Bounded Support Finite Mixtures for Multidimensional Data Modeling and Clustering

Muhammad Azam

A Thesis
in
The Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy (Electrical & Computer Engineering) at
Concordia University
Montréal, Québec, Canada

November 2019

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Muhammad Azam

Entitled: Bounded Support Finite Mixtures for Multidimensional Data Modeling and Clustering

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

Doctor of Philosophy (Electrical & Computer Engineering)

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

Signed by the final examining committee:

_____ Chair
Dr. Mehdi Hojjati

_____ External Examiner
Dr. Mohand Said Allili

_____ External to Program
Dr. Ali Dolatabadi

_____ Examiner
Dr. Wahab Hamou-Lhadj

_____ Examiner
Dr. Jia Yuan Yu

_____ Examiner
Dr. Abdessamad Ben Hamza

_____ Thesis Supervisor
Dr. Nizar Bouguila

Approved by _____
Dr. Rastko Selmic, Graduate Program Director

November 11, 2019 _____
Dr. Amir Asif, Dean
Gina Cody School of Engineering & Computer Science

ABSTRACT

Bounded Support Finite Mixtures for Multidimensional Data Modeling and Clustering

Muhammad Azam, Ph.D.
Concordia University, 2019

Data is ever increasing with today's many technological advances in terms of both quantity and dimensions. Such inflation has posed various challenges in statistical and data analysis methods and hence requires the development of new powerful models for transforming the data into useful information. Therefore, it was necessary to explore and develop new ideas and techniques to keep pace with challenging learning applications in data analysis, modeling and pattern recognition. Finite mixture models have received considerable attention due to their ability to effectively and efficiently model high dimensional data. In mixtures, choice of distribution is a critical issue and it has been observed that in many real life applications, data exist in a bounded support region, whereas distributions adopted to model the data lie in unbounded support regions. Therefore, it was proposed to define bounded support distributions in mixtures and introduce a modified procedure for parameters estimation by considering the bounded support of underlying distributions. The main goal of this thesis is to introduce bounded support mixtures, their parameters estimation, automatic determination of number of mixture components and application of mixtures in feature extraction techniques to overall improve the learning pipeline. Five different unbounded support distributions are selected for applying the idea of bounded support mixtures and modified parameters estimation using maximum likelihood via Expectation-Maximization (EM). Probability density functions selected for this thesis include Gaussian, Laplace, generalized Gaussian, asymmetric Gaussian and asymmetric generalized Gaussian distributions, which are chosen due to their flexibility and broad applications in speech and image processing. The proposed bounded support mixtures are applied in various speech and images datasets to create leaning applications to demonstrate the effectiveness of proposed approach. Mixtures of bounded Gaussian and bounded Laplace are also applied in feature extraction and data representation techniques, which further improves the learning and modeling capability of underlying models. The proposed feature representation via bounded support mixtures is applied in both speech and images datasets to examine its performance. Automatic selection of number of mixture components is very important in clustering and parameter learning is highly dependent on model selection and it is proposed for mixture of bounded Gaussian and bounded asymmetric generalized Gaussian using minimum message length. Proposed model selection criterion and parameter learning are simultaneously applied in speech and images datasets for both models to examine the model selection performance in clustering.

To my parents

&

To all my teachers

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to Prof. Nizar Bouguila. He provided me an excellent opportunity to pursue PhD program in his research lab, introduced me to very interesting research topics, guided me throughout, and generously provided great opportunities to excel. Not only I substantially benefitted from his vast technical knowledge, but I also learnt how a gentleman should behave when he is in authority. I am proud to be his student.

Many thanks to my committee members, namely Dr. Mohand Said Allili, Dr. Abdessamad Ben Hamza, Dr. Jia Yuan Yu, Dr. Wahab Hamou-Lhadj and Dr. Ali Dolatabadi, for their guidance and comments which have significantly helped to improve my work.

I would also like to thank the University of Azad Jammu and Kashmir, Pakistan who gave me the opportunity to study in Canada through the faculty development program, and Concordia University for international tuition remission award.

I would not forget to thank my friends with whom I have spent many years in the lab. Especially, I would like to thank Taoufik Bdiri, Mohamed Al Mashrgy, Walid Masoudi, Basem Alghabashi, Nuha Zamzami, Samr Ali, Elise Epaillard, Eddy K. Ihou, Dinh Hieu Nguyen, Mathin Henry Kamal Maanicshah, Narges Manouchehri, Meeta Kalra, and Jaspreet Singh Kalsi for providing great company and maintaining conducive learning environment in the lab. I would also like to thank my friends Haroon Gardezi, Raja Abullah Ahmad and Asif Raza Butt, for making my stay in Montreal memorable and joyous.

Naturally, best of my thanks go to my family. I would never have completed this thesis without the endless support and love of my father, and my mother. None of my success would have been possible without them being beside me. I also thank my brother, and my sisters for their continuous support and encouragements. Many thanks go to my beloved wife for her support and love, and for being beside me at the most difficult times during this thesis.

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter 1

Introduction

Recently, due to the rapid development in sensor networks and communication technologies, data storage and data collection capabilities have been increased. Due to the accumulation of large databases, data analysis and modeling provide a platform to revolutionize all science and engineering domains and provide great opportunities in a many areas including, e-commerce, industry, medical and social media [4, 5]. Several applications of data analysis in different areas have increased the demand for development of advanced data mining techniques. Development in this area is needed to improve information retrieval, knowledge discovery and learning from the patterns in data for making smart and intelligent decisions [5–7]. In last few years, application of machine learning has tremendously increased to extract information and patterns from data. The information and patterns extracted from data help machines to learn and improve their intelligence which further assist in smart decision making in broad areas of applications. Development of machine learning algorithms and techniques has become an active area of research in last few decades due to fast growing need in application of AI in different areas. Machine learning algorithms are backbone of AI in numerous applications of data mining in all engineering and natural sciences domains, including business, finance, physical, cognitive, biological and biomedical sciences [4, 8, 9].

In machine learning, data clustering is defined as unsupervised classification of patterns into groups which are called clusters. The task of data clustering has been addressed in different ways and in many research fields [10]. Clustering algorithms aim to classify elements of data into categories, or clusters based on their similarity [11], where degree of similarity is represented by an affinity function, which takes a data-pair as its input [12]. Data clustering has been extensively used in image segmentation, object and character recognition, information retrieval and many more applications of speech and image processing [13, 14]. Many clustering techniques have been proposed in last few decades to solve different pattern recognition tasks. As cluster analysis is quite

prevalent for multivariate data [15], it is not limited to few algorithms or techniques. K-Means is a well known and a popular algorithm for data clustering but it has several limitations in cluster analysis such as sensitivity to the initialization and outliers, choosing the number of clusters and problems with high dimensional data.

Finite mixture model is well known clustering approach that provides solution to many problems observed with K-Means. Mixture models can be employed to model complex data sets by assuming that each observation of data has arisen from one of the different groups or components [16, 17]. Mixture model is a probabilistic approach, which is capable of utilizing prior information to model uncertainty [18–20]. Popular applications of mixture models include anomaly detection [21], image segmentation [22, 23], biomedical diagnostics and prediction of diseases i.e. Alzheimer [24, 25], speech recognition, speaker identification and classification [1, 26–29].

1.1 Finite Mixture Models and Parameters Learning via EM

Finite mixture models are created by considering a linear combinations of a finite number of basic distributions. These densities are called components of the mixture model. If we consider that, $\vec{X} = [X_1, ..., X_D]^T$ is a D-dimensional vector, which follows a K component mixture distribution, then its probability density function can be written as:

$$p(\vec{X}|\Theta) = \sum_{j=1}^{K} p(\vec{X}|\xi_j)p_j \tag{1.1}$$

with the constraints that $p_j \geq 0$ and $\sum_{j=1}^{K} p_j = 1$. In Eq. (1.1), ξ_j represents the mixture model parameters of jth component, p_j is mixing weight, $\Theta = \{\xi_1, ..., \xi_K, p_1, ..., p_K\}$ represents the complete set of parameters to characterize the mixture model and $K \geq 1$ is number of components in the mixture model [30–34]. For data $\mathscr{X} = (\vec{X}_1, ..., \vec{X}_N)$, having a mixture of K distributions, the model is given by:

$$p(\mathscr{X}|\Theta) = \prod_{i=1}^{N} \sum_{j=1}^{K} p(\vec{X}_i|\xi_j)p_j \tag{1.2}$$

In a finite mixture, parameters estimation is performed by computation of maximum likelihood (ML) estimate which is described as follows:

$$\hat{\Theta}_{\text{ML}} = \arg\max_{\Theta} \{p(\mathscr{X}|\Theta)\} \tag{1.3}$$

The ML estimate for the computation of parameters of mixture model cannot be found analytically and usual choice for estimation is EM algorithm [30, 31, 35, 36], which is an iterative procedure to

find the local maxima of log-likelihood. In EM algorithm, it is assumed that our data are incomplete and in mixture models, the missing part is label associated with each data sample. In EM algorithm, parameters are estimated in two steps which are called expectation (E-step) and maximization (M-step). In E-step, conditional expectation of complete log-likelihood is computed whereas in M-step, parameters of mixture model are updated [30, 35, 37].

1.2 Probability Density Function Selection

Gaussian mixture model is one of the most popular approaches in this family which utilizes Gaussian distribution for modeling the data and it has been employed in many applications. Although, GMM has been the first choice for many clustering applications, it has some limitations due to its high sensitivity to outliers. The choice of distribution in a mixture model is very important and it depends on many factors including nature of data, modeling capabilities, handling issues posed by outliers, ability to cluster high dimensional datasets and ease for applying in real applications. Mixture models are an active area of research and a lot of work has been performed to introduce new techniques and algorithms to deal with more complex and challenging tasks in data modeling [38–44]. Student's t mixture model (SMM) was introduced to improve the robustness of mixture model for different shapes of data [16, 18, 45, 46]. Generalized Gaussian mixture model has been proposed to improve the data modeling capabilities since generalized Gaussian distribution has an extra shape parameter that helps to model Laplacian and Gaussian data [47–51]. A mixture of Laplace distributions was introduced as a generalization to k-median algorithm, which can be used in clustering applications primarily to handle outliers and this approach proves to be very effective in many data modeling applications where density of data is more close to Laplace distribution [52–54]. With a mixture of Gaussians, it is assumed that component distribution is symmetric in nature. In many applications, it is possible that data might not be symmetrical and GMM could not model the data very well. For applications with asymmetric nature of data, symmetric distribution is better choice for data modeling [32, 55–57]. One such example is asymmetric Gaussian distribution which has two standard deviation parameters on the left and right side of distribution, which make it possible to model the asymmetric data [32]. It should be noted that by considering the equal left and right standard deviations, it turns out to be a symmetric distribution. Another such example is asymmetric generalized Gaussian distribution which also has a shape parameter. By considering the equal left and right standard deviations in this distribution will make it a generalized Gaussian distribution which can be used further to generalize the Laplace and Gaussian distributions by changing the value of shape parameter [32, 56, 58, 59].

1.3 Bounded Support Mixture Models

One limitation with above mentioned models is that their distributions have unbounded support range $(-\infty,+\infty)$. In many real applications, it is observed that data lie within bounded support regions [60–62]. Bounded support mixture has been proposed to overcome the problems associated with unbounded mixture models [18, 61–66], which has demonstrated its success for many speech processing applications [67]. In majority of data applications modeled through Gaussian mixtures, the problems in modeling posed by unbounded support have not been addressed. Considering bounded support in a mixture model results in a modified probability density function and a modified expectation maximization algorithm. In order to accurately model the data by considering a bounded support region (∂_j) in $\mathbb{R}$, an indicator function $\mathrm{H}(X|j)$ for each component of mixture model is defined as:

$$\mathrm{H}(X|j) = \begin{cases} 1 & \text{if} \quad X \in \partial_j \\ 0 & \text{otherwise} \end{cases} \tag{1.4}$$

If we consider the finite mixture model discussed in Eq. (1.1) with uni-variate data, then bounded support model can be obtained by multiplying unbounded mixture model with $\mathrm{H}(X|j)$ and normalize

$$\begin{aligned} p(X_i|\Theta) &= \frac{p(X|\Theta)\mathrm{H}(X|j)}{\int_{\mathbb{R}} p(\mathrm{u}|\Theta)\mathrm{H}(\mathrm{u}|j)d\mathrm{u}} \\ &= \begin{cases} \frac{\sum_{j=1}^{K} p(X|\xi_j)p_j}{\sum_{j=1}^{K} p_j \int_{\partial_j} p(\mathrm{u}|\xi_j)d\mathrm{u}} & \text{if} \quad X \in \partial_j \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} \frac{\sum_{j=1}^{K} p(X|\xi_j)p_j}{\sum_{j=1}^{K} p_j f_j} & \text{if} \quad X \in \partial_j \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{1.5}$$

where $f_j = \int_{\partial_j} p(\mathrm{u}|\xi_j)d\mathrm{u}$ is the share of $p(\mathrm{u}|\xi_j)$ that belongs to the support region. The denominator in Eq. (1.5) guarantees that $p(X|\Theta)$ integrates to unity and can be indicated as $Fs = \sum_{j=1}^{K} p_j f_j$ of the whole underlying mixture model that belongs to the support region. The bounded support mixture model can be rewritten as:

$$p(X|\Theta) = \frac{\mathrm{H}(X|j)\sum_{j=1}^{K} p(X|\xi_j)p_j}{Fs} = \sum_{j=1}^{K} \pi_j \frac{p(X|\xi_j)}{f_j}\mathrm{H}(X|j), \quad \text{where} \quad \pi_j = p_j\frac{f_j}{Fs} \tag{1.6}$$

The parameters of a bounded support mixture model can be estimated by maximum likelihood approach using EM algorithm for optimization of estimated parameters as described in [18, 62]. Due to flexibility in modeling and requirements in many applications, Gaussian, Laplace, generalized Gaussian, asymmetric Gaussian and asymmetric generalized Guaissain distributions are adopted

to create bounded support mixtures in this thesis.

1.4 Selection of number of components

In the clustering process performed by finite mixture models, a very important problem is to find the best number of components of mixture model called model selection. There is a trade off in selection of number of components in a mixture model. By selecting too many components, mixture model may lead to an over-fitting, while selecting too few components may not be flexible enough to effectively model the behavior of different patterns of data [30, 68, 69]. A lot of research has been performed in the past to find the optimal number of components in a mixture model. There are several deterministic and stochastic algorithms to estimate the optimal number of mixture components [30]. An approximate Bayesian criteria was introduced in [39], which was later termed as Laplace-empirical criterion [31] and it has been proven very effective in model selection in many studies [31]. Bayesian inference criterion was introduced in [70–73] which got a lot of attention in many applications due to its simplicity. Minimum description length (MDL) is based on information theory and it was presented in [74]. Akaike's information criterion (AIC) and informational complexity criterion (ICOMP) were introduced in [75] and [76], respectively and these approaches also use concepts from information theory. Approximate weight of evidence (AWE) and classification likelihood criterion (CLC) were introduced in [77, 78] and these approaches are dependent on the use of complete likelihood of data. Normalized entropy criterion (NEC) and integrated classification likelihood (ICL) method were introduced in [79, 80] and [81], respectively. Minimum message length (MML) criterion to find the optimal number of components in mixtures was introduced in [82–84], which has been found very effective in model selection in numerous studies and applications in the past.

1.5 Contributions

The aim of this thesis is to propose several novel approaches for multidimensional data modeling and clustering. The overall contributions of this thesis are as follows:

- ☞ **Multivariate Bounded Gaussian Mixture Model with Minimum Message Length Criterion for Model Selection**
  We propose BGMM to speech and image processing applications for clustering and further extend our experiments by proposing it to code-book generation for speech and image datasets. A model selection criterion is also proposed and tested with different datasets.

☞ **Speaker Verification Using Adapted Bounded Gaussian Mixture Model**

We proposed a speaker verification framework using BGMM. In the proposed framework, a universal background model is trained via BGMM. In adapted speaker approach, hypothesized speaker model is derived by adapting the parameters of BGMM based UBM using speaker's training speech and maximum a posteriori (MAP).

☞ **Bounded Laplace Mixture Model with Applications to Image Clustering and Content Based Image Retrieval**

We proposed BLMM for texture images categorization and CBIR. In both frameworks, feature are also extracted using BLMM and experiments are conducted with 3 datasets to demonstrate the effectiveness of proposed approach in feature extraction, image categorization and CBIR.

☞ **Multivariate Bounded Support Laplace Mixture Model**

We extended our previous work and proposed BLMM in clustering synthetic data, 10 medical datasets and feature extraction in wavelet domain, texture image categorization and CBIR. We also introduced 3 different similarity measures for CBIR and proposed a closed form solution for one measure. Texture image categorization and CBIR experiments are conducted with 5 different datasets.

☞ **Texture Image Categorization in Wavelet Domain via Naive Bayes Classifier Based on Laplace and Generalized Gaussian Distribution**

We proposed Naive Bayes classifiers using Laplace and generalized Gaussian distributions for texture image categorization with feature extraction using BLMM adopted from previous work. Experiments are conducted on 3 different datasets with texture images.

☞ **Unsupervised keyword spotting using bounded generalized Gaussian mixture model with ICA**

We proposed an ICA mixture in unsupervised keyword spotting for the generation of posteriorgrams for test speech files and reference keyword examples. The posteriorgrms are compared using segmental dynamic time warping for making a decision about the occurrence of keywords in speech data. The experiments are performed using TIMIT speech corpus.

☞ **Speaker Classification via Supervised Hierarchical Clustering Using ICA Mixture Model**

In this work, a speaker classification framework is developed using ICA mixture model and experiments are conducted using TSP and TIMIT speech datasets. The experiments are performed for male and female speakers categorization and 10 speakers classification.

☞ **Blind Source Separation as Pre-processing to Unsupervised Keyword Spotting via an ICA Mixture Model**
This work is an extension of unsupervised keyword spotting, where test data is mixed with other speech files and it is observed that source mixing cause a decrease in recognition for keyword spotting and application of BSS in the pre-processing improves the keyword spotting task.

☞ **Bounded Generalized Gaussian Mixture Model with ICA**
This work is an extension of BSS and unsupervised keyword spotting. Several experiments are performed to see the effectiveness of ICA mixture in BSS. BSS is also applied to unsupervised keyword spotting and performance is observed with BSS being applied in the pre-processing with test data is mixed with other sources.

☞ **Multivariate Bounded Asymmetric Gaussian Mixture Model**
We proposed BAGMM, which uses ML and EM along Newton-Raphson for parameter estimation. The proposed model is applied in textual spam detection, object clustering and texture image clustering and experiments are conducted with 4 different datasets.

☞ **Multivariate Bounded Support Asymmetric Generalized Gaussian Mixture Model with Model Selection using Minimum Message Length**
We proposed Bounded Support Asymmetric Generalized Gaussian Mixture Model, which uses ML and EM along Newton-Raphson for parameter estimation. The proposed model is applied in image spam detection, object recognition and visual scene categorization. A model selection criteria is also proposed using MML with is tested datasets from clustering experiments.

1.6 Thesis Overview

The organization of this thesis is as follows:

❑ Chapter 1, contains an introduction to mixture models and an overview over the thesis.

❑ In chapter 2, BGMM is proposed to different clustering applications in speech and images datasets. For experiments, male and female speakers categorization, spoken and written digits recognition and clustering of fashion images are chosen from speech and images datasets. Further, BGMM is also applied in code-book generation to improve pre-processing and experiments are conducted on same datasets for speech and images. A model selection criterion is also proposed using MML for BGMM, which is applied in medical datasets and all the

datasets used in clustering experiments from speech and images. This work is submitted to **Expert Systems journal**. BGMM is also proposed for speaker verification in adapted speaker model and universal background model. This work is published in **2018 IEEE International Conference on Information Reuse and Integration (IRI)** [67].

- In chapter 3, BLMM is proposed, which adopt maximum likelihood via EM along with Newton Raphson for its parameter estimation. Proposed model is applied in clustering synthetic data, medical datasets, feature extraction and modeling the texture images and CBIR. In this chapter, 3 similarity measures are also used and a close form solution for one of the measures is proposed. Initial results of this research are published in **2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)** [85] and complete work is submitted to **Soft Computing journal**. The features extracted from BLMM in wavelet domain are further used in supervised learning and modeled through proposed Naive Bayes classifiers using Laplace and generalized Gaussian distributions. This work is published in **2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)** [86].

- In chapter 4, multivariate bounded generalized Gaussian mixture model with ICA is proposed which is initially applied to unsupervised keyword spotting. This work is published in **2015 IEEE Global Conference on Signal and Information Processing** [2]. The proposed model was applied in speaker classification via hierarchical clustering and published in **2016 7th International Conference on Image and Signal Processing (ICISP)** [87]. This was further applied to BSS and unsupervised keyword spotting and initial results were published in **2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)** [88] and the complete work was published in **Neural Processing Letters** [89].

- In chapter 5, two asymmetric bounded mixture are proposed. First model is bounded asymmetric Gaussian mixture which uses ML and EM with Newton-Raphson for parameter estimate. The proposed model is applied to textual spam detection, object categorization and texture image clustering. Overall 4 different datasets are used in experiments and this work is published as a book chapter in **Mixture Models and Applications** [90]. Our second proposed model is bounded asymmetric generalized Gaussian mixture which also uses ML and EM with Newton-Raphson for parameter estimate. The model is applied in image spam detection, object recognition, and visual scene categorization. A model selection criteria using MML is also proposed tested with all the datasets used in clustering experiments. This work is submitted to **Multimedia Tools and Applications journal**.

- Chapter 6 summarizes our contributions and present some potential future works.

