.

[44] Haberman's Survival Data, T. S. Lim, 1999. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Haberman.

[45] J.C. Juskevich and C.G. Guyer, "Bovine Growth Hormone: Human Food Safety Evaluation," Science, 249, 4971,pp. 875- 884, 1990.

[46] K. Bachmann, T.J. Sullivan, J.H. Reese, et al., "Controlled Study of the Putative Interaction Between Famotidine and Theophylline in Patients with Chronic Obstructive Pulmonary Disorder," Journal of Clinical Pharmacology, 35, pp.529-535, 1995.

[47] P.E. Bennett , "The Statistical Measurement of a Stylistic Trait in Julius Caesar and As You Like It," Shakespeare Quarterly,8, pp. 33-50,1957.

[48] M. Amayri, A. Arora, ,S. Ploix, ,S. Bandhyopadyay, Q. D. Ngo and V. R. Badarla ,"Estimating occupancy in heterogeneous sensor environment," Energy and Buildings, 129, pp. 46-58, 2016.

[49] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Computer Vision Proc. 8th IEEE Int'l Conf., volume 2, pages 416–423, 2001.

[50] "The Berkeley Segmentation Dataset and Benchmark" dataset [Online]. Available: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

[51] T. Gevers, A. W. M. Smeulders. Color-Based Object Recognition. Pattern Recognition, 32(3):453–464, 1999.

[52] X. Yang and Sh. M. Krishnan. Image Segmentation using Finite Mixtures and Spatial Information. Image and Vision Computing, 22(9):735–745, 2004.

[53] A. Sefidpour and N. Bouguila, "Spatial color image segmentation based on finite non-Gaussian mixture models," Expert Syst. Appl., 39, (10), pp. 8993-9001, 2012.

[54] C. S. Wallace and D. L. Dowe, "Mml clustering of multistate, poisson, von mises circular and gaussian distributions," Statistics and Computing, vol. 10, no. 1, pp. 73–83, 2000.

[55] R. A. Baxter, "Minimum message length inference: Theory and applications," in Monash University, Australia. Citeseer, 1996.

[56] C. S. Wallace and P. R. Freeman, "Estimation and inference by compact coding," Journal of the Royal Statistical Society. Series B (Methodological), pp. 240–265, 1987.

[57] R. A. Baxter and J. J. Oliver, "Finding overlapping components with mml," Statistics and Computing, vol. 10, no. 1, pp. 5–16, 2000.

[58] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 3, pp. 381–396, 2002.

[59] N. Bouguila and D. Ziou, "Unsupervised selection of a finite dirichlet mixture model: an mml-based approach," IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 8, pp. 993–1009, 2006.

[60] N. Bouguila and D. 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, (10), pp. 1716, 2007.

[61] M. A. T. Figueiredo and A. K. Jain, " Unsupervised Learning of Finite Mixture Models," IEEE Trans. Pattern Anal. Mach. Intell. 24(3): 381-396, 2002.

[62] N. Bouguila and D. 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, vol. 29, no. 10, 2007.

[63] Y. Agusta and D.L. Dowe, "Unsupervised Learning of Gamma Mixture Models Using Minimum Message Length," Proc. Third ASTED Conf. Artificial Intelligence and Applications, M.H. Hamza, ed., pp. 457-462, 2003.

[64] T. Bdiri and N. Bouguila, "Positive vectors clustering using inverted dirichlet finite mixture models," Expert Systems with Applications, vol. 39, no. 2, pp. 1869–1882, 2012.

[65] W. H. Jefferys and J. O. Berger, "Ockham's razor and bayesian analysis," American Scientist, vol. 80, no. 1, pp. 64–72, 1992.

[66] A. Bertolino , Software testing research: Achievements Achievements, challenges, dreams, Future of Software Engineering IEEE Computer Society, pp. 85–103, 2007.