Chapter 2

Multivariate Bounded Support Gaussian Mixture Model

Bounded support Gaussian mixture model (BGMM) has been proposed for data modeling as an alternative to unbounded support mixture models for the cases when the data lies in bounded support. In this chapter, we propose applications of multivariate BGMM in data clustering for more insightful analysis of the model. We also propose minimum message length (MML) criterion for model selection in data clustering using multivariate BGMM. The presented model is applied to data clustering in several speech (TSP and Spoken Digits) and image databases (MNIST and Fashion MNIST). We also propose the application of BGMM in code-book generation at feature extraction phase. Inspired by the success of bag of visual words approach in computer vision, it is also introduced in speech data representation and validated through experiments presented in this chapter. For validation of model selection criterion, MML is applied to different medical, speech and image datasets. Experimental results obtained during the model selection through MML are further compared with 7 different model selection criteria. The results presented in the chapter demonstrate the effectiveness of BGMM.

We also propose the application of bounded Gaussian mixture model (BGMM) to speaker verification. In the proposed approach, BGMM is employed for universal background model (UBM) and adapted speaker model. The proposed UBM is a large BGMM trained to represent speaker-independent distribution of features. In adapted speaker approach, hypothesized speaker model is derived by adapting the parameters of BGMM based UBM using speaker's training speech and maximum a posteriori (MAP). We have applied TIMIT and TSP speech corpora for the development of UBM and further testing of speaker verification by adapted speaker model. The proposed framework has demonstrated its effectiveness by improved speaker detection rate.

2.1 Introduction

In majority of data applications modeled through Gaussian mixtures, the problems in modeling posed by unbounded support have not been addressed. Considering bounded support in a mixture model results in a modified probability density function and a modified expectation maximization algorithm. BGMM was proposed in [62] for speech data applications and it was applied to images in [60], but it has not been further discussed and applied to more applications in speech and image datasets. We intend to apply BGMM in various high dimensional datasets for more insightful analysis and also propose model selection criterion for BGMM to accurately find the number of clusters in a dataset to support unsupervised learning of mixture model.

In this chapter, BGMM is further explored in clustering applications and analyzed for multi-cluster and high dimensional datasets in speech and image processing applications. For speech data clustering, TSP and Spoken Digits datasets are selected and MNIST and Fashion MNIST datasets are employed for image data clustering [91–94]. TSP dataset is composed of two categories of data namely male and female speakers. As first step in our data clustering applications, BGMM is applied on TSP dataset to categorize the speech signals of male and female speakers. Inspired from bag of visual words (BoVW) approach in computer vision and image processing and bag of words (BoW) approach in natural language processing (NLP), BoW approach is extended to speech dataset [95–99] and we applied Mel Frequency Cepstral Coefficients (MFCC) to represent the speech files before BoW stage. For speech data representation, this approach is termed as bag of audio words (BoAW) and it has been successfully applied in many speech processing application recently [100–102]. As next step in our data clustering applications, BGMM is applied to Spoken Digits dataset which is composed of 10 categories of audio digits and features are extracted in a similar manner as described for TSP dataset. Clustering is performed by selecting the data from 2, 3, 4, 5 and 10 different categories to examine the behavior and modeling capabilities of BGMM in a multi-cluster scenario. As our next step, BGMM is applied to MNIST and Fashion MNIST datasets where each dataset is composed of 10 different categories. BoVW approach is used to represent the data which are generated through Scale-Invariant Feature Transform (SIFT) descriptors for each image. Similar to clustering experiments on Spoken Digits dataset, data from 2, 3, 4, 5 and 10 different classes are chosen to examine the clustering performance of BGMM in a multi-cluster and high dimensional scenario.

BoW approach is used to represent the data for both kinds of applications (speech and image) discussed in this chapter. Code-Book generation through BoW requires clustering which is mostly done by K-Means. In literature, it is proved by several studies that application of mixtures to replace the K-Means in Code-Book generation can improve the effectiveness of data representation for model learning [59, 103]. In this chapter, we also propose the application of mixture model

in Code-Book generation instead of K-Means and we examine the performance of BGMM in clustering the features to create BoW for more effective representation of data. We have previously examined the performance of BGMM in second stage, where it is applied to cluster the data after feature extraction from BoW stage. In order to differentiate these experiments from the above mentioned set of experiments, application of mixtures to clusters the data in BoW is called stage 1 clustering and application of mixture in clustering the data after feature extraction to categorize it into different groups is called stage 2 clustering. Several experiments are performed to examine the performance of proposed approach using BGMM in Code-Book generation and it is compared with K-Means and GMM in a similar setting. In these experiments, 2 categories of data are selected from TSP dataset, and 10 categories of data are chosen from Spoken Words, MNIST and Fashion MNIST datasets. For these experiments we have 3 comparison options at stage 1 (BGMM, GMM, K-Means) and 2 options at stage 2 (BGMM, GMM) clustering.

MML criterion for model selection is proposed for BGMM in this chapter and validated through application on several medical datasets which are found to be difficult in clustering and model selection by conventional approaches. In order to validate the proposed model selection criterion, similar experiments are conducted with different approaches selected from literature. We also extended our experiments on model selection for speech and images datasets discussed in the clustering process. TSP dataset with 2 classes is selected for our experiments, whereas parts of rest of datasets are chosen with 2, 3, 4 and 5 classes to validate the model selection and compared with other algorithms.

2.2 Multivariate Bounded Support Gaussian Mixture Model

Bounded support mixtures were introduced to overcome the problems and challenges posed by unbounded support range of underlying distribution when observed data have a bounded support range. A bounded Gaussian mixture was presented in [62]. In this section, we present BGMM and its parameters estimation using maximum likelihood estimate via EM algorithm.

2.2.1 Mixture of Multivariate Gaussian Distributions

Mixture of Gaussian distributions using EM algorithm was introduced in [104], however one of the first major studies that used the mixture of Gaussian distributions was around 125 years ago by the renowned biometrician Karl Pearson [105]. If a vector $\vec{X}$, follows a K component mixture which is represented by Eq. (1.1), then Gaussian distribution for each component of mixture model is given

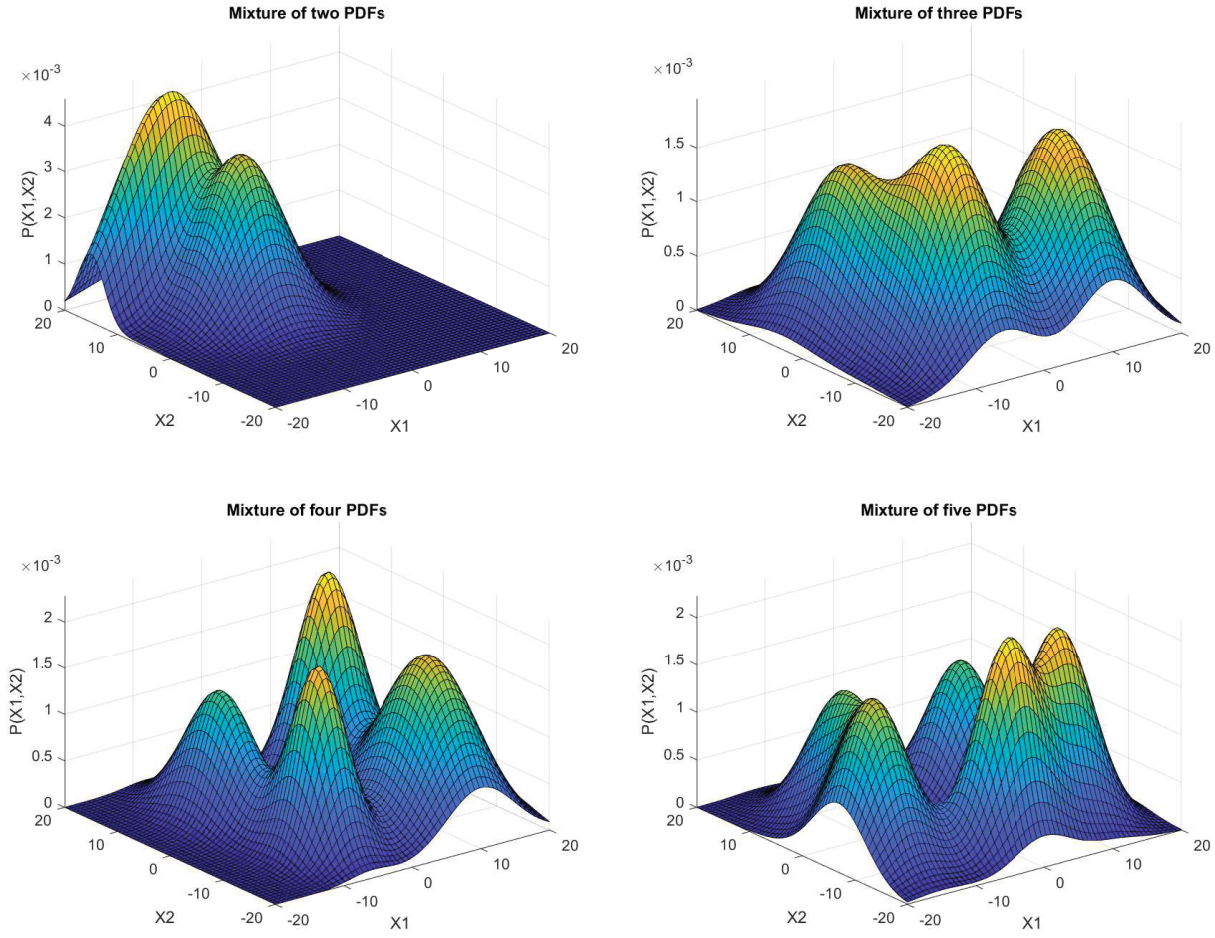


Figure 2.1: Example demonstrating mixtures of different numbers of Gaussian distributions for two dimensional data

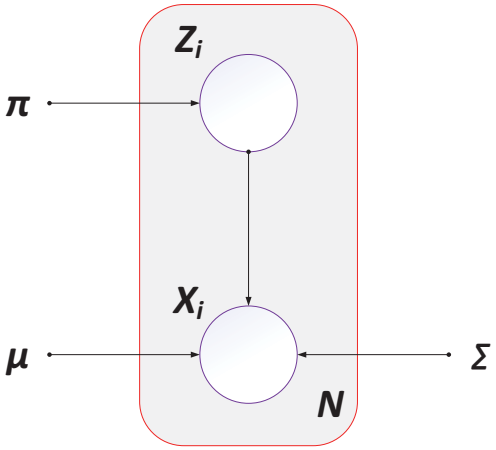


Figure 2.2: Graphical representation of Gaussian mixture model

as follows:

$$f(\vec{X}|\vec{\mu}_j,\Sigma_j) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma_j|^{1/2}}\exp\left\{-\frac{1}{2}(\vec{X}-\vec{\mu}_j)^{\mathrm{T}}{\Sigma_j}^{-1}(\vec{X}-\vec{\mu}_j)\right\} \tag{2.1}$$

where $\vec{\mu}_j$ and Σ_j are mean and co-variance of Gaussian distribution for each component. For estimation of parameters in Gaussian mixture model, ML approach via EM algorithm gives a closed form solution for all parameters of mixture model as follows:

$$\hat{\vec{\mu}}_j = \frac{1}{\sum_{i=1}^{N}\hat{Z}_{ij}}\sum_{i=1}^{N}\hat{Z}_{ij}\vec{X}_i \tag{2.2}$$

$$\hat{\Sigma}_j = \frac{1}{\sum_{i=1}^{N}\hat{Z}_{ij}}\sum_{i=1}^{N}\hat{Z}_{ij}\left(\vec{X}_i-\vec{\mu}_j\right)\left(\vec{X}_i-\vec{\mu}_j\right)^T \tag{2.3}$$

$$\hat{p}_j = \frac{1}{N}\sum_{i=1}^{N}p(j|\vec{X}_i) \tag{2.4}$$

where $p(j|\vec{X}_i)$ is posterior probability estimated for GMM and N represents the total number of observations. Some examples of data modeling via Gaussian mixture for two dimensional data for several numbers of components are shown in Fig. (2.1) and graphical representation of Gaussian mixture model is given in Fig. (2.2).

2.2.2 Mixture of Bounded Gaussian Distributions

For BGMM, the term $p(\vec{X}|\xi_j)$ in Eq. (1.2), is bounded Gaussian distribution (BGD). For defining the BGD, it is required to present the indicator function defining the boundary conditions. For each component of mixture model, ∂ is defined as bounded support region in $\mathbb{R}$, and the indicator function is defined as:

$$\mathrm{H}(\vec{X}|j) = \begin{cases} 1 & \text{if} \quad \vec{X}\in\partial_j \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

By using the indicator function $\mathrm{H}(\vec{X}|j)$, BGD is defined as:

$$p(\vec{X}|\xi_j) = \frac{f(\vec{X}|\vec{\mu}_j,\Sigma_j)\mathrm{H}(\vec{X}|j)}{\int_{\partial_j} f(\vec{\mathrm{u}}|\vec{\mu}_j,\Sigma_j)d\mathrm{u}} \tag{2.6}$$

where $f(\vec{X}|\vec{\mu}_j,\Sigma_j)$ represents the multivariate Gaussian distribution as given in Eq. (2.1). In Eq. (2.6), $\xi_j = (\vec{\mu}_j,\Sigma_j)$ is set of parameters with $\vec{\mu}_j = (\mu_{j1},...,\mu_{jD})$ and Σ_j as D-dimensional mean and $D\times D$ co-variance matrix of the BGD, respectively [62]. The term $\int_{\partial_j} f(\vec{\mathrm{u}}|\vec{\mu}_j,\Sigma_j)d\mathrm{u}$ in Eq. (2.6) is the normalization constant that indicates the share of $f(\vec{X}|\vec{\mu}_j,\Sigma_j)$ which belongs to the support

region ∂_j.

We introduce stochastic indicator vectors $\vec{Z}_i = (Z_{i1}, ..., Z_{iK})$, one vector for each observation of data. The role is to encode the membership of each observation into the relative component of the mixture model. In other words, Z_{ij}, the unobserved variable in each indicator vector, equals 1 if $\vec{X}_i$ belongs to class j and 0, otherwise. The complete data likelihood is given below.

$$p(\mathscr{X}, \mathscr{Z}|\Theta) = \prod_{i=1}^{N}\prod_{j=1}^{K}\Big(p(\vec{X}_i|\xi_j)p_j\Big)^{Z_{ij}} \tag{2.7}$$

where Z_{ij} is the posterior probability and its expectation can be written as:

$$\hat{Z}_{ij} = p(j|\vec{X}_i) = \frac{p(\vec{X}_i|\xi_j)p_j}{\sum_{j=1}^{K} p(\vec{X}_i|\xi_j)p_j} \tag{2.8}$$

and $\mathscr{Z} = \{\vec{Z}_1, ..., \vec{Z}_N\}$.

2.2.3 Parameters Learning

The parameters are estimated from the maximization of log-likelihood function. The log-likelihood function can be written as:

$$\mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{K}\hat{Z}_{ij}\log\Big(p(\vec{X}_i|\xi_j)p_j\Big) \tag{2.9}$$

$$\mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{K}\hat{Z}_{ij} \times \left\{\log p_j + \log f(\vec{X}_i|\xi_j) + \log \mathrm{H}(\vec{X}_i|\Omega_j) - \log\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right\} \tag{2.10}$$

The complete-data log-likelihood can be maximized with respect to the model parameters. This can be done by taking the gradient of the log-likelihood with respect to p_j, $\vec{\mu}_j$ and Σ_j. The parameters estimation for bounded support Gaussian mixture model is given below.

2.2.3.1 Mixing parameter estimation

For the estimation of mixing parameter, in order to ensure the constraints $p_j > 0$ and $\sum_{j=1}^{M} p_j = 1$, a Lagrange multiplier is introduced while estimating p_j. Thus, the augmented log-likelihood

function can be expressed by:

$$\Phi(\mathscr{X},\mathscr{Z},\Theta,\Lambda)=\sum_{i=1}^{N}\sum_{j=1}^{K}\hat{Z}_{ij}\log\left(p(\vec{X}_i|\xi_j)p_j\right)+\Lambda\left(1-\sum_{j=1}^{K}p_j\right) \tag{2.11}$$

where Λ is the Lagrange multiplier. Differentiating the augmented function with respect to p_j and equating it to zero, we get the estimated value of p_j as follows:

$$\hat{p}_j=\frac{1}{N}\sum_{i=1}^{N}p(j|\vec{X}_i) \tag{2.12}$$

The complete derivation procedure is given in Appendix A.1.

2.2.3.2 Mean Parameter estimation

The new value of mean, can be estimated by maximizing the log-likelihood function given in Eq. 2.10 with respect to $\vec{\mu}_j$.

$$\frac{\partial\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial\vec{\mu}_j}=0 \tag{2.13}$$

The computation of log-likelihood derivative with respect to $\vec{\mu}_j$ is given in Appendix A.2 and by using this derivative in Eq. (2.13), an estimate of $\hat{\vec{\mu}}_j$ can be compute with procedure given in Appendix A.3. An estimate of $\hat{\vec{\mu}}_j$ is as follows:

$$\hat{\vec{\mu}}_j=\frac{\sum_{i=1}^{N}\hat{Z}_{ij}\left\{\vec{X}_i-\frac{\int_{\partial_j}f(\vec{\mathrm{u}}|\xi_j)(\vec{\mathrm{u}}-\vec{\mu}_j)d\mathrm{u}}{\int_{\partial_j}f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}}\right\}}{\sum_{i=1}^{N}\hat{Z}_{ij}} \tag{2.14}$$

Note that, in (2.14), the term $\int_{\partial_j}f(\vec{\mathrm{u}}|\xi_j)(\vec{\mathrm{u}}-\vec{\mu}_j)d\mathrm{u}$ is the expectation of function $(\vec{\mathrm{u}}-\vec{\mu}_j)$ under the probability distribution $f(\vec{\mathrm{u}}|\xi_j)$. Then, this expectation can be approximated as:

$$\int_{\partial_j}f(\vec{\mathrm{u}}|\xi_j)(\vec{\mathrm{u}}-\vec{\mu}_j)d\mathrm{u}\approx\frac{1}{M}\sum_{m=1}^{M}(\mathrm{s}_{m_j}-\mu_{jd})\mathrm{H}(\mathrm{s}_{m_j}|\Omega_j) \tag{2.15}$$

where $s_{m_j}\sim f(\vec{\mathrm{u}}|\xi_j)$ is a set of random variables drawn from the Gaussian distribution for the jth component of the mixture model. The set of data with random variables have M vectors with D dimensions. M is a large integer chosen to generate the set of random variables.
Similarly, term $\int_{\partial_j}f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}$ in (2.14) can be approximated as:

$$\int_{\partial_j}f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\approx\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{s}_{m_j}|\Omega_j) \tag{2.16}$$

$$\hat{\vec{\mu}}_j = \frac{\sum_{i=1}^{N} \hat{Z}_{ij} \left\{ \vec{X}_i - \frac{\sum_{m=1}^{M} (\mathrm{s}_{m_j} - \vec{\mu}_j) \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j)}{\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j)} \right\}}{\sum_{i=1}^{N} \hat{Z}_{ij}} \tag{2.17}$$

2.2.3.3 Co-variance Matrix estimation

The new value of co-variance $\hat{\Sigma}_j$, can be estimated by maximizing the log-likelihood function given in Eq. 2.10 with respect to Σ_j.

$$\frac{\partial \mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta)}{\partial \Sigma_j} = 0 \tag{2.18}$$

The computation of derivative of log-likelihood with respect to Σ_j is given in Appendix A.4 and by using this derivative in Eq. (2.18), we can estimate $\hat{\Sigma}_j$ as follows:

$$\hat{\Sigma}_j = \frac{\sum_{i=1}^{N} \hat{Z}_{ij} \left\{ (\vec{X}_i - \vec{\mu}_j)(\vec{X}_i - \vec{\mu}_j)^T - \frac{\int_{\partial_j} (-\Sigma_j + (\vec{\mathrm{u}} - \vec{\mu}_j)(\vec{\mathrm{u}} - \vec{\mu}_j)^T) f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \right\}}{\sum_{i=1}^{N} \hat{Z}_{ij}} \tag{2.19}$$

The estimation procedure using maximum likelihood for $\hat{\Sigma}_j$ is described in Appendix A.5. The term $\int_{\partial_j} f(\mathrm{u}|\xi_j)(\vec{\mathrm{u}} - \vec{\mu}_j)(\vec{\mathrm{u}} - \vec{\mu}_j)^T d\mathrm{u}$ can be approximated as below:

$$\int_{\partial_j} (-\Sigma_j + (\vec{\mathrm{u}} - \vec{\mu}_j)(\vec{\mathrm{u}} - \vec{\mu}_j)^T) f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \approx \tag{2.20}$$
$$\frac{1}{M} \sum_{m=1}^{M} (-\Sigma_j - (\mathrm{s}_{m_j} - \vec{\mu}_j)(\mathrm{s}_{m_j} - \vec{\mu}_j)^T) \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j)$$

where $\mathrm{s}_{m_j} \sim f(\vec{\mathrm{u}}|\xi_j)$ is a set of random variables drawn from the Gaussian distribution for the particular component j of the mixture model.

$$\hat{\Sigma}_j = \frac{\sum_{i=1}^{N} \hat{Z}_{ij} \left\{ (\vec{X}_i - \vec{\mu}_j)(\vec{X}_i - \vec{\mu}_j)^T - \frac{\sum_{m=1}^{M} (-\Sigma_j - (\mathrm{s}_{m_j} - \vec{\mu}_j)(\mathrm{s}_{m_j} - \vec{\mu}_j)^T) \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j)}{\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j)} \right\}}{\sum_{i=1}^{N} \hat{Z}_{ij}} \tag{2.21}$$

The complete learning procedure for BGMM is given in Algorithm 1, where t_{min} is minimum threshold used to examine convergence criteria in each iteration.

Algorithm 1 Model Learning with BGMM

Input:Dataset $\mathscr{X} = \{\vec{X}_1, \ldots, \vec{X}_N\}$, t_{min}.
Output: Θ.
{Initialization}: K-Means Algorithm.
  K-Means Algorithm (Computation of $\vec{\mu}_1, \ldots, \vec{\mu}_K$ & cluster assignment)
    for all $1 \leq j \leq K$ do
      Computation of p_j
      Computation of Σ_j
    end for
{Expectation Maximization}:
while relative change in log-likelihood $\geq t_{min}$ do
  {[**E Step**]}:
    for all $1 \leq j \leq K$ do
      Compute $p(j|\vec{X}_i)$ for $i = 1, \ldots, N$. using Eq. (2.8).
    end for
  {[**M step**]}:
    for all $1 \leq j \leq K$ do
      Update the mixing parameter $\hat{p}_j$ using Eq. (2.12).
      Update the mean $\hat{\vec{\mu}}_j$ using Eq. (2.17).
      Update Co-variance matrix $\hat{\Sigma}_j$ using Eq. (2.21).
    end for
end while

2.3 Experiments and results for Clustering via BGMM applied to speech and images datasets

We propose the application of BGMM to speech and image datasets for categorizing the data in an unsupervised manner. For the validation of clustering performance of BGMM, we have chosen speech and image datasets. Speech data clustering is backbone in many speech processing applications including speaker verification, classification, speech recognition and dialog systems. TSP and Spoken Digits datasets are selected for our experiments. TSP dataset is composed of categories from male and female speakers. Spoken Digits dataset is composed of 10 categories of data from different speakers.

Image clustering is one of the most challenging tasks in computer vision. In case of hand-written characters, as each person has unique writing style, the same digit could vary from one individual to another with different angles, stress and complexity. In this experiment, we focus on two widely used datasets: MNIST containing real-life handwritten digits and Fashion MNIST with images of 10 different clothing pieces. The performance of the proposed method is compared with the widely used Gaussian mixture model to validate its effectiveness.

The clustering performance is examined through different metrics and performance measure

used in our experiments are defined here. It is worthy to mention that accuracy computed through the ratio of correctly predicted instances to all the instances, i.e., $\frac{TP+TN}{TP+TN+FP+FN}$, is used to interpret the performance of clustering tasks. In this expression, TP, TN, FP and FN represents the true positives, true negatives, false positives and false negatives, respectively. In general, accuracy itself is not sufficient to ensure the effectiveness of the clustering approach. For this it is required to consider some other fundamental metrics, for instance, i) precision ($\frac{TP}{TP+FP}$), ii) sensitivity ($\frac{TP}{TP+FN}$), iii) specificity ($\frac{TN}{TN+FP}$) and iv) false positive rate ($\frac{FP}{FP+TN}$). Particularly, precision measures the ratio of accurately returned class labels to all the returned ones, sensitivity calculates the ratio of the correctly predicted true classes to the total actual true classes, specificity evaluates the proportion of the correctly predicted negative classes to all the actual negative classes and lastly false positive rate gives the ratio of inaccurate predicted positive classes to all actual negative classes. Moreover, for the case of imbalance in classes, it becomes necessary to examine the harmonic mean of precision and sensitivity, i.e., F1-Score ($(\frac{(\beta^2+1)\times sensitivity\times precision}{sensitivity+\beta\times precision}), \beta > 0$), the geometric mean of precision and sensitivity, i.e., G-mean1 ($\sqrt{precision \times sensitivity}$), the geometric mean of specificity and sensitivity, i.e., G-mean2 ($\sqrt{specificity \times sensitivity}$) and Mathew's correlation coefficient (MCC) for measuring quality of classification, i.e., ($\frac{TP.TN-FP.FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$) [106–109]. Primarily, these measures are used for classification task but they are equally valid to demonstrate the clustering performance.

2.3.1 TSP dataset

In this experiment our objective is to check how our model performs, when it comes to clustering between male and female voices. For this we use the TSP dataset [91], which is composed of speech files contributed by 22 speakers among which 11 are male and 11 are female. Each speaker contributes with 60 speech utterances. Hence, we have 660 samples from each class, contributing 1320 samples overall. This dataset has momentary pauses between the speeches which makes it important to removing these pauses before feature extraction. We do this by using voice activity detection (VAD) that makes sure that unnecessary data are not used for training the model. As a next step we extract MFCC feature descriptors and create a bag of audio words representation of the data. We compare the results of clustering using our model with GMM. Table 2.1 clearly shows the pre-eminence of our model compared to GMM. The confusion matrices are given in first row of Fig. (2.3), where first matrix represents the results when clustering was performed with GMM and second matrix indicates when clustering was performed with BGMM.

Table 2.1: Performance on TSP data for male and female data categorization based on different metrics

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	72.65	76.67	68.64	70.97	31.36	73.71	45.45	73.76	72.54
GMM	70.38	73.33	67.42	69.24	32.58	71.23	40.83	71.26	70.32

Table 2.2: Performance of Free Spoken Digit categorization based on different metrics (2 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	88.25	78.00	98.50	88.11	01.50	86.91	78.16	87.48	87.65
GMM	86.00	81.00	91.00	90.00	09.00	85.26	72.36	85.38	85.85

2.3.2 Free Spoken Digit Dataset

Recognizing spoken digits has been an important task in a number of voice recognition based security applications. The dataset we used for this experiment is the Free Spoken Digit Dataset (FSDD) [92]. The dataset consists of 2000 .wav recordings collected from 4 speakers. Each speaker contributes 50 recordings to each digit. All the recordings are at 8kHz frequency and are trimmed at both the ends. We extract Mel Frequency Cepstral Coefficients (MFCC) from the recordings. MFCC is the Cepstral representation of the recordings which is a better approximation of the response of human auditory system when compared to other linearly space frequency band representations. This method outputs multiple feature descriptors for a single audio file. We use a bag of audio words histogram of features model inspired from the bag of visual words approach on the feature descriptors thus obtained. This data acts as input to our model. We evaluate the performance of our model for different number of clusters against GMM. Tables 2.2, 2.3, 2.4, 2.5 and 2.6 clearly show that our model performs better than the standard GMM model.

2.3.3 MNIST Dataset for Hand Written Digits

We randomly sampled 1000 images from each category of the MNIST database of handwritten digits making the dataset size of 10000. Sample images from MNIST dataset are presented in Fig. (2.7). For pre-processing steps, as the representation of images is one of the crucial aspects

Table 2.3: Performance of Free Spoken Digit categorization based on different metrics (3 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	85.33	85.33	92.67	86.20	07.33	85.30	78.51	85.77	88.92
GMM	83.17	83.17	91.58	84.22	08.42	83.12	75.37	83.69	87.27

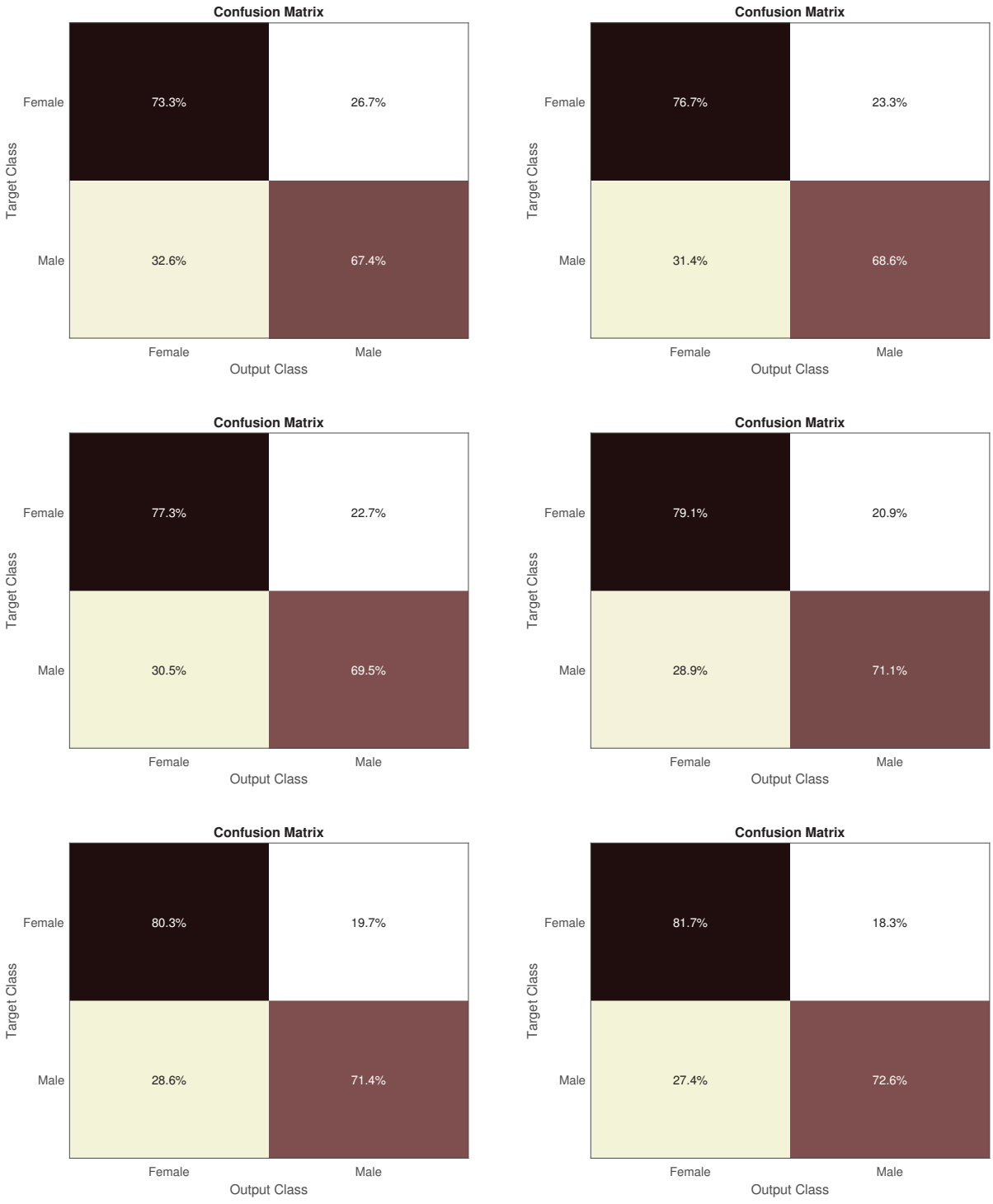


Figure 2.3: TSP Dataset with BGMM applied for code-book generation (compared with K-Means and GMM) and in learning data after feature extraction (compared with GMM)

Table 2.4: Performance of Free Spoken Digit categorization based on different metrics (4 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	84.25	84.25	94.75	84.49	05.25	84.34	79.11	84.37	89.35
GMM	82.67	82.64	94.22	84.18	05.78	82.83	77.58	83.41	88.24

Table 2.5: Performance of Free Spoken Digit categorization based on different metrics (5 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	83.90	83.90	95.97	84.76	04.02	84.02	80.23	84.33	89.73
GMM	81.10	81.10	95.27	83.27	04.73	80.89	77.22	82.18	87.90

Table 2.6: Performance of Free Spoken Digit categorization based on different metrics (10 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	71.93	71.95	96.88	78.96	03.12	73.63	71.67	75.38	83.49
GMM	68.60	68.60	96.51	75.62	03.49	69.80	67.76	72.03	81.37

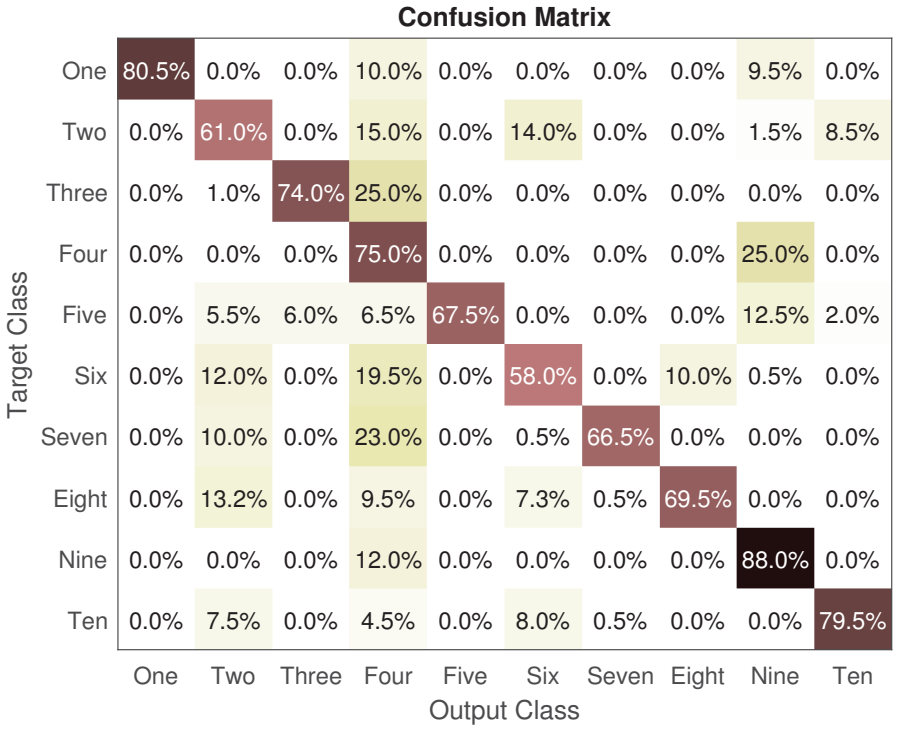


Figure 2.4: Spoken Digit Dataset with BGMM (K-Means in code-book generation and BGMM at second stage)

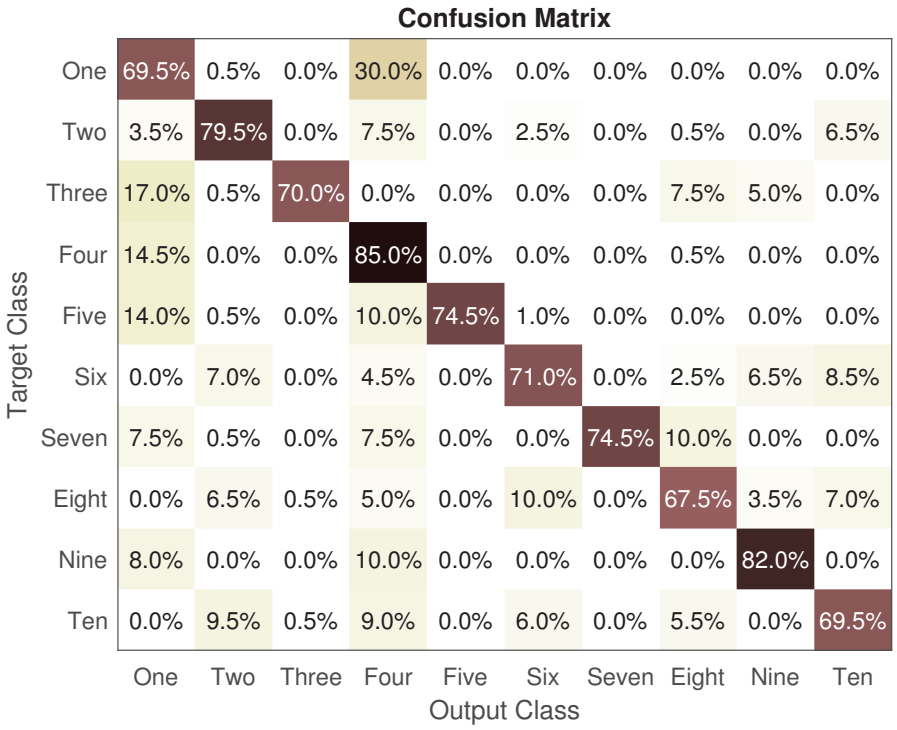


Figure 2.5: Spoken Digit Dataset with BGMM (BGMM in code-book generation and GMM at second stage)

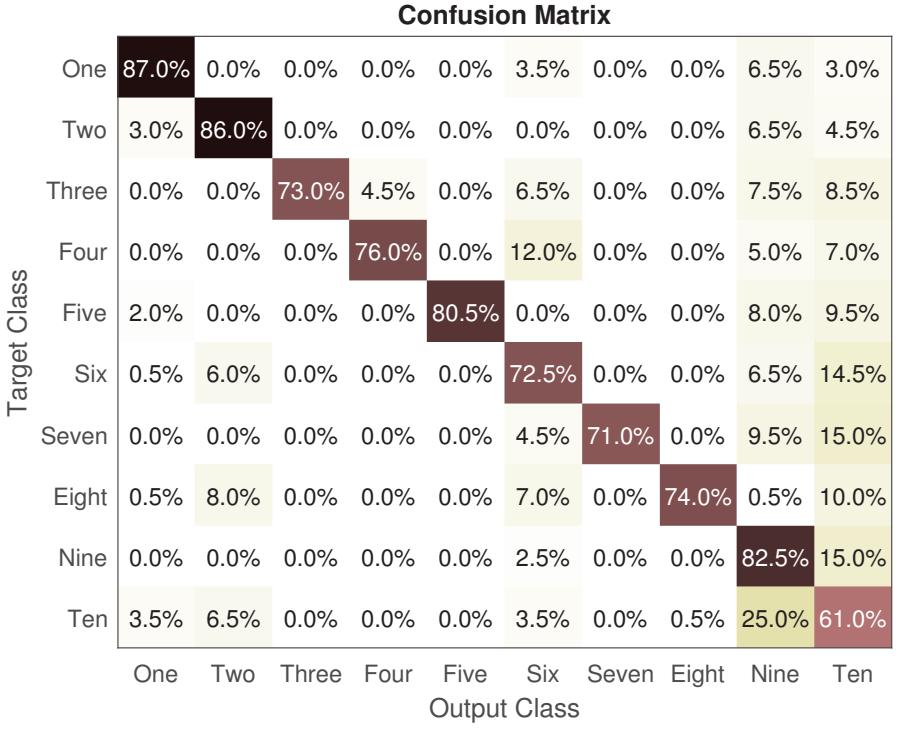


Figure 2.6: Spoken Digit Dataset with BGMM (BGMM at both stages)

Table 2.7: Performance of MNIST data categorization based on different metrics (2 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	92.95	87.00	98.90	98.75	01.10	92.50	86.51	92.69	92.76
GMM	91.40	90.50	92.30	92.16	07.70	91.32	82.81	91.33	91.40

Table 2.8: Performance of MNIST data categorization based on different metrics (3 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	91.20	91.20	95.60	92.02	04.40	91.25	87.25	91.61	93.37
GMM	89.60	89.60	94.80	90.07	05.20	89.52	84.68	89.83	92.16

of the test, we extracted scale-invariant feature transform (SIFT). All the 128D descriptors are assembled to a collection of features and K-means is then used for clustering the corpus to build our Bag of visual words (BoVW), in which each visual word is represented by a centroid. For this experiment, we tested our model's clustering performance with 2, 3, 4, and 5 classes randomly selected from the dataset. After that, we have also tested with all 10 categories to ensure the effectiveness for further large data applications. From the results in Tables 2.7, 2.8, 2.9, 2.10, and 2.11, the proposed method (BGMM) outperforms conventional Gaussian mixture (GMM) using most performance metrics. Furthermore, from confusion matrix presented in Fig. (2.8), most of the data points are accurately categorized despite the size of the dataset and number of classes, which validates the performance of the proposed method.

2.3.4 Fashion MNIST

Similar to previous application on MNIST dataset, we also tested on 10000 images randomly selected from the Fashion MNIST dataset with 1000 images in each category. Sample images from Fashion MNIST dataset are presented in Fig. (2.11). Likewise, the experiment is also conducted with 2, 3, 4, and 5 classes randomly selected from the dataset. Then, we also raised the difficulty up to all 10 classes to further challenge the proposed method. The feature extraction is performed by SIFT and building BoVW is done by gathering all the 128D feature vectors into a corpus

Table 2.9: Performance of MNIST data categorization based on different metrics (4 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	88.98	88.98	96.33	89.01	03.68	88.89	85.32	88.99	92.58
GMM	85.43	85.43	95.14	86.29	04.86	85.40	80.94	85.86	90.15

Table 2.10: Performance of MNIST data categorization based on different metrics (5 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	86.18	86.18	96.55	86.34	03.45	86.11	82.78	86.26	91.22
GMM	83.34	83.34	95.83	83.90	04.17	83.24	79.40	83.62	89.37

Table 2.11: Performance of MNIST data categorization based on different metrics (10 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	76.13	76.13	97.35	77.71	02.65	76.44	74.09	76.91	86.09
GMM	74.20	74.20	97.13	74.97	02.87	74.30	71.61	74.58	84.90

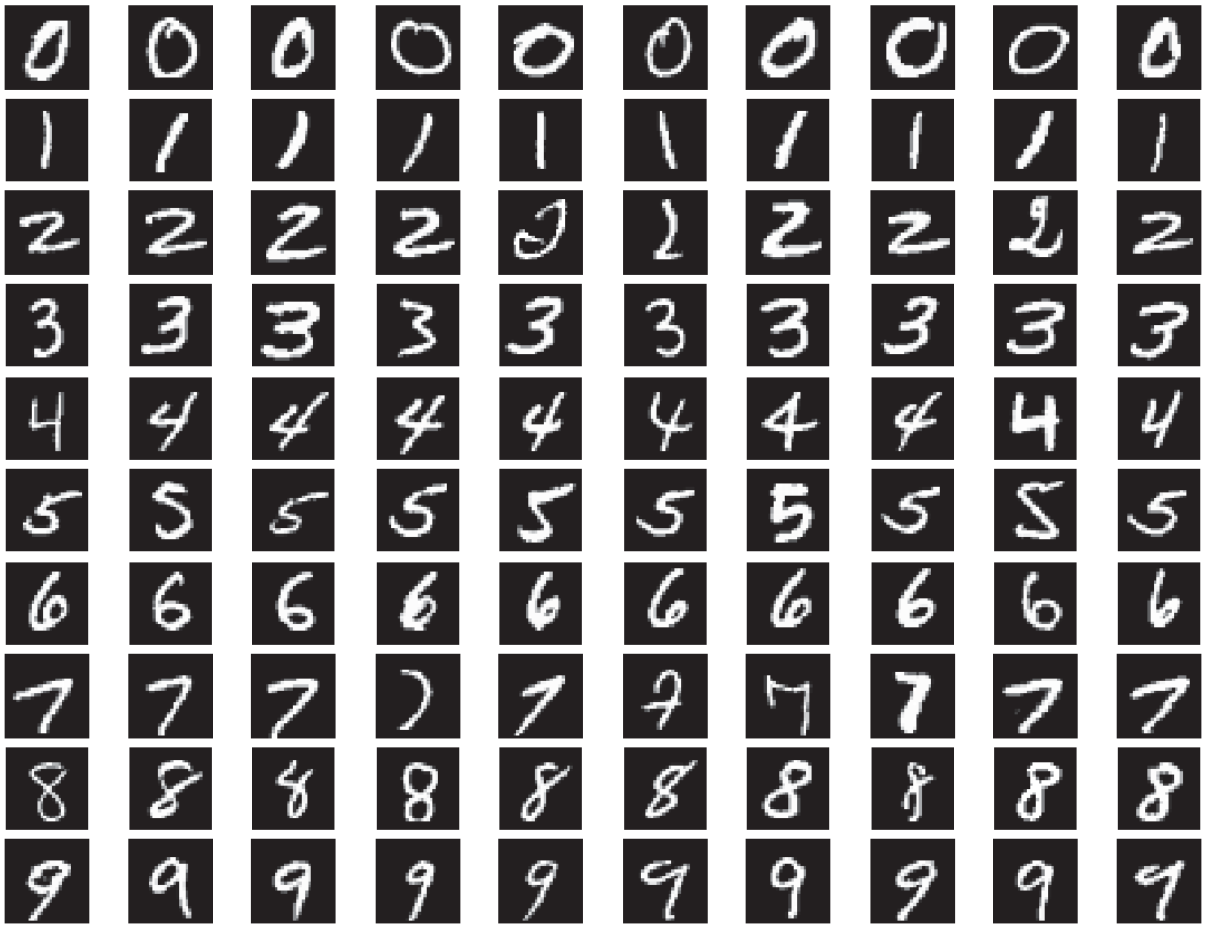


Figure 2.7: Samples of MNIST Dataset

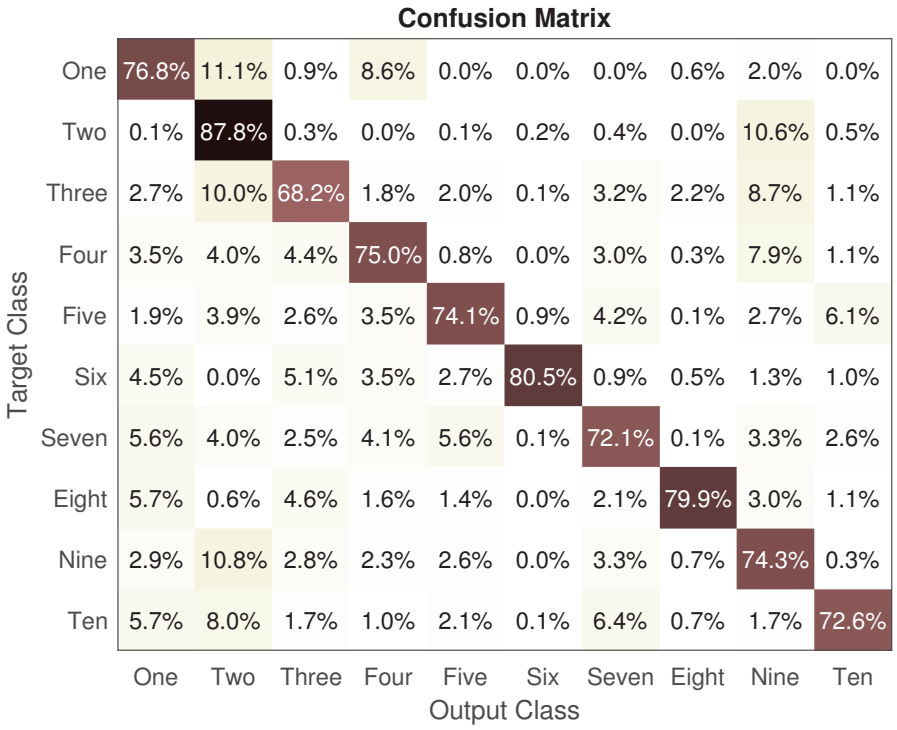


Figure 2.8: MNIST dataset with BGMM (K-Means in code-book generation and BGMM at second stage)

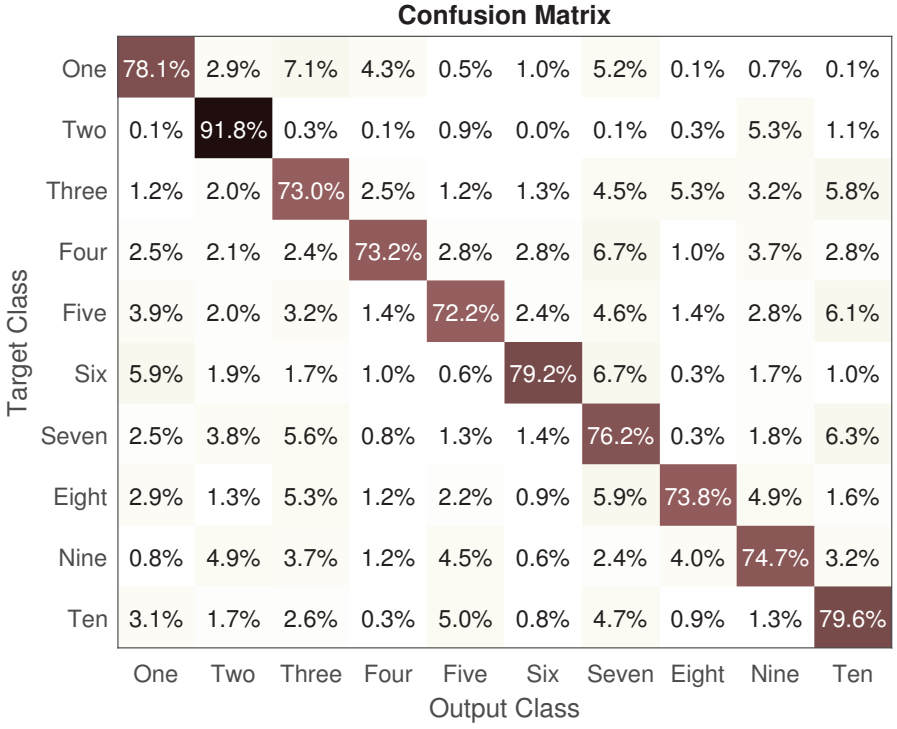


Figure 2.9: MNIST dataset with BGMM (BGMM in code-book generation and GMM at second stage)

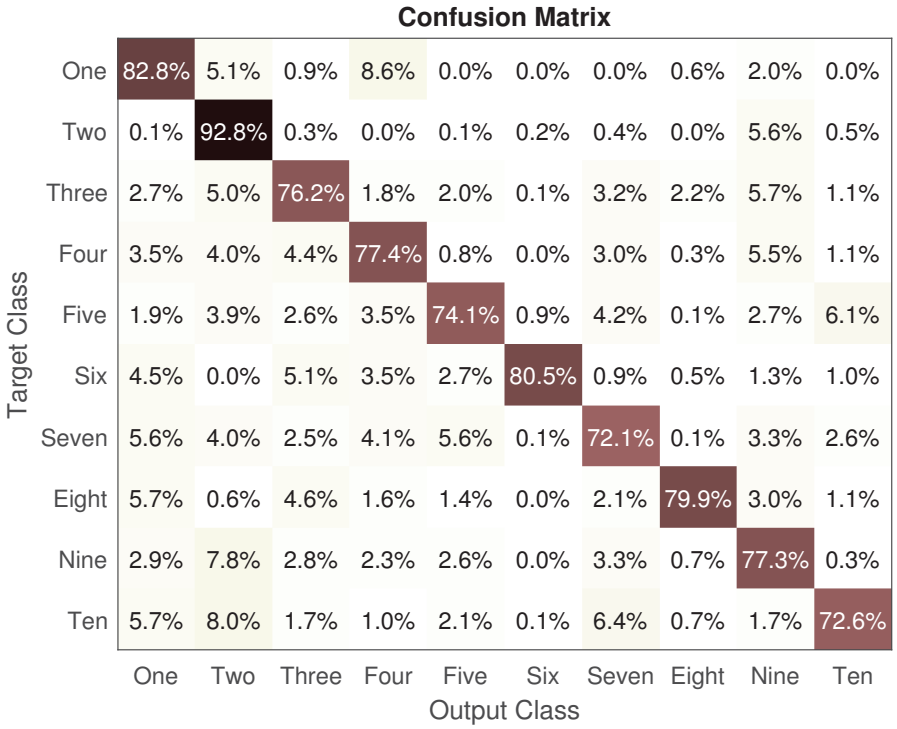


Figure 2.10: MNIST dataset with BGMM (BGMM at both stages)

Table 2.12: Performance of Fashion MNIST data categorization based on different metrics (2 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	91.90	92.70	91.10	91.24	08.90	91.96	83.81	91.96	91.89
GMM	89.75	87.00	92.50	92.06	07.50	89.46	79.62	89.49	89.70

and cluster them with K-Means. The outcomes of each case in Tables 2.12, 2.13, 2.14, 2.15 and 2.16 indicates the effectiveness of BGMM compared with the widely used GMM using most performance metrics. It is noteworthy that BGMM outperforms GMM when testing with 10 classes as most datapoints are accurately classified with minimum mis-classification as shown in confusion matrix given in Fig. (2.12).

Table 2.13: Performance of Fashion MNIST data categorization based on different metrics (3 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	88.73	88.73	94.36	88.74	05.63	88.73	83.10	88.73	91.50
GMM	85.10	85.10	92.55	85.18	07.45	85.13	77.68	85.14	88.74

Table 2.14: Performance of Fashion MNIST data categorization based on different metrics (4 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	87.30	87.30	95.76	87.50	04.23	87.24	83.15	87.40	91.43
GMM	84.47	84.47	94.82	84.798	05.17	84.47	79.44	84.63	89.50

Table 2.15: Performance of Fashion MNIST data categorization based on different metrics (5 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	86.34	86.34	96.58	86.42	03.41	86.34	82.96	86.38	91.31
GMM	84.26	84.26	96.06	84.42	03.93	84.16	80.37	84.34	89.96

2.3.5 Discussion on Application of BGMM for Speech and Image Data Clustering

As first step to apply BGMM, in data clustering, we choose two speech and two images datasets. In order to evaluate the performance of any clustering algorithm, it is needed to examine its performance by choosing data from different types. In our experiments on TSP speech data, BGMM is applied to categorize the male and female speakers and it has shown better performance as compared to GMM in a similar setting. TSP dataset is composed of only two classes and it is very important to examine the performance with datasets having higher number of classes and Spoken Digits dataset is selected for this task. With Spoken Digits dataset, we created several multi-cluster scenarios (2, 3, 4, 5 and 10) and it is observed that BGMM is always better in performance as compared to GMM. However, performance of clustering is decreased when we consider higher number of classes in the dataset for speech categorization which makes sense due to the increase in complexity caused in model learning with higher number of classes. We conducted our clustering experiments on images datasets with a similar multi-cluster scenario described for Spoken Digits, however images datasets (MNIST and Fashion MNIST) have larger size as compared to speech datasets. Experimental results indicates that BGMM has demonstrated its success for images categorization as compared to GMM in a similar experimental setting. However clustering performance starts decreasing slightly when we choose higher number of classes for an experiment.

Table 2.16: Performance of Fashion MNIST data categorization based on different metrics (10 Classes)

	Performance Metrics (%)								
Models	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	74.86	74.86	97.21	75.78	02.79	74.89	72.37	75.32	85.30
GMM	73.16	73.16	97.02	75.18	02.98	73.36	70.88	74.16	84.25

Figure 2.11: Samples of Fashion MNIST Dataset

2.4 Application of BGMM in Code Book Generation

Feature extraction and pre-processing is always considered to be very important in learning applications and representation of data in an effective manner can improve the performance of modeling capabilities of machine learning algorithms. In our experiments for speech data clustering, we have examined the performance of BGMM for different number of categories as compared to GMM in a similar setting. Feature extraction from speech signals is performed using MFCCs, which are further used to create the BoAW. In the creation of BoAW, K-means is used which is very standard in BoW applications for image, text and speech datasets. We propose to apply BGMM for the creation of code-book in a similar manner as it is being used to represent data with K-Means. This should improve the pre-processing of speech data and BoAW representation for clustering. We have conducted several experiments to see the effectiveness of BGMM in pre-processing and compared with GMM and K-Means for TSP and Spoken Digits dataset.

We extend our focus on pre-processing for images datasets as well which will improve the

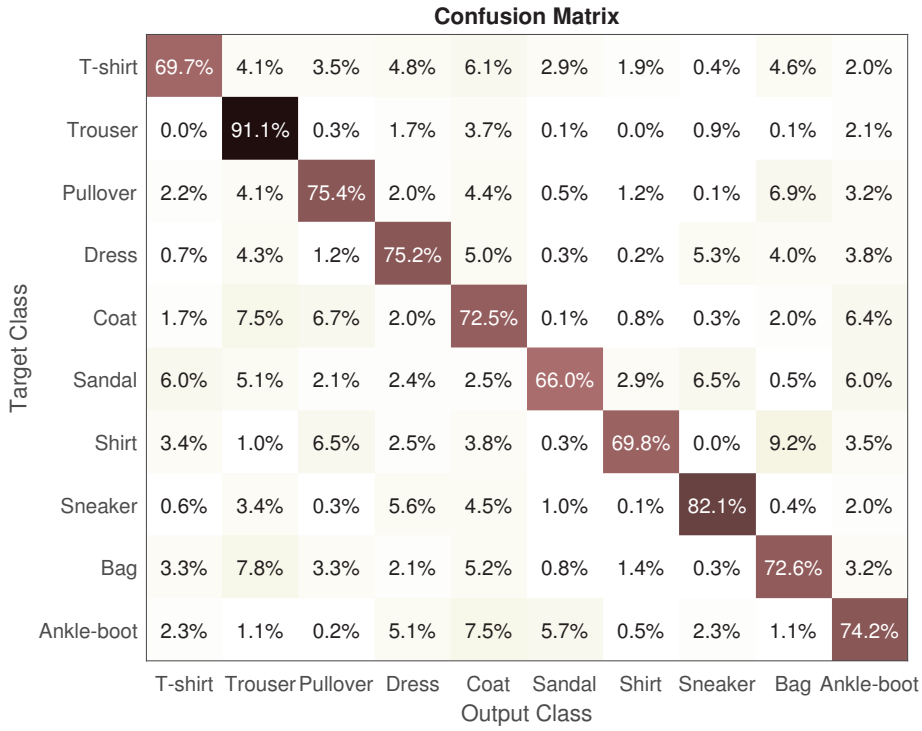


Figure 2.12: Fashion MNIST dataset with BGMM (K-Means in code-book generation and BGMM at second stage)

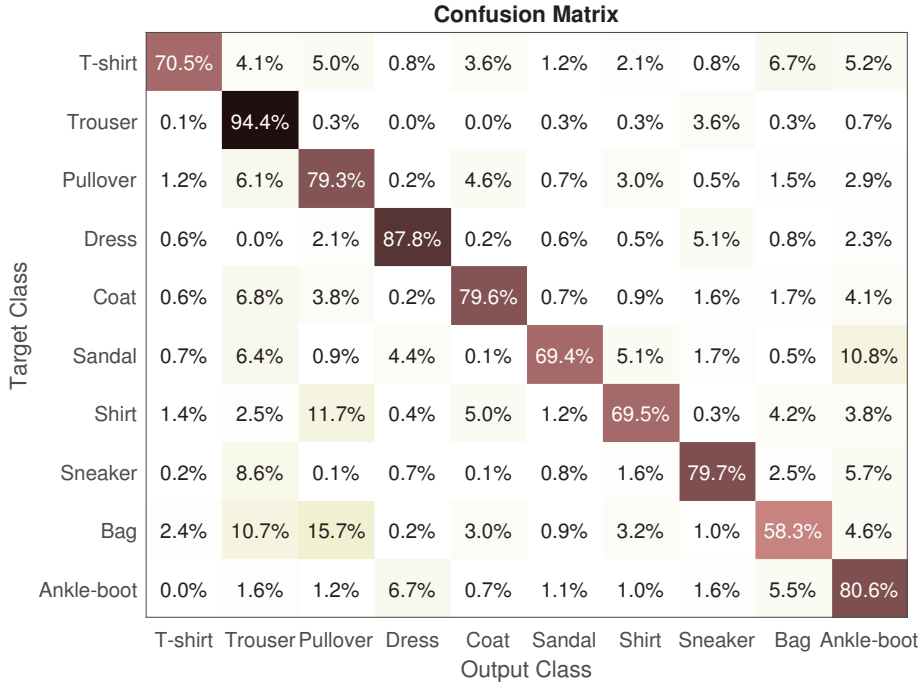


Figure 2.13: Fashion MNIST dataset with BGMM (BGMM in code-book generation and GMM at second stage)

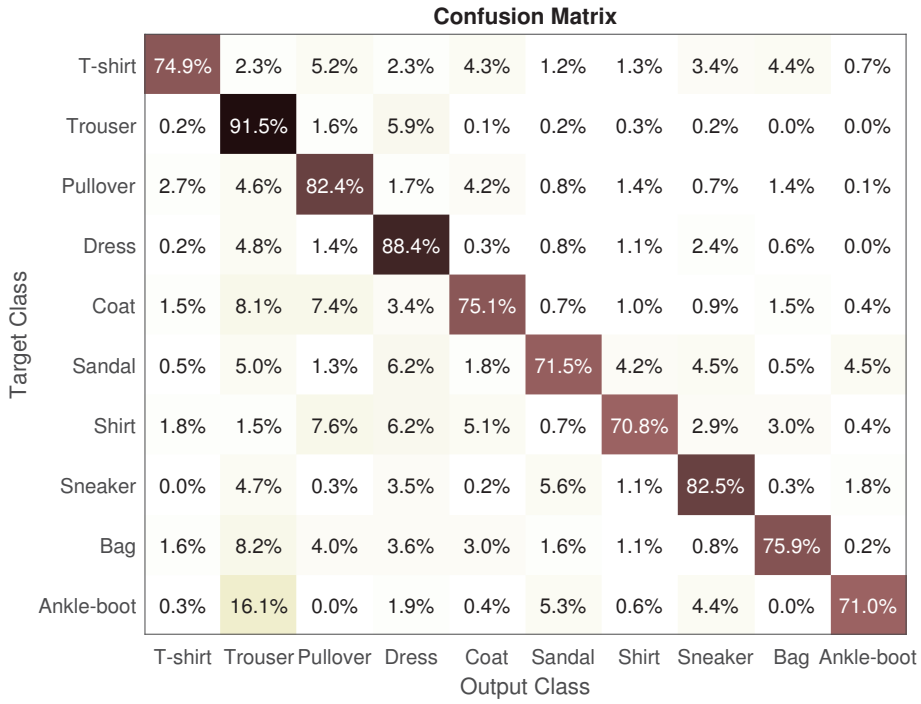


Figure 2.14: Fashion MNIST dataset with BGMM (BGMM at both stages)

data representation in learning applications. In computer vision analysis, the pre-processing step has always been considered among the most important aspects. Indeed, there are many methods to efficiently extract features of an image to convert it to a descriptor such as SIFT, which has been extensively applied to various applications [110–112]. After that, the code-book construction is usually done with K-Means, a vector quantization method which minimizes squared errors for clustering. We propose applying BGMM to the building step to enhance the robustness of the code book compared with GMM and K-means.

2.4.1 TSP Dataset

In this section, we test the performance of BGMM for code-book generation for speech. In Section 2.3.1, BGMM was applied in categorizing male and female speakers for TSP dataset. In our previous experiment, BOAW was chosen as methods for speech data representation where code-book was generated using MFCC features with K-Means. We extend the same experimental framework for testing the performance of BGMM in code-book generation and propose to apply BGMM in the pre-processing stage. The proposed framework in the pre-processing will be compared with GMM and K-Means. In our previous experiments, we examined the performance of BGMM in clustering after the code-book generation and compared with GMM. With use of BGMM in the

Table 2.17: Performance of BGMM in Code Book Generation using TSP Dataset

Models in BOW	Performance Metrics (%) with second stage clustering using GMM								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	75.08	79.09	71.06	73.21	28.94	76.04	50.31	76.09	74.97
GMM	73.41	77.27	69.55	71.73	30.45	74.40	46.96	74.45	73.31
K-Means	70.38	73.33	67.42	69.24	32.58	71.23	40.83	71.26	70.32

Table 2.18: Performance of BGMM in Code Book Generation using TSP Dataset

Models in BOW	Performance Metrics (%) with second stage clustering using BGMM								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	77.12	81.67	72.58	74.86	27.42	78.12	54.47	78.19	76.99
GMM	75.83	80.30	71.36	73.71	28.64	76.87	51.87	76.94	75.70
K-Means	72.65	76.67	68.64	70.97	31.36	73.71	45.45	73.76	72.54

pre-processing stage, the performance of BGMM has to be evaluated in two steps of clustering. The first step is to generate the code-book and the second step is the clustering of data. In the first step we use BGMM, GMM and K-Means for clustering whereas in the second step we use BGMM and GMM. Hence, there are 6 combinations and the results for all combinations are shown in Tables 2.17 and 2.18. In the first case, we applied the BGMM in the pre-processing stage and compared it with GMM and K-Means whereas second stage in this scenario is modeled with GMM and best performance is achieved with BGMM. In the second scenario, we choose the pre-processing with BGMM (and both comparison scenarios using GMM and K-Means) and replace the second stage of clustering from GMM to BGMM to see the effect of proposed model in both stages of clustering. It is observed that best performance for clustering is achieved when BGMM is applied in both stages. The results from Tables 2.17 and 2.18 explain that BGMM has effectively demonstrated its viability in clustering speech feature vectors to improve the representation of speech data. It also explains that BGMM has improved the clustering performance when it is also used in the second stage within the same pipeline for speech data clustering. The clustering performance indicated in Tables 2.17 and 2.18 also demonstrates that False Positive Rate (FPR) is low with BGMM as compared to the other cases.

2.4.2 Free Spoken Digit Dataset

The experiments for code-book generation on TSP dataset are further extended with Spoken Digits and BGMM is applied in a similar manner. The performance of the proposed approach for code-book generation is compared for both stages which make 6 possible comparison scenarios as described in Tables 2.19 and 2.20. The clustering performance is evident from both tables and

Table 2.19: Performance of BGMM in Code Book Generation using Spoken Digits Dataset

Models in BOW	Performance Metrics (%) with second stage clustering using GMM								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	74.30	74.30	97.14	78.77	02.86	75.31	73.22	76.50	84.96
GMM	71.35	71.35	96.82	76.46	03.18	72.43	70.15	73.86	83.11
K-Means	68.60	68.60	96.51	75.62	03.49	69.80	67.76	72.03	81.37

Table 2.20: Performance of BGMM in Code Book Generation using Spoken Digits Dataset

Models in BOW	Performance Metrics (%) with second stage clustering using BGMM								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	76.35	76.35	97.37	82.30	02.63	77.92	76.16	79.27	86.22
GMM	74.10	74.10	97.12	77.21	02.88	74.95	72.50	75.64	84.83
K-Means	71.93	71.95	96.88	78.96	03.12	73.63	71.67	75.38	83.49

different performance measures are computed for each combination. The results clearly show that BGMM has demonstrated its success in code-book generation as compared to GMM and K-Means and the performance is further improved when BGMM is applied on both stages of clustering pipeline. It is also observed that in all our experiments FPR for our BGMM is always low. This proves the efficiency of our proposed model.

2.4.3 MNIST Dataset

In the first stage, unlike the last experiment where the BoVW is constructed based on solely K-means, we applied also GMM, and BGMM to test the efficiency of the proposed method compared with other widely used models in the construction of BoVW. Then, the proposed method and GMM are applied to test with all three scenarios in the second stage and their performances are shown in Tables 2.21 and 2.22, respectively. It is clear that the performance of using BGMM in creating BoVW is significantly better than conventional GMM and K-Means. Furthermore, from Fig. 2.9, which presents the best performance of Table 2.21, it is observed that applying BGMM only in first stage has significantly improved the clustering performance. From Fig. 2.10, which indicates the best performance of Table 2.22, it is evident that when BGMM is applied in both stages, most observations have been accurately clustered with minimum mis-classification which verifies the capability of the proposed method.

2.4.4 Fashion MNIST Dataset

Encouraged by the efficiency of our model in previous experiment, we extended the work with code book generation using Fashion MNIST dataset. Results in Tables 2.23 and 2.24 along with

Table 2.21: Performance of BGMM in Code Book Generation using MNIST data

Models in BOVW	Performance Metrics (%) with second stage clustering using GMM								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	77.18	77.18	97.46	77.72	02.54	77.24	74.84	77.45	86.73
GMM	76.24	76.24	97.36	76.80	02.64	76.29	73.79	76.52	86.16
K-Means	74.20	74.20	97.13	74.97	02.87	74.30	71.61	74.58	84.90

Table 2.22: Performance of BGMM in Code Book Generation using MNIST data

Models in BOVW	Performance Metrics (%) with second stage clustering using BGMM								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	78.57	78.57	97.62	79.66	02.38	78.71	76.58	79.11	87.58
GMM	77.61	77.61	97.51	78.84	02.49	77.83	75.58	78.22	86.99
K-Means	76.13	76.13	97.35	77.71	02.65	76.44	74.09	76.91	86.09

confusion matrices presented in Figs. (2.13 & 2.14) clearly describe that the proposed method outperforms GMM and K-means in both the construction of BoVW and clustering with extracted features.

2.5 Model Selection with Minimum Message Length (MML) Criterion

In order to estimate the number of components of mixture model, different model selection methods have been discussed in [39, 71, 72, 74, 75, 82–84]. We have proposed a deterministic approach using MML for model selection in BGMM. By applying MML, the optimal number of classes is obtained by minimizing the following equation [113, 114]:

$$MessLen(K) \simeq -\log(p(\Theta_K)) - \mathscr{L}(\Theta_K, Z, \mathscr{X}) + \frac{1}{2}\log|F(\Theta_K)| + \frac{N_p}{2}(1+\log(k_{N_p})) \qquad (2.22)$$

where N_p is number of free parameters, Θ_K is set of parameters when mixture contains K components, $p(\Theta_K)$ is prior probability and $|F(\Theta_K)|$ is determinant of the Fisher information matrix of

Table 2.23: Performance of BGMM in Code Book Generation using Fashion MNIST data

Models in BOVW	Performance Metrics (%) with second stage clustering using GMM								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	76.91	76.91	97.43	78.33	02.57	76.89	74.79	77.62	86.57
GMM	75.34	75.34	97.26	77.54	02.74	75.16	73.22	76.43	85.60
K-Means	73.16	73.16	97.02	75.18	02.98	73.36	70.88	74.16	84.25

Table 2.24: Performance of BGMM in Code Book Generation using Fashion MNIST data

Models in BOVW	Performance Metrics (%) with second stage clustering using BGMM								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BGMM	78.40	78.40	97.60	80.02	02.40	78.51	76.56	79.21	87.47
GMM	77.11	77.11	97.46	79.59	02.54	77.35	75.44	78.34	86.69
K-Means	74.86	74.86	97.21	75.78	02.79	74.89	72.37	75.32	85.30

minus the log-likelihood of mixture model. k_{N_p} is optimal quantization lattice constant $\mathbb{R}^{N_p}$ [115] and its written as $k_1 = 1/12 \simeq 0.83$ for $N_p = 1$. As N_p grows, k_{N_p} will become an asymptotic value as $1/2\pi e \simeq 0.05855$ and it is noted that k_{N_p} does not vary a lot and it can be approximated by $1/12$ [116]. The estimation of the number of classes is carried out by finding the minimum with respect to Θ of the message length [32, 47, 116]. The derivation of $p(\Theta_K)$ and $|F(\Theta_K)|$ is given as follows.

2.5.1 Derivation of the prior $p(\Theta)$

In the model selection, a prior $p(\Theta)$ is specified to express the lack of knowledge about the parameters of mixture model. It is logic to assume that different components of mixture have independent parameters, since having information about the parameters in one class does not provide any information about the parameters of another class. Thus, it is assumed that parameters of a mixture model are mutually independent, which cede the following prior distribution over the parameters π, μ and Σ:

$$p(\Theta) = p(\pi)p(\mu)p(\Sigma) \tag{2.23}$$

where $\pi = (p_1, ..., p_K)$. Each of these densities in the prior distribution is defined separately. Beginning with $p(\pi)$, we know that vector π is defined on the simplex as $\{(p_1, ..., p_K) : \sum_{j=1}^{K} p_j = 1\}$. In this case, a natural choice as a prior for vector π is Dirichlet distribution, which is defined as:

$$p(\pi) = \frac{\Gamma(\sum_{j=1}^{K} \eta_j)}{\sum_{j=1}^{K} \Gamma(\eta_j)} \sum_{j=1}^{K} {p_j}^{\eta_j - 1} \tag{2.24}$$

where $(\eta_1, ..., \eta_K)$ is the parameters vector of Dirichlet distribution. By choosing, $\eta_1 = 1, ..., \eta_K = 1$, we get a uniform prior over the space $p_1 + ... + p_K = 1$, which is represented as:

$$p(\pi) = (K-1)! \tag{2.25}$$

For the prior distributions of parameters μ and σ, a methodology described in [39, 82, 117, 118] is adopted. A flat prior is normally considered for μ and a conjugate inverted Wishart prior is adopted

for covariance matrix Σ. In [39], it is described that dependent and independent prior distributions will be equivalent in this case and in [118], a joint prior distribution for μ and Σ is proposed as follows:

$$p(\mu,\Sigma) \propto \prod_{j=1}^{K} |\Sigma|^{-\frac{D+1}{2}} \tag{2.26}$$

Finally, by replacing the priors of parameters in Eq. (2.23) by Eqs. (2.25 & 2.26), we get:

$$p(\Theta) \propto ((K-1)!)|\Sigma|^{-K\frac{D+1}{2}} \tag{2.27}$$

2.5.2 Derivation of the Fisher information matrix $|F(\Theta)|$

Fisher information matrix is defined as expected value of Hessian matrix. It is difficult to reproduce the expected Fisher Information matrix because it leads to a complicated analytical form of MML. Therefore, Hessian matrix can be approximated by complete Fisher information matrix, where its determinant is computed by taking the product of determinant of Fisher information matrix of each mixture component which is further multiplied by determinant of Fisher information matrix for π as follows:

$$|F(\Theta)| = |F(\pi)| \prod_{j=1}^{K} \left|F(\vec{\mu}_j)\right| \left|F(\Sigma_j)\right| \tag{2.28}$$

where $\left|F(\vec{\mu}_j)\right|$ and $\left|F(\Sigma_j)\right|$ are the determinants of Fisher information matrices with respect to the mean and covariance, respectively, for BGMM, corresponding to jth mixture component. $|F(\pi)|$ is Fisher information matrix with respect to mixing parameter, where it is required to satisfy the constraint $\sum_{j=1}^{K} p_j = 1$. We consider the generalized Bernoulli process with a sequence of trials with K possible outcomes labeling the first cluster, second cluster and then continue until Kth cluster. The number of trials for each component in this case, can be represented by multinomial distribution of mixing parameters of $p_1, p_2, ..., p_K$ [32, 47]. The determinant of Fisher information matrix for mixing parameters can be given as follows:

$$|F(\pi)| = \frac{N^{K-1}}{\sum_{j=1}^{K} p_j} \tag{2.29}$$

where N is number of observations. For the computation of $\left|F(\vec{\mu}_j)\right|$ and $\left|F(\Sigma_j)\right|$, we consider the data in jth cluster, after classifying all data $\mathscr{X}$ using maximum a posteriori probability defined in Eq. (2.8), which is represented as $\mathscr{X}_j = (\vec{X}_l, ..., \vec{X}_{l+n_j-1})$. Here n_j is number of observations belonging to the jth cluster. Here notation of data in each class can be simplified by the choice of jth class without loss of generality. The Hessian matrices with respect to parameters $\vec{\mu}_j$ and Σ_j for

each jth component are represented as follows:

$$F(\vec{\mu}_j)_{k_1,k_2} = \frac{\partial^2 \mathscr{L}(\Theta, Z, \mathscr{X}_j)}{\partial \mu_{jk_1} \partial \mu_{jk_2}} \tag{2.30}$$

$$F(\Sigma_j) = \frac{\partial^2 \mathscr{L}(\Theta, Z, \mathscr{X}_j)}{\partial \Sigma_{jk_1} \partial \Sigma_{jk_2}} \tag{2.31}$$

where $(k_1, k_2) \in (1, ..., D)$. The computations of derivatives for Eqs. (2.30 & 2.31) is given in Appendix A.6 and we have used only diagonal values from the Hessian matrices for computation of Fisher information. The computation of $\left|F(\vec{\mu}_j)\right|$ for jth class of data is given below:

$$\begin{aligned} \left|F(\vec{\mu}_j)\right| = \prod_{d=1}^{D} \sum_{i=l}^{l+n_j-1} \Bigg| \Sigma_j^{-1} \Bigg\{ -1 + \frac{\Sigma_j^{-1} \left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)(\vec{\mathrm{u}} - \vec{\mu}_j) d\mathrm{u} \right)^2}{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right)^2} \\ - \frac{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) \left((\vec{\mathrm{u}} - \vec{\mu}_j) \Sigma_j^{-1} (\vec{\mathrm{u}} - \vec{\mu}_j)^T - 1 \right) d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \Bigg\} \Bigg| \end{aligned} \tag{2.32}$$

By considering the approximations for estimation of mean parameters as mentioned in Section 2.2, $\left|F(\vec{\mu}_j)\right|$ can be written as follows:

$$\begin{aligned} \left|F(\vec{\mu}_j)\right| = \prod_{d=1}^{D} \sum_{i=1}^{N} \Bigg| \Sigma_j^{-1} \Bigg\{ -1 + \frac{\Sigma_j^{-1} \left(\sum_{m=1}^{M} (\mathrm{s}_{m_j} - \vec{\mu}_j) \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j) \right)^2}{\left(\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j) \right)^2} \\ - \frac{\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j) \left((\mathrm{s}_{m_j} - \vec{\mu}_j) \Sigma_j^{-1} (\mathrm{s}_{m_j} - \vec{\mu}_j)^T - 1 \right) d\mathrm{u}}{\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j)} \Bigg\} \Bigg| \end{aligned} \tag{2.33}$$

The computation of $\left|F(\Sigma_j)\right|$ for jth class is given below:

$$\begin{aligned} \left|F(\Sigma_j)\right| = \sum_{i=l}^{l+n_j-1} \Bigg| \Bigg\{ \frac{1}{2} \Sigma_j^{-2} - (\vec{X} - \vec{\mu}_j) {\Sigma_j}^{-3} (\vec{X} - \vec{\mu}_j)^T \\ + \frac{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) \{ -\frac{1}{2} \Sigma_j^{-1} + \frac{1}{2} (\vec{\mathrm{u}} - \vec{\mu}_j) \Sigma_j^{-2} (\vec{\mathrm{u}} - \vec{\mu}_j) \} d\mathrm{u} \right)^2}{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right)^2} \\ - \frac{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) \left[\left(-\frac{1}{2} \Sigma_j^{-1} + \frac{1}{2} (\vec{\mathrm{u}} - \vec{\mu}_j) \Sigma_j^{-2} (\vec{\mathrm{u}} - \vec{\mu}_j)^T \right)^2 \right] d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \\ - \frac{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) \left[\left(\frac{1}{2} \Sigma_j^{-2} + (\vec{\mathrm{u}} - \vec{\mu}_j) \Sigma_j^{-3} (\vec{\mathrm{u}} - \vec{\mu}_j)^T \right) \right] d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \Bigg\} \Bigg| \end{aligned} \tag{2.34}$$

By considering the approximations for estimation of co-variance parameters as mentioned in Section 2.2, $\left|F(\Sigma_j)\right|$ can be written as follows:

$$\left|F(\Sigma_j)\right| = \sum_{i=l}^{l+n_j-1} \left| \left\{ \frac{1}{2}\Sigma_j^{-2} - (\vec{X}-\vec{\mu}_j){\Sigma_j}^{-3}(\vec{X}-\vec{\mu}_j)^T \right. \right. \tag{2.35}$$
$$+ \frac{\left(\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j)\{-\frac{1}{2}\Sigma_j^{-1} + \frac{1}{2}(\mathrm{s}_{m_j}-\vec{\mu}_j)\Sigma_j^{-2}(\mathrm{s}_{m_j}-\vec{\mu}_j)\}\right)^2}{\left(\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j)\right)^2}$$
$$- \frac{\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j)\left[\left(-\frac{1}{2}\Sigma_j^{-1} + \frac{1}{2}(\mathrm{s}_{m_j}-\vec{\mu}_j)\Sigma_j^{-2}(\mathrm{s}_{m_j}-\vec{\mu}_j)^T\right)^2\right]}{\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j)}$$
$$\left. \left. - \frac{\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j)\left[\left(\frac{1}{2}\Sigma_j^{-2} + (\mathrm{s}_{m_j}-\vec{\mu}_j)\Sigma_j^{-3}(\mathrm{s}_{m_j}-\vec{\mu}_j)^T\right)\right]}{\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_j}|\Omega_j)} \right\} \right|$$

Model selection algorithm also serves as a complete clustering solution because it provides the optimal number of mixture components which helps to estimate the optimal parameters learned through EM. The complete learning of model selection with MML in an EM algorithm is given in Algorithm 2.

2.6 Experiments on model selection and results

Model selection using MML is applied to different data clustering applications in order to validate the performance of proposed approach. As first step in our experiments, it is applied to 10 different medical datasets which are used to model the behavior of different human conditions and prediction based on this model. In unsupervised learning, finding the correct number of clusters is very important in correctly categorizing the data. Experiments and results on model selection are given in Section 2.6.2. In all our experiments for model selection, MML is compared with seven other model selection criteria to examine and validate its performance. Details of several model selection criteria used for a comparison with MML in all our experiments are given in Section 2.6.1. Our next experiments for model selection are conducted on the datasets used in clustering for speech and image processing datasets with different clustering scenarios and they are discussed in details in Sections 2.6.1-2.6.6 along with comparisons with other model selection criteria.

2.6.1 Comparison with other model selection criteria

The proposed model selection via MML approach is compared with different deterministic model selection criteria given in literature. The comparison methods for model selection include MDL [74], AIC [119], Bayesian inference criterion (BIC) [73], Consistent AIC (CAIC) [120], Mixture

Algorithm 2 Complete Model Learning with BGMM and Model Selection using MML

Input:Dataset $\mathscr{X} = \{\vec{X}_1, \ldots, \vec{X}_N\}$, t_{min} and K_{max}.
Output: K^* and Θ_{K^*}.
Step 1: for $M = 1 : K_{max}$ **do**{
{Initialization}:
  K-Means Algorithm (Computation of $\vec{\mu}_1, \ldots, \vec{\mu}_K$ & cluster assignment)
    for all $1 \leq j \leq K$ do
      Computation of p_j
      Computation of Σ_j
    end for
{Expectation Maximization}:
while relative change in log-likelihood $\geq t_{min}$ do
  {[**E Step**]}:
    for all $1 \leq j \leq K$ do
      Compute $p(j|\vec{X}_i)$ for $i = 1, \ldots, N$. using Eq. (2.8).
    end for
  {[**M step**]}:
    for all $1 \leq j \leq K$ do
      Update the mixing parameter $\hat{p}_j$ using Eq. (2.12).
      Update the mean $\hat{\vec{\mu}}_j$ using Eq. (2.17).
      Update Co-variance matrix $\hat{\Sigma}_j$ using Eq. (2.21).
    end for
end while
Calculate the associated message length using Eq. (5.71).
}**end for**
Step 2: Select the Model K^* with smallest message length

MDL (MMDL) [121], MML_{like} [30], LEC [16, 39]. In general, any deterministic model selection criterion can be written in the following form:

$$C(\hat{\Theta}(K), K) = -\mathscr{L}(\Theta_K, Z, \mathscr{X}) + f(K) \tag{2.36}$$

where $f(K)$ is an increasing function which penalizes higher values of K and optimal number of components in a mixture is determined as follows:

$$\hat{K} = arg\ min\{C(\hat{\Theta}(K), K), K = K_{min}, ..., K_{max}\} \tag{2.37}$$

Although model selection criteria have this common point, they are different conceptually and they are described by the following equations:

$$MDL(K) = -\mathscr{L}(\Theta_K, Z, \mathscr{X}) + \frac{N_p}{2}\log(N) \tag{2.38}$$

where N_p is number of free mixture parameters and computed as $K*((D*D-D)/2+2D+1)-1$ in our case.

$$AIC(K) = -\mathscr{L}(\Theta_K, Z, \mathscr{X}) + \frac{N_p}{2} \tag{2.39}$$

$$BIC(K) = -2\mathscr{L}(\Theta_K, Z, \mathscr{X}) + N_p \log(N) \tag{2.40}$$

$$CAIC(K) = -2\mathscr{L}(\Theta_K, Z, \mathscr{X}) + N_p(1+\log(N)) \tag{2.41}$$

$$MMDL(K) = -\mathscr{L}(\Theta_K, Z, \mathscr{X}) + \frac{1}{2}N_p \log(N) + \frac{c}{2}\sum_{j=1}^{K}\log(p_j) \tag{2.42}$$

where c is the number of free parameters for each mixture component and computed as $(D*D-D)/2+2D+1$ in our case.

$$MML_{Like}(K) = -\mathscr{L}(\Theta_K, Z, \mathscr{X}) + \frac{K}{2}\log\left(\frac{N}{12}\right) + \frac{c}{2}\sum_{j=1}^{K}\log\left(N\frac{p_j}{12}\right) + \frac{N_p}{2} \tag{2.43}$$

For model selection through LEC, prior probability and determinant of Fisher information matrix computed for MML is adopted in the following equation.

$$LEC(K) = \mathscr{L}(\Theta_K, Z, \mathscr{X}) - \log(P(\Theta_K)) - \frac{1}{2}N_p \log(2\pi) + \frac{1}{2}\log(|F(\Theta_K)|) \tag{2.44}$$

2.6.2 Model Selection on Medical Datasets

In this section, performance of model selection via MML is validated through 10 different real medical datasets taken from UCI repository [122–130]. The results of MML are compared with other model selection criteria to examine the effectiveness of our proposed approach and on the basis of these results, important conclusions are made. In order to perform the clustering on these datasets, class labels were removed. Each dataset is unique and hence tested to examine the performance and demonstrate the viability of the MML which eventually improves the whole clustering process. Table 2.25 summarizes the results of each method against each dataset. Model selection performance is also shown graphically in Figs (2.15-2.24) which reflects the performance of each method. A detailed discussion on model selection performance for each datasets is described as follows.

2.6.2.1 Cryotherapy Dataset

This dataset is composed of cryotherapy treatment for 90 patients and it has 7 features for each observation. The dataset is divided into two groups that either the patient was cured after treatment or he still has the symptoms of disease. We applied model selection criteria to find the correct

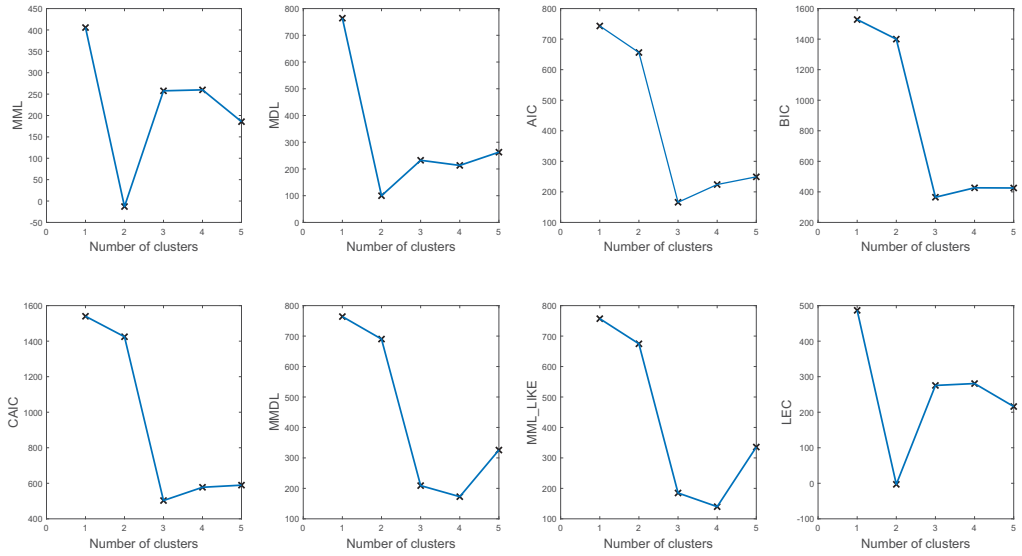


Figure 2.15: Model Selection Criteria for Cryotherapy Dataset

number of clusters in the dataset and it was observed that MML has successfully determined the number of categories in the dataset. We also observed that MDL and LEC have also determined the correct categories in the dataset. However, rest of the criteria have failed to find the correct number of clusters. The results of this experiments are given in first row of Table 2.25 and plotted in Fig. (2.15).

2.6.2.2 Statlog (Heart) Dataset

This dataset is collected from the information of 270 patients and each observation has 13 attributes. The dataset is divided into two groups where heart disease is present or absent in a patient. We applied our model selection and clustering algorithm to examine the performance of proposed technique. It was observed that MML, MDL and LEC have correctly determined the number of classes in the dataset, but rest of approaches could find the correct number of classes in our experiment. The results are given in second row of Table 2.25 and plotted in Fig. (2.16) to examine the performance of each model selection criterion as compared to MML.

2.6.2.3 Parkinsons Dataset

This dataset is composed of biomedical voice measurements from 31 people, where 23 of them were diagnosed with Parkinson disease. The data were collected by recording 195 voices from these individuals and each observation has 23 features. These features represent particular voice

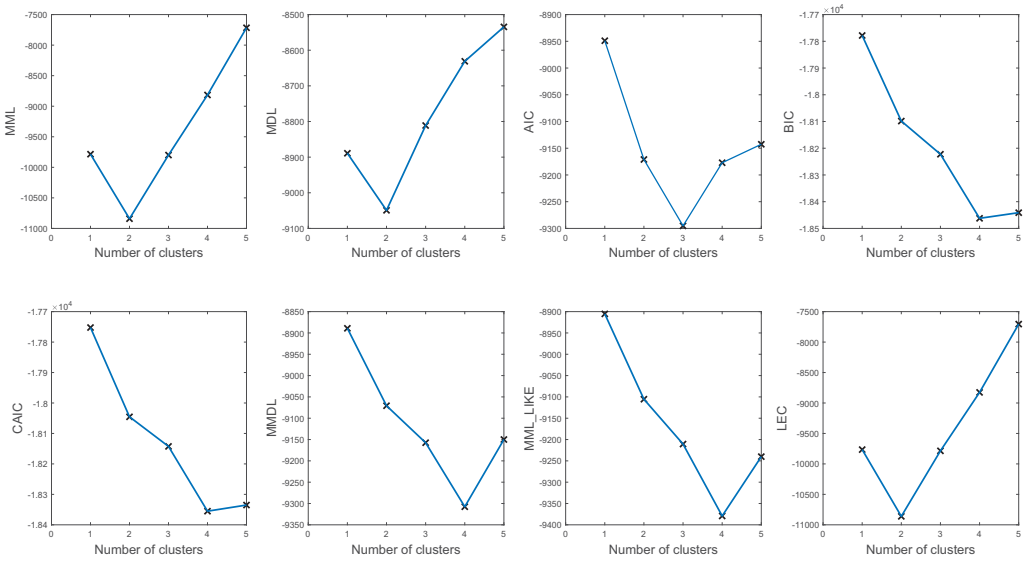


Figure 2.16: Model Selection Criteria for Statlog (Heart) Dataset

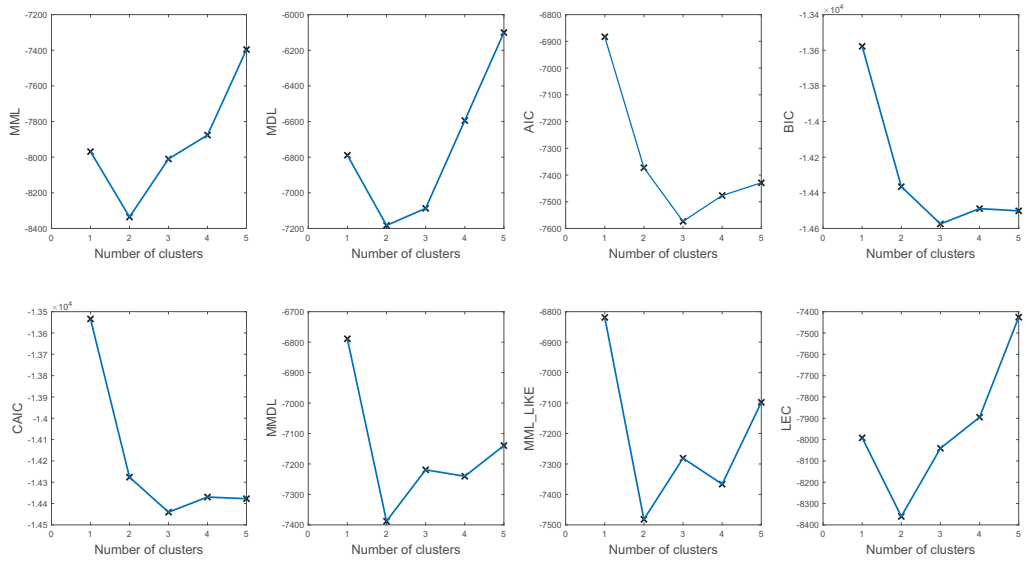


Figure 2.17: Model Selection Criteria for Parkinsons Dataset

measures. The objective of this dataset is to categorize between healthy and people with Parkinson disease [124, 125]. We applied the model selection to determine the number of classes from this dataset and MML, MDL, MMDL, MML_{Like} and LEC have correctly identified the number of categories. However, AIC, BIC and CAIC failed to determine the number of clusters. Model selection results are given in Table 2.25 and plotted in Fig. (2.17).

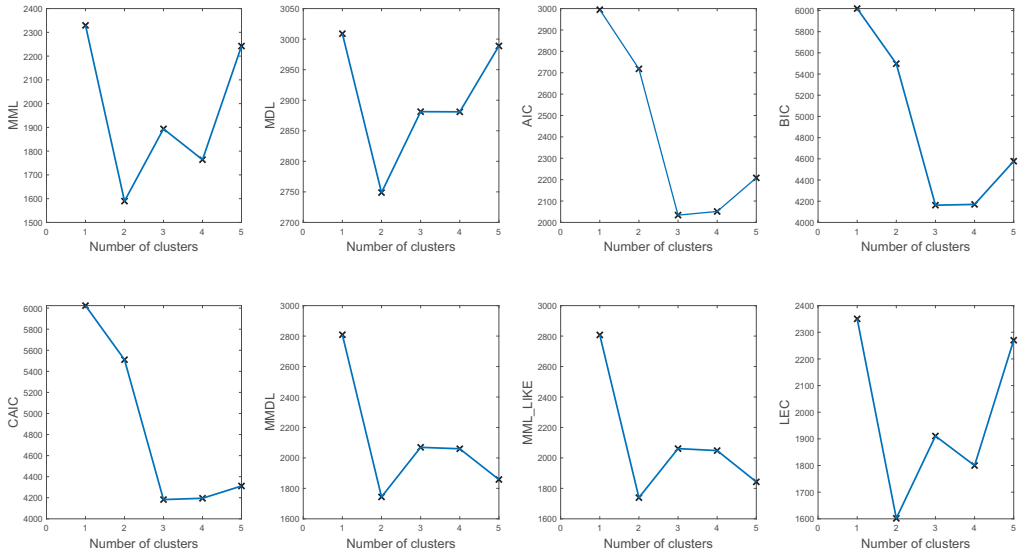


Figure 2.18: Model Selection Criteria for Haberman Dataset

2.6.2.4 Haberman's Survival Dataset

This dataset is composed of information from a study on patients who survived from breast cancer after treatment. The data is categorized between patients who survived 5 years or longer after the surgery or died within 5 years. We have applied model selection through MML to facilitate the clustering process and it is observed that MML, MDL, MMDL, MML$_{Like}$ and LEC have successfully identified the number of classes whereas rest of model selection techniques have failed to give the correct information. The results are give in Table 2.25 and plotted in Fig. (2.18).

2.6.2.5 Breast Cancer Coimbra Dataset

The dataset is composed of 116 observations recorded for 64 patients and 52 healthy persons and it has 10 quantitative features from anthropometric data, gathered from routine blood analysis. The dataset was created to make prediction model which can potentially be used as biomarker for breast cancer. We applied model selection criteria to investigate the performance of our proposed approach and it was observed that MML, MDL MMDL, MML$_{Like}$ and LEC have correctly identified the number of clusters in the dataset whereas rest of the criteria (AIC, BIC and CAIC) were unsuccessful in this test. Experimental results are provided in Table 2.25 and plotted in Fig. (2.19).

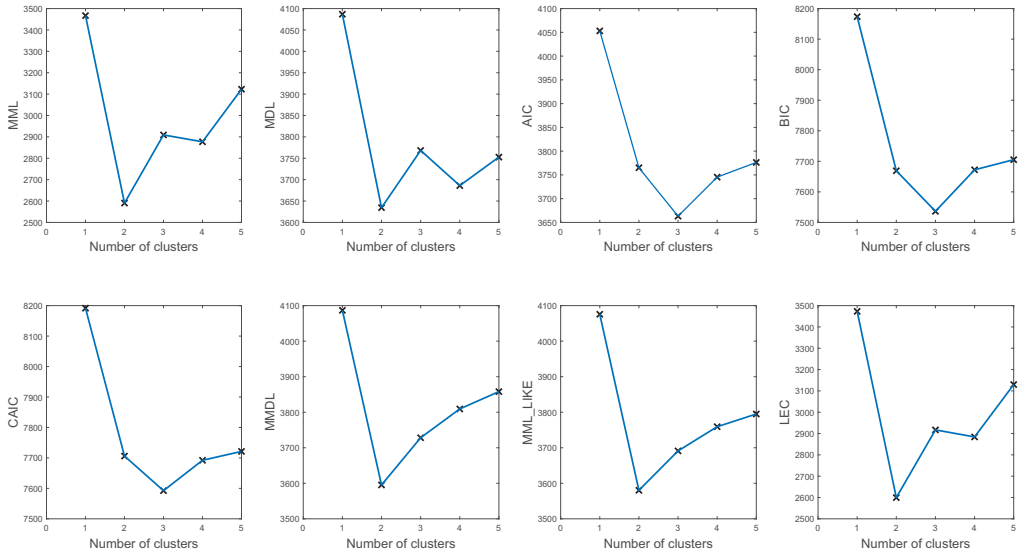


Figure 2.19: Model Selection Criteria for Breast Cancer Dataset

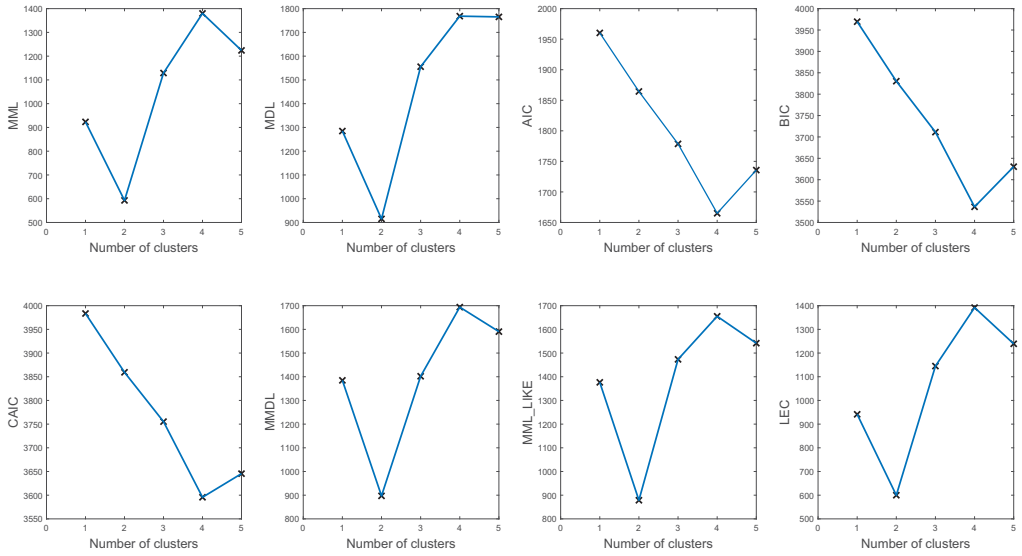


Figure 2.20: Model Selection Criteria for Immunotherapy Dataset

2.6.2.6 Immunotherapy Dataset

This dataset was collected with information about Immunotherapy treatment on 90 patients and it contains 8 features for each observation. The classes are defined as whether the patient is cured after the treatment or not. We conducted our experiments for finding the number of clusters in the dataset and it was observed that MML, MDL MMDL, MML_{Like} and LEC have successfully

identified the number of clusters in the dataset and AIC, BIC and CAIC have been unsuccessful during this experiment. The results of this experiment are provided in $6th$ row of Table 2.25 and plotted in Fig. (2.20).

2.6.2.7 Mammographic-Masses Dataset

Mammography is considered to be the most effective method for breast cancer screening and the purpose of this dataset is to help and improve the diagnostic of breast cancer after mammographic screening which is very difficult task and many computer aided systems have been developed to improve this process [124]. The dataset is composed of 961 observations, with 6 attributes and data is categorized into benign or malignant categories. We conducted our experiments for model selection using MML and other criteria and it was observed that MML, MDL and LEC were successful for determining the number of clusters in data whereas rest of techniques were unable to give correct results. Experimental results are given in $7th$ row of Table 2.25 and they are plotted in Fig. (2.22).

2.6.2.8 Blood Transfusion Service Center Dataset

This dataset was created from blood donation information to demonstrate a RFMTC marketing model which has details given in [124, 128]. The dataset was created by selecting the information of 748 donors randomly from donor database of blood donor transfusion service center and it is composed of 5 attributes and categorized into blood donated or not donated. We applied model selection criteria to examine the performance of our proposed approach and it was observed that MML, MDL MMDL, MML_{Like} and LEC have correctly identified the number of mixture components during clustering. The results of this test are demonstrated in Table 2.25 and Fig. (2.23).

2.6.2.9 Fertility Diagnosis Dataset

The dataset is composed of semen samples from 100 volunteers, which are analyzed according to WHO 2010 criteria and each observation has 10 attributes and classified as normal or altered. We conducted our model selection experiments on this dataset and it was observed that only MML and LEC have correctly identified the number of clusters in the data. The results of this experiment are provided in Table 2.25 and demonstrated in Fig. (2.21).

2.6.2.10 SPECTF Heart Dataset

This dataset deals with diagnosis of cardiac Single Proton Emission Computed Tomography (SPECT) images. The database is composed of SPECT images from 267 patients, and it was processed to

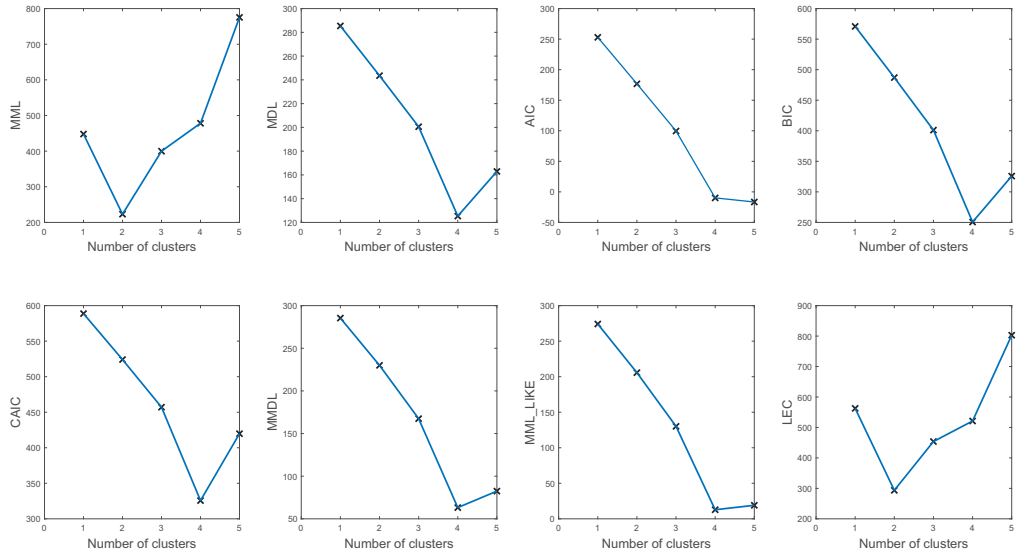


Figure 2.21: Model Selection Criteria for Fertility Diagnosis Dataset

Table 2.25: Number of Clusters Determined by Different Criteria using BGMM for Medical Datasets

Data set	D	N	$\hat{K_*}$	Model Selection Criteria							
				MML	MDL	AIC	BIC	CAIC	MMDL	MML_Like	LEC
Cryotherapy	7	90	2	**2**	**2**	3	3	3	4	4	**2**
Statlog (Heart)	13	270	2	**2**	**2**	3	4	4	4	4	**2**
Parkinsons	23	195	2	**2**	**2**	3	3	3	**2**	**2**	**2**
Haberman	3	306	2	**2**	**2**	3	3	3	**2**	**2**	**2**
Breast Cancer	10	116	2	**2**	**2**	3	3	3	**2**	**2**	**2**
Immunotherapy	8	90	2	**2**	**2**	4	4	4	**2**	**2**	**2**
Mammographic	6	961	2	**2**	**2**	4	3	3	3	3	**2**
Transfusion	5	748	2	**2**	**2**	4	3	3	**2**	**2**	**2**
Fertility	10	100	2	**2**	4	5	4	4	4	4	**2**
SPECTF Heart	44	267	2	**2**	**2**	3	3	3	4	4	**2**

extract the features to represent original images and it is composed of 44 features for each image and classified as normal or abnormal. We conducted our experiment to examine the viability of model selection on this dataset and it was observed that MML, MDL and LEC have correctly identified the number of categories in the data and rest of criteria used in the test were unable to get the correct result. The results are provided in Table 2.25 and demonstrated in Fig. (2.24).

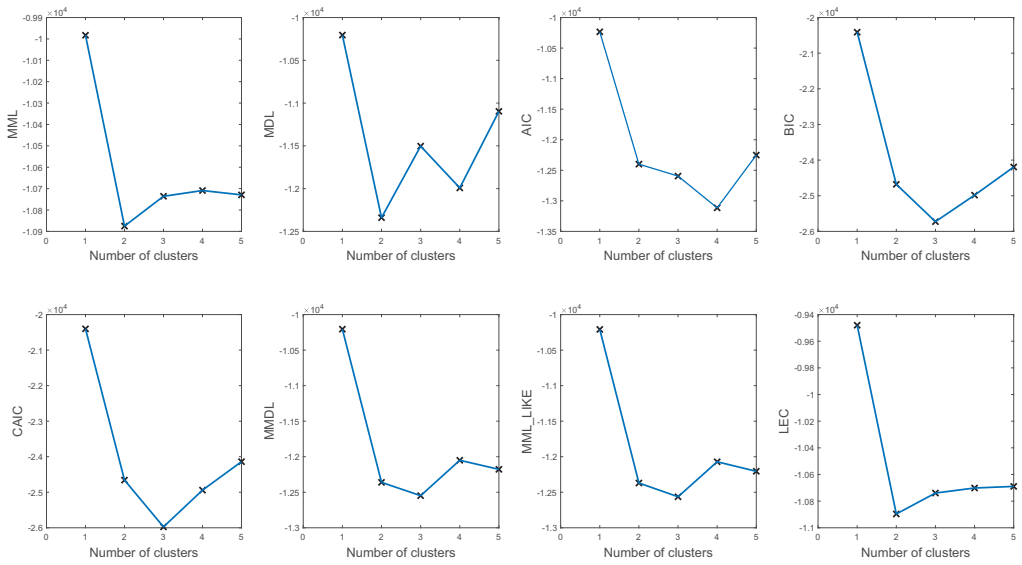


Figure 2.22: Model Selection Criteria for Mammographic-Masses Dataset

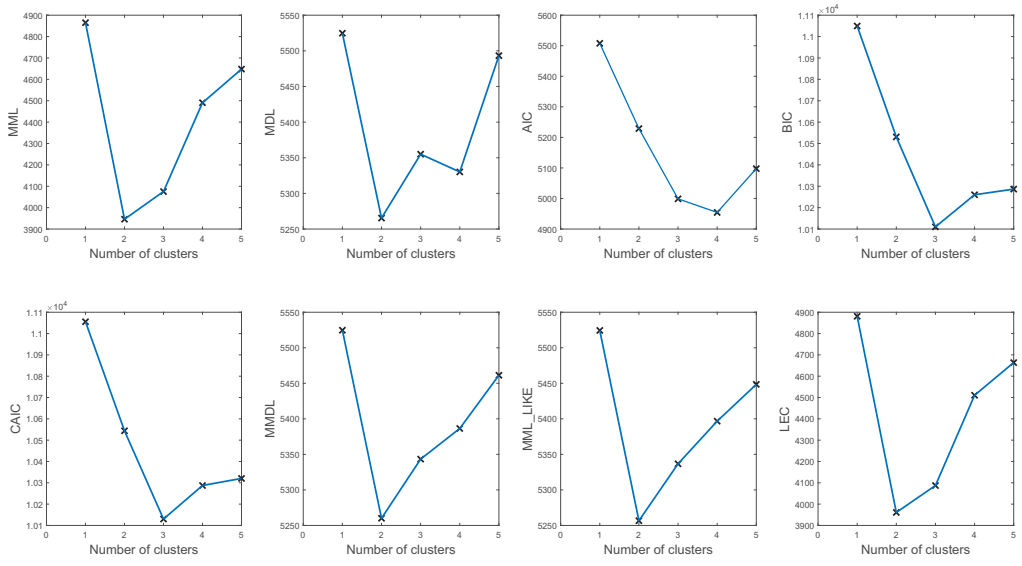


Figure 2.23: Model Selection Criteria for Transfusion Dataset

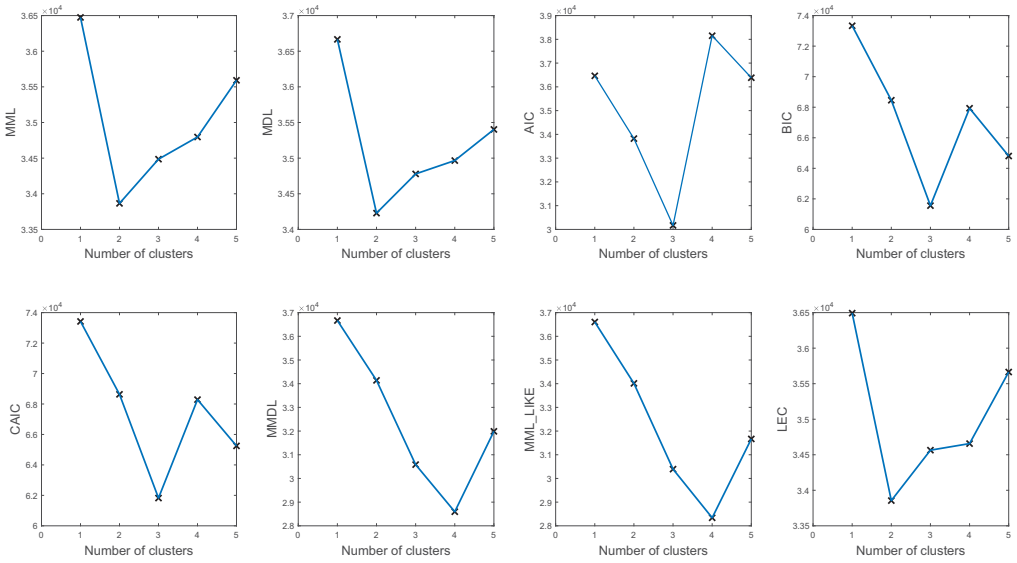


Figure 2.24: Model Selection Criteria for SPECTF Heart Dataset

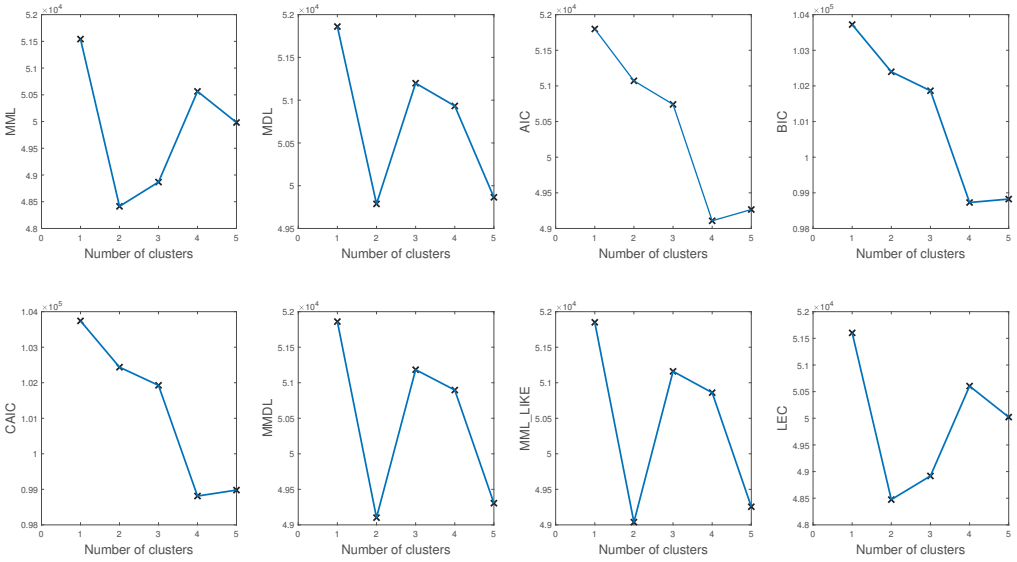


Figure 2.25: Model Selection Criteria for TSP Speech Dataset

2.6.3 Model Selection on TSP Speech Dataset

After observing the performance of proposed model selection in several medical datasets, we conducted a similar set of experiments on TSP dataset which is composed of data from two speaker classes (male and female). The feature are extracted in a similar manner are described in our clustering experiment for this dataset. From the set of experiments conducted on TSP dataset, it is

observed that MML, MDL MMDL, MML$_{Like}$ and LEC have successfully identified the number of classes whereas AIC, BIC and CAIC were unable to give correct results. Experimental results for model selection on TSP dataset are demonstrated in Table 2.26 and Fig. (2.25).

2.6.4 Model Selection on Free Spoken Digits Dataset

We extended our experiments on model selection using MML for Spoken Digits dataset and selected the parts of dataset with 2, 3, 4 and 5 categories for a comprehensive analysis of model selection criteria to examine the performance of proposed approach in speech data applications. For the experiments, 400, 600, 800 and 1000 speech files were selected for 2, 3, 4 and 5 classes, respectively. The features are extracted in a similar fashion described in our clustering experiment for this dataset. We conducted experiments for all the mentioned scenarios for different classes and it is observed that for 2 classes, MML and LEC have demonstrated their success in determining the number of clusters. For experiments with 3 categories of speech data from spoken digits, MML, MDL MMDL, MML$_{Like}$ and LEC have correctly identified number of clusters. With data from 4 categories, MML, MDL MMDL, MML$_{Like}$ and LEC have their success in model selection. For the case with 5 categories, MML, MDL and LEC have demonstrated their success in correctly identifying the number of classes in the data. The results of these experiments are provided in Table 2.26 and plotted in Figs. (2.26, 2.27, 2.28 & 2.29) to examine the performance of each model selection criterion as compared to MML.

2.6.5 Model Selection on MNIST Dataset

We have selected MNIST dataset for testing the proposed model selection criteria with 2, 3, 4 and 5 classes. We have selected 2000, 3000, 4000 and 5000 images for 2, 3, 4 and 5 categories, respectively and features are extracted in a similar fashion as described in the clustering experiments for this dataset. Model selection criteria with MML and different techniques are applied in all these experimental scenarios and for the case when data is composed of 2 and 3 categories, it is observed that MML and LEC have demonstrated their success in model selection. For 4 categories of data from MNIST dataset, MML, MDL MMDL, MML$_{Like}$ and LEC have correctly identified the number of clusters in the dataset. For 5 categories, MML, MMDL, MML$_{Like}$ and LEC have shown their success in model selection. Complete results of model selection for MNIST dataset is provided in Table (2.26) plotted in Figs. (2.30,2.31,2.32 & 2.33) to demonstrate the performance of model selection using MML.

Table 2.26: Number of Clusters Determined by Different Criteria using BGMM for Speech and Image Datasets used in clustering applications

Data set	D	N	$\hat{K^*}$	Model Selection Criteria							
				MML	MDL	AIC	BIC	CAIC	MMDL	MML_Like	LEC
TSP	40	1320	2	**2**	**2**	4	4	4	**2**	**2**	**2**
Spoken Digits 2	40	400	2	**2**	3	3	3	3	3	3	**2**
Spoken Digits 3	40	600	3	**3**	**3**	2	2	2	**3**	**3**	**3**
Spoken Digits 4	40	800	4	**4**	**4**	3	3	3	**4**	**4**	**4**
Spoken Digits 5	40	1000	5	**5**	**5**	6	6	6	4	4	**5**
MNIST 2	50	2000	2	**2**	5	5	5	5	5	5	**2**
MNIST 3	50	3000	3	**3**	5	5	5	5	5	5	**3**
MNIST 4	50	4000	4	**4**	**4**	7	7	7	**4**	**4**	**4**
MNIST 5	50	5000	5	**5**	4	6	6	6	**5**	**5**	**5**
Fashion MNIST 2	50	2000	2	**2**	3	4	4	4	3	3	**2**
Fashion MNIST 3	50	3000	3	**3**	**3**	5	5	**3**	**3**	**3**	**3**
Fashion MNIST 4	50	4000	4	**4**	**4**	6	6	6	**4**	**4**	**4**
Fashion MNIST 5	50	5000	5	**5**	4	6	6	6	4	4	**5**

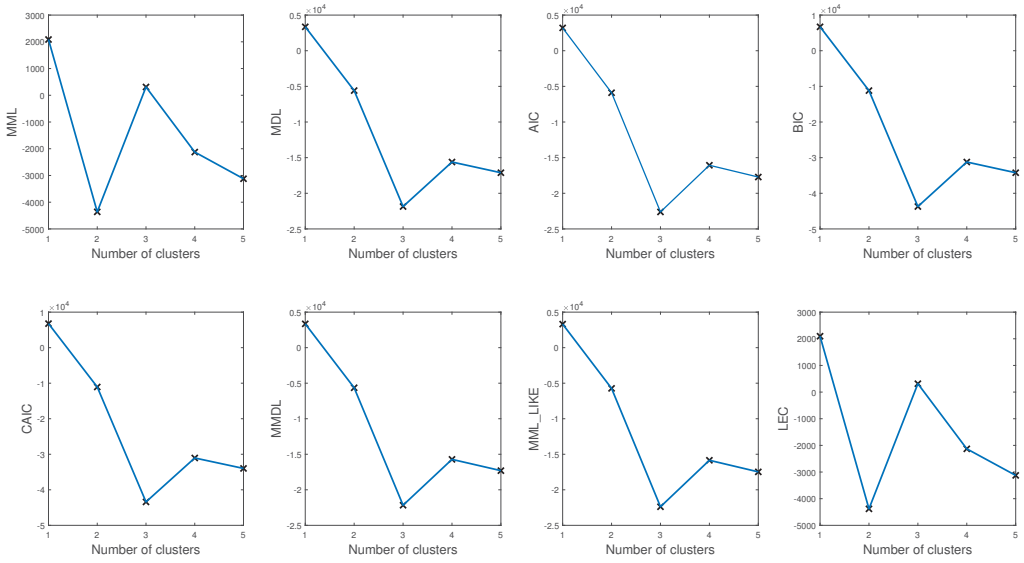


Figure 2.26: Model Selection Criteria for Spoken Digits Dataset with 2 classes

2.6.6 Model Selection on Fashion MNIST Dataset

We conducted our experiments to test the model selection using MML on Fashion MNIST dataset. The dataset is composed of 10 categories and we selected data from 2, 3, 4 and 5 categories in similar way as selected in Section 2.6.5. Model selection on Fashion MNIST dataset is extension on our

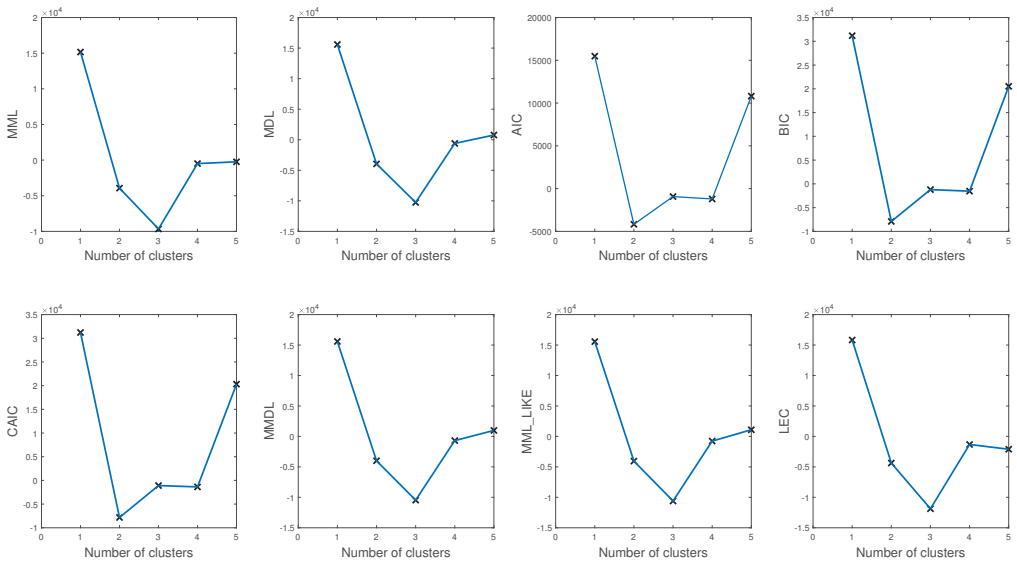


Figure 2.27: Model Selection Criteria for Spoken Digits Dataset with 3 classes

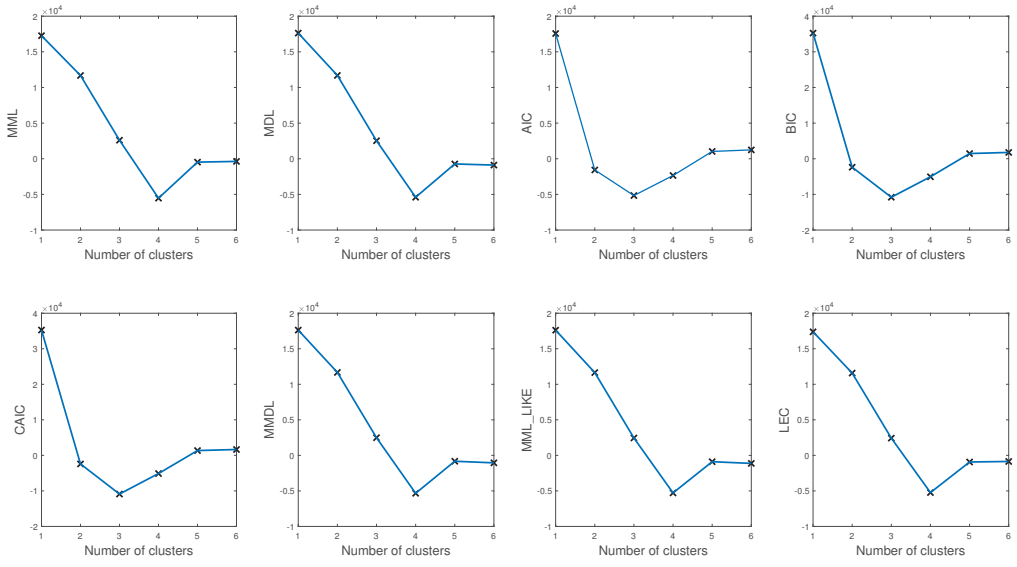


Figure 2.28: Model Selection Criteria for Spoken Digits Dataset with 4 classes

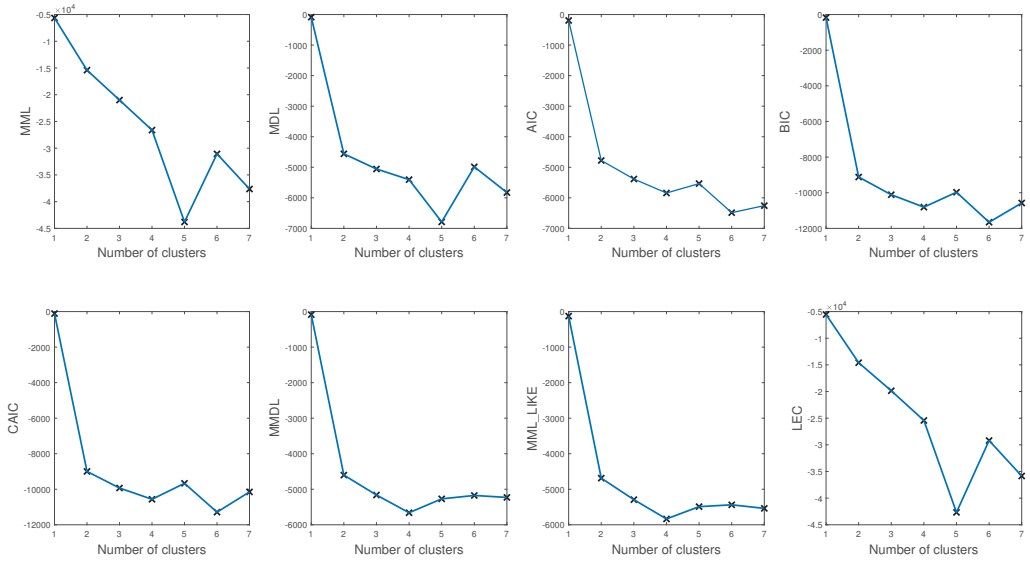


Figure 2.29: Model Selection Criteria for Spoken Digits Dataset with 5 classes

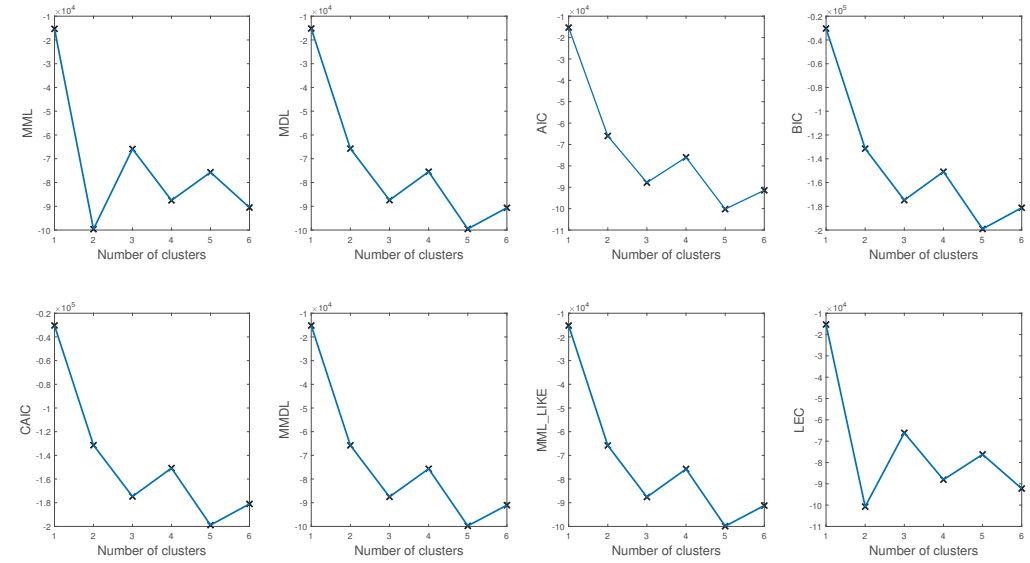


Figure 2.30: Model Selection Criteria for MNIST Dataset with 2 classes

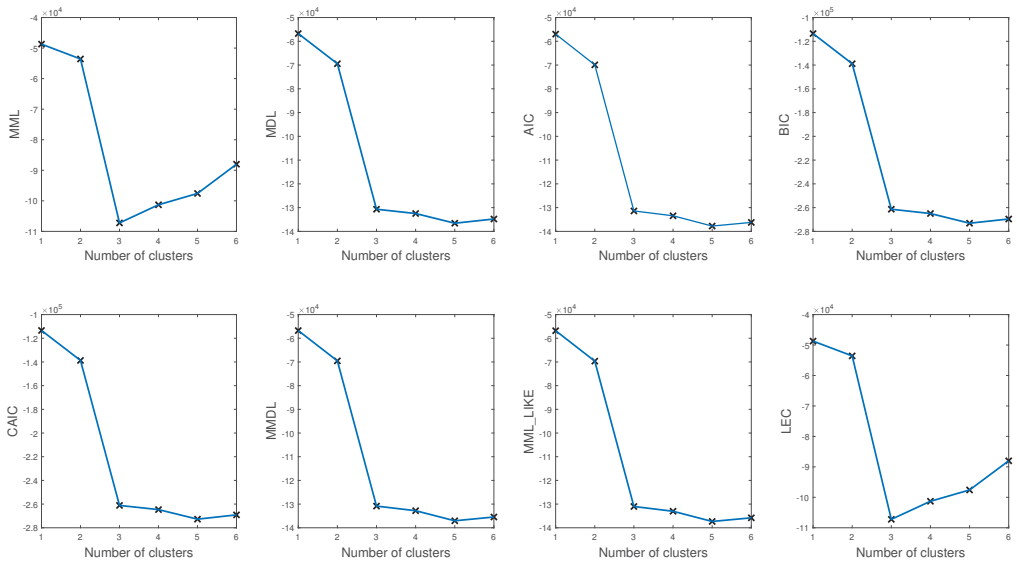


Figure 2.31: Model Selection Criteria for MNIST Dataset with 3 classes

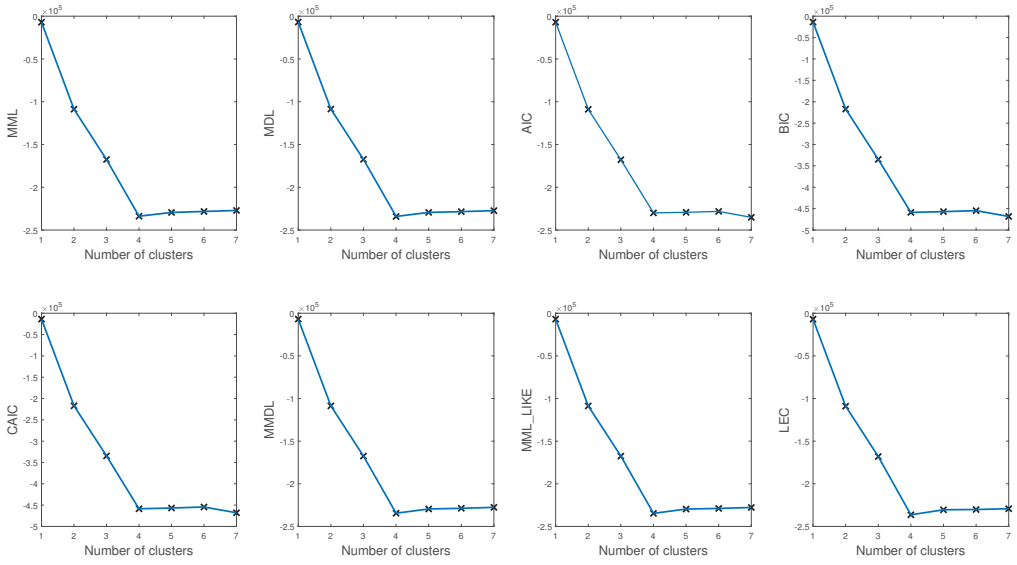


Figure 2.32: Model Selection Criteria for MNIST Dataset with 4 classes

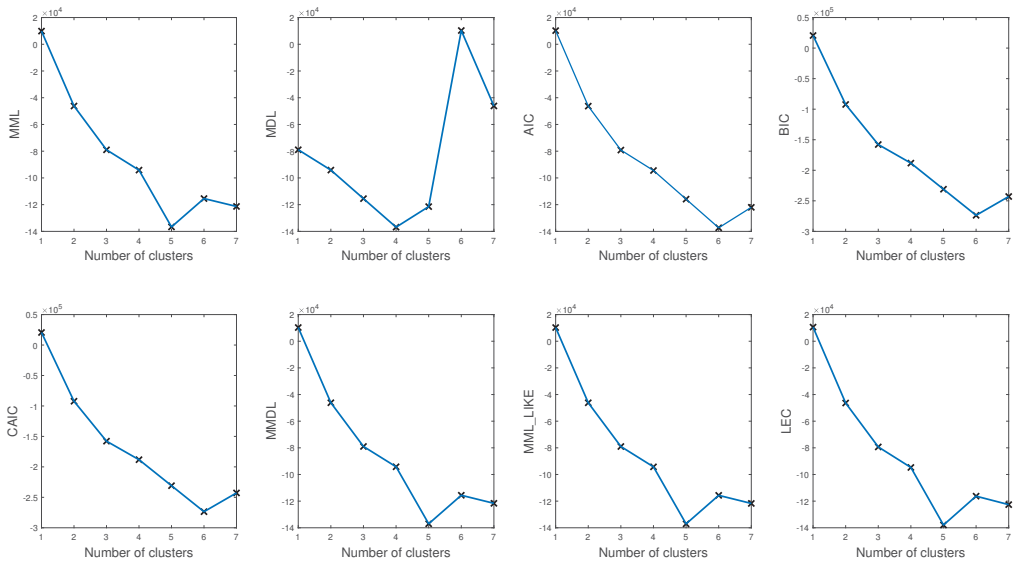


Figure 2.33: Model Selection Criteria for MNIST Dataset with 5 classes

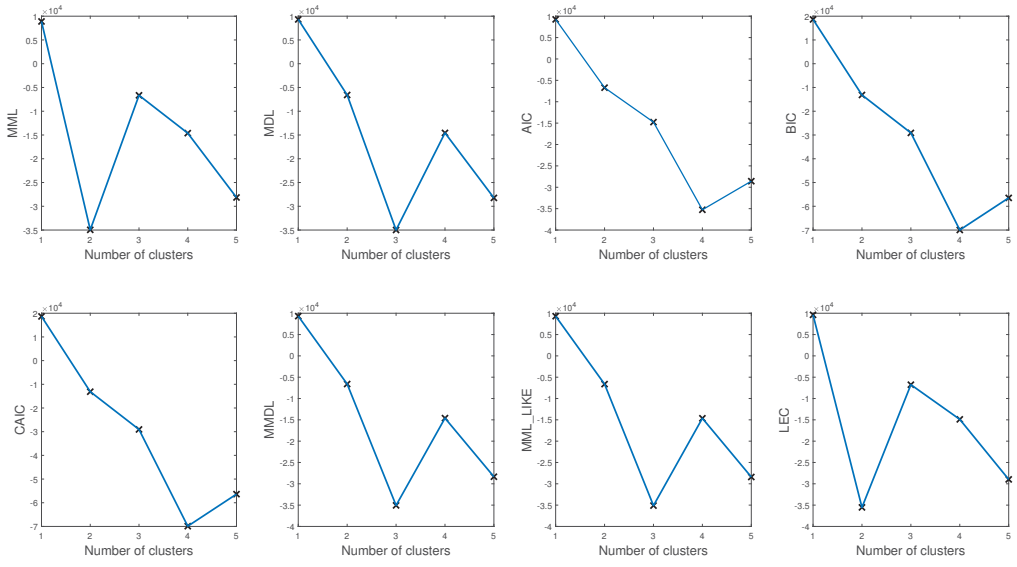


Figure 2.34: Model Selection Criteria for Fashion MNIST Dataset with 2 classes

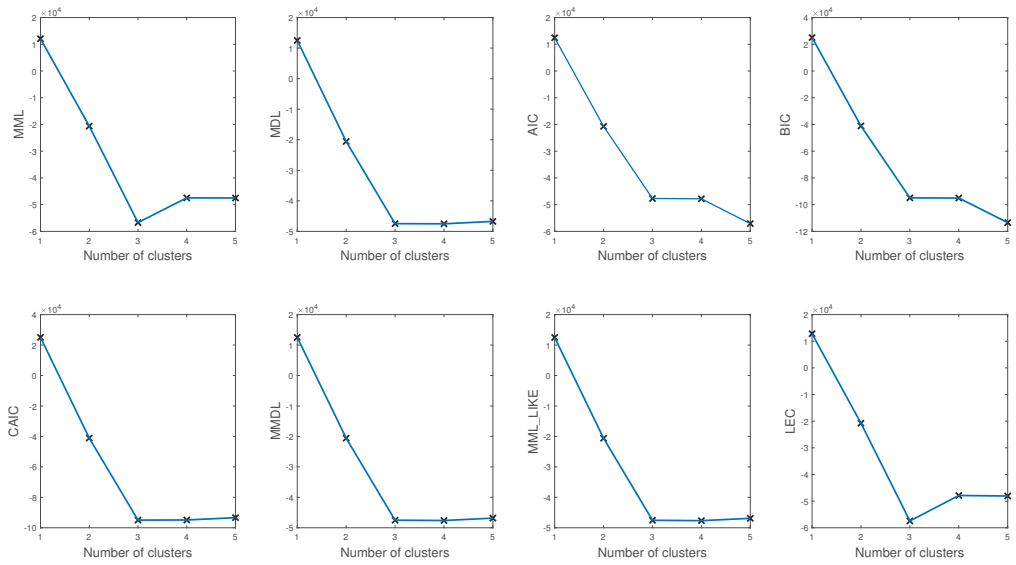


Figure 2.35: Model Selection Criteria for Fashion MNIST Dataset with 3 classes

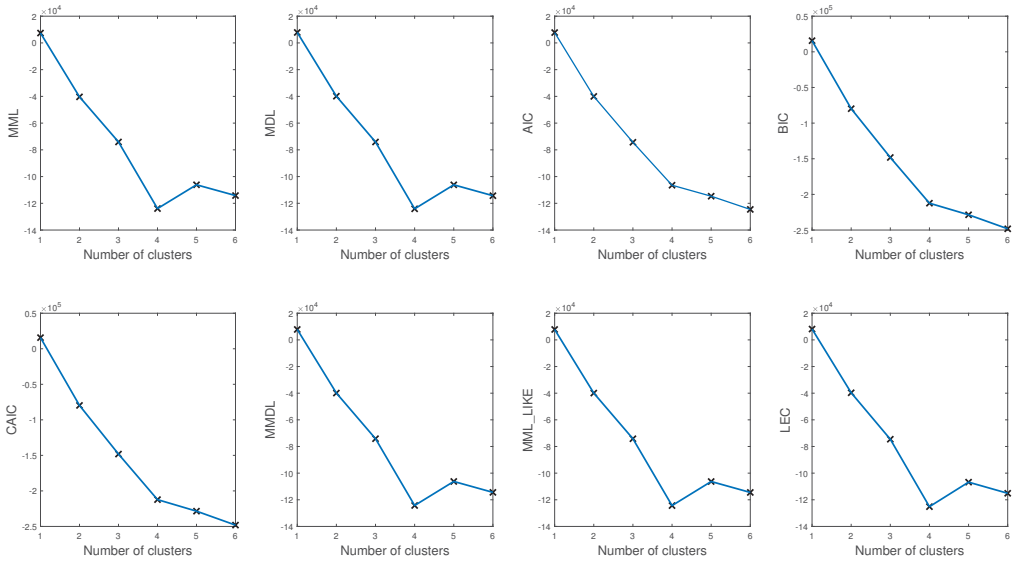


Figure 2.36: Model Selection Criteria for Fashion MNIST Dataset with 4 classes

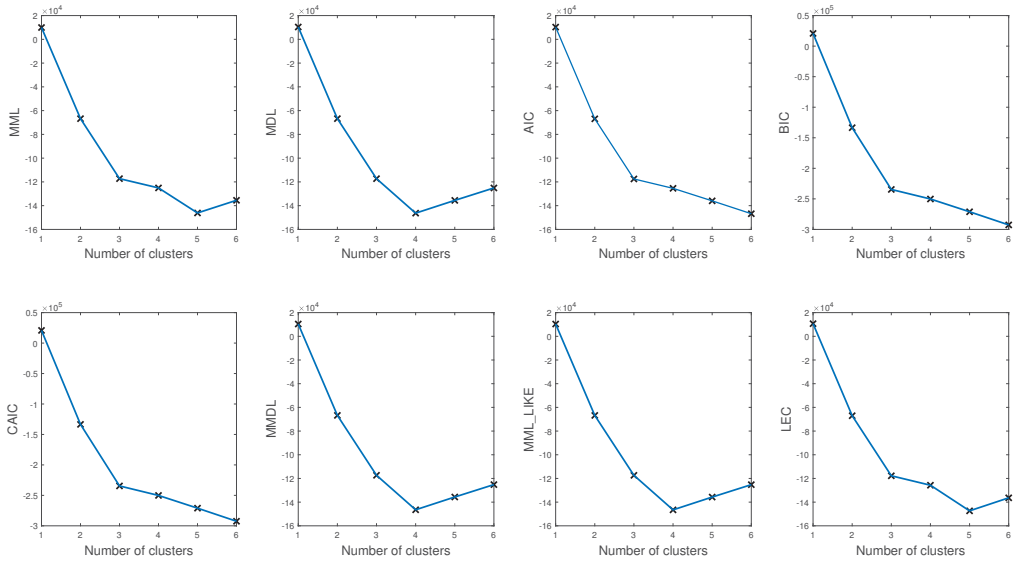


Figure 2.37: Model Selection Criteria for Fashion MNIST Dataset with 5 classes

previous experiments for clustering using BGMM and features are extracted in a similar manner as described in clustering experiments. We applied model selection using MML and other criteria on this dataset and for experiments with 2 categories of data, MML and LEC have demonstrated their success in model selection. For 3 categories, MML, MDL CAIC, MMDL, MML_{Like} and LEC have correctly identified the number of categories in data. For data with 4 categories, MML, MDL MMDL, MML_{Like} and LEC have shown their success in model selection. With data from 5 categories of Fashion MNIST, MML and LEC have correctly identified the number of categories of the data. Results for all model selection experiments on Fashion MNIST dataset are presented in Table 2.26 and demonstrated through plots for all the models applied using Figs. (2.34, 2.35, 2.34 & 2.35).

2.7 Discussion about BGMM and MML

In this chapter, multivariate bounded support Gaussian mixture model is introduced for data clustering in speech and image datasets to examine its performance. For this application, two speech datasets (TSP and Spoken Digits) and two images datasets (MNIST and Fashion MNIST) are selected. In speech datasets, MFCCs are used as method for feature extraction and inspired by the success of BoVW approach in computer vision applications, BoAW approach is also applied on MFCC features extracted from speech files. For images datasets, BoVW extracted from SIFT descriptors is employed during the pre-processing phase to represent each image of dataset. After the

code-book generation, BGGM is applied to perform clustering in speech and image datasets. In TSP dataset, clustering is performed to categorize the speech of male and female speakers. Spoken Digits dataset is composed of 10 categories and clustering is performed between 2, 3, 4, 5 and 10 categories. In MNIST and Fashion MNIST datasets, we also have 10 categories of images and BGMM is applied for clustering with 2, 3, 4, 5 and 10 classes of data. A similar experimental setting is also created using GMM in order to have a comparison with the performance of BGMM. From the set of experiments, it is observed that BGMM has performed well in clustering the speech and image datasets as compared to GMM. In code-book generation using bag of words approach in speech and image datasets, K-Means is applied to cluster the SIFT descriptors or MFCC features. We have proposed the application of BGMM for code-book generation and in order to observe the performance of our proposed approach, we have created a similar scenario with GMM and performance of BGMM, GMM and K-Means is examined for code-book generation using BoW. In order to examine the performance of BGMM in BoW creation, we have employed 3 clustering comparison scenarios at stage 1 (BGMM, GMM and K-Means) and 2 clustering comparison scenarios at stage 2 (BGMM, GMM). BGMM has demonstrated it effectiveness in clustering at both stages in this pipeline for categorizing the data in different classes for speech and image datasets.

In this chapter, we also have proposed model selection criterion for BGMM using MML which is validated through 10 different medical experiments datasets as first step in our validation process. The medical experiments datasets are used to model the behavior of different symptoms in patients which is further used to perform diagnostics using data modeling and it is very critical to correctly find the number of categories in the datasets to improve the data modeling capabilities and diagnostics process. The proposed MML criterion is also compared with 7 different methods for model selection in order to examine its performance. As second step to validate the performance of MML in model selection, speech and image datasets with different number of categories are considered. For TSP dataset, MML is performed with data of two classes, whereas for Spoken Digits, MNIST and Fashion MNIST, data are selected with 2, 3, 4 and 5 categories. The results of model selection using MML are compared with other model selection criteria for speech and image datasets and model selection criterion proposed for BGMM has demonstrated its effectiveness.

From the set of experiments performed for clustering, BGMM has demonstrated its success in data modeling as compared to GMM and model selection via MML has also proven effectiveness for correctly finding the number of clusters in data.

2.8 Speaker Verification Using Adapted Bounded Gaussian Mixture Model

Speaker recognition and verification has obtained significance importance and increased visibility in society as speech and audio technology, speech content and artificial intelligence based applications in business and other aspects of life continue to expand [131]. Due to rapid growth in artificial intelligence, speech data mining based on audio content and speaker identity is also growing and it has come to a point where it is becoming integral part of many applications and devices. A speaker recognition system performs two tasks: speaker identification and verification. The goal of speaker identification is to label an unknown speech signal with a speaker identity whereas in speaker verification, the task is to validate and confirm the claim of a speaker about its identity [131, 132]. Speaker verification has been used in many applications such as human-machine dialog systems, medical, forensics and security.

Mixture models have been extensively used in speaker verification in the past and many frameworks have been proposed [27, 28]. Adapted Gaussian mixture model or GMM-UBM speaker verification system was proposed in [29] and extensively applied in many applications and further researched [131, 133–140]. We propose the application of BGMM for adapted speaker model based on UBM. We propose to train the UBM using BGMM and then apply this trained UBM to adapt the speaker model similar to the one proposed in [29]. This approach is termed as BGMM-UBM or adapted bounded Gaussian mixture model for speaker verification. The proposed model is validated through several experiments on speech data and detection results have demonstrated its effectiveness as compared to Gaussian mixture model.

2.8.1 Universal Background Model for Speaker Verification

A UBM is employed in biometric verification system to represent speaker independent feature characteristics as compared to the speaker-dependent feature characteristics while making the decision of acceptance or rejection [134]. In [29], a verification system is modeled around likelihood ratio test, using GMMs for likelihood functions, UBM for alternative hypothesis modeling and Bayesian adaption to obtain speaker models from UBM. In this chapter, we have proposed BGMM for training in UBM and adaptation of speaker model. In the subsections below, we will describe the application of BGMM in the speaker verification system based on UBM, speaker adaptation and likelihood ratio test. This model is an extension of the work proposed in [29] and is referred as Bounded Gaussian Mixture Model-Universal Background Model (BGMM-UBM) speaker verification system and it is given in Fig. (2.38).

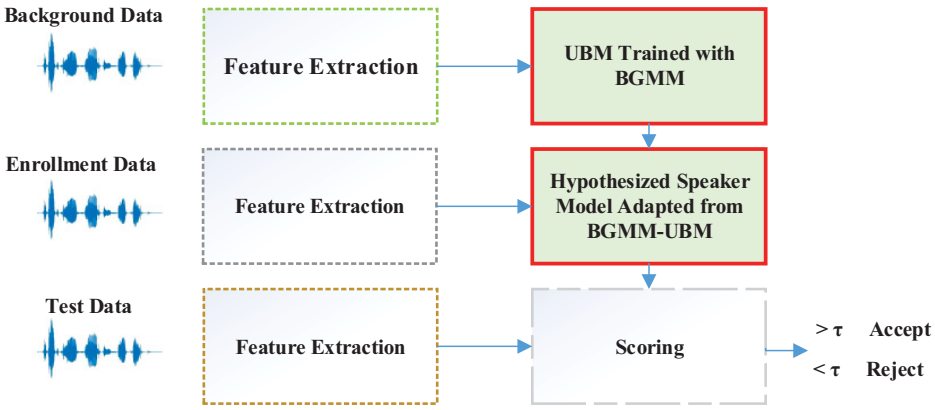


Figure 2.38: Block diagram of Speaker Verification with BGGM-UBM

2.8.1.1 Likelihood Ratio Detector

For the development of UBM in adapted speaker model, likelihood ratio test is very important to describe. If we have a test speech signal T and a hypothesized speaker S, the task of speaker verification is to find out if speaker T is from S and it can be stated as hypothesis test between

$$H_0 : T \text{ is from hypothesied speaker } S$$
$$H_1 : T \text{ is not from hypothesied speaker } S$$

The likelihood ratio test for deciding these two hypothesis is as follows:

$$\frac{p(T|H_0)}{p(T|H_1)} \begin{cases} \geq \tau \text{ accept } H_0 \\ < \tau \text{ reject } H_0 \end{cases}, \tag{2.45}$$

where τ is the decision threshold and $p(T|H_i), i = 0,1$, is the probability density function for hypothesis H_i computed for test speech signal T. It is also referred as likelihood of the hypothesis for test speech segment T. The primary objective of developing speaker verification system is to determine techniques to compute the likelihood ratio by computing the two likelihoods, $p(T|H_0)$ and $p(T|H_1)$. The first stage in speaker verification system is front-end processing which has major goal to extract features to express the speaker-dependent information for speech data. The sequence of feature vectors for test signal T can be represented as $\mathscr{Y} = \{\vec{Y}_1, ..., \vec{Y}_L\}$, where $\vec{Y}_l$ is a feature vector indexed for lth segment as: $[l \in 1, ..., L]$. The likelihoods of H_0 and H_1 are computed using these features vectors extracted from test speech signal. In speaker verification systems, hypotheses H_0 and H_1 are represented by λ_s and λ_0 for test speech from hypothesized speaker

and test speech not from hypothesized speaker, respectively. In our proposed approach, feature vectors for test speech signal will be represented by BGD for H_0 and λ_s will represent mean vector and covariance matrix parameters of BGD. The alternative hypothesis is modeled through pool of several speakers and we have proposed BGMM for this modeling [29, 131, 133–136, 141, 142]. It is termed as universal background model and details are given in subsection (2.8.1.2). Usually, likelihood ratio test is performed in the logarithmic scale, which can be expressed as follows:

$$\Lambda(\mathscr{Y}) = \log p(\mathscr{Y}|\lambda_s) - \log p(\mathscr{Y}|\lambda_0) \tag{2.46}$$

2.8.1.2 Universal Background Model using BGMM

Hypothesis test H_0 can be modeled with a mixture model (BGMM in our case) and it is well defined and it is estimated by using training speech data from speaker S. The model for λ_0 is not well defined since it has to represent the entire space of possible alternatives to hypothesized speaker