[67] A. Boucher and M. Badri, Predicting fault-prone classes in objectoriented software: An adaptation of an unsupervised hybrid som algorithm, in Software Quality, Reliability and Security (QRS), IEEE International Conference, pp. 306–317, 2017.

[68] N. Kawashima and O. Mizuno, O, Predicting faultprone modules by word occurrence in identifiers, in Software Engineering Research, Management and Applications. Springer, pp. 87–98, 2015.

[69] E. Shihab, Practical software quality prediction, in Software Maintenance and Evolution (ICSME), 2014 IEEE International Conference , pp. 639–644, 2014.

[70] A. G. Koru and H. Liu, Building effective defectprediction models in practice, IEEE software, vol. 22, no. 6, pp. 23–29 2005.

[71] M. R. Lyu, et al. , Handbook of software reliability engineering, IEEE computer society press CA, vol. 222. inference, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.33, no.11, pp.2160–2173, 1996.

[72] S. Aleem ,L. F. Capretz and F. Ahmed , Benchmarking machine learning technologies for software defect detection, nternational Journal of Software Engineering Applications (IJSEA), Volume 6, No.3, pp. 11-23, 2015.

[73] T. J. McCabe, A complexity measure, IEEE Transactions on software Engineering, no. 4, pp. 308–320, 1976.

[74] L. C. Briand, V. Brasili and C. J. Hetmanski, Developing interpretable models with optimized set reduction for identifying high-risk software components, IEEE Transactions on Software Engineering, vol. 19, no. 11, pp. 1028–1044, 1993.

[75] K. El Emam , S. Benlarbi , N. Goel and S. N. Rai , Comparing casebased reasoning classifiers for predicting high risk software components, Journal of Systems and Software, vol. 55, no. 3, pp. 301–320, 2001.

[76] S. Oboh and N. Bouguila, Unsupervised learning of finite mixtures using scaled dirichlet distribution and its application to software modules categorization, in Proceedings of 2017 IEEE International Conference on Industrial Technology (ICIT), pp. 1085–1090, 2017.

[77] N. Bouguila, and O. Amayri, A discrete mixturebased kernel for SVMs: Application to spam and image categorization, Information Processing and Management, vol. 45, pp. 631-642, 2009.

[78] PROMISE Software Engineering Repository data set 2004, NASA , accessed 2 December 2004, http://promise.site.uottawa.ca/SERepository/datasetspage.html

[79] E. Blanzieri and A. Bryl, A survey of learningbased techniques of email spam filtering, Artificial Intelligence Review, vol. 29, pp. 63–92, 2008.

[80] Y. Zhu and Y. Tan, A local-concentrationbased feature extraction approach for spam filtering. IEEE Transactions on Information Forensics and Security, vol. 6, no. 2, pp. 486, 2011.

[81] L. Ozgur and T. Gungor, Optimization of dependency and pruning usage in text classification, Pattern analysis and applications, vol. 15, no. 1, pp. 45-58, 2012.

[82] W. Fan and N. Bouguila, Variational learning of a Dirichlet process of generalized Dirichlet distributions for simultaneous clustering and feature selection, Pattern Recognition, vol. 46, pp. 2754-2769, 2013.

[83] O. Amayri and N. Bouguila, A study of spam filtering using support vector machines, Artificial Intelligence Review, vol. 34, no.1, pp. 73-108, 2010.

[84] Spambase UCI Repository data set 1999, accessed 2 August 1999, https://archive.ics.uci.edu/ml/machine-learningdatabases/ spambase/spambase.data

[85] N. Manouchehri and N. Bouguila,"Learning of Finite Two-Dimensional Beta Mixture Models", 9th International Symposium on Signal, Image, Video Communications (ISIVC2018)

[86] N. Manouchehri and N. Bouguila,"A Probabilistic Approach based on a Finite Mixture Model of Multivariate Beta Distributions", 21st International Conference on Enterprise Information Systems (ICEIS2019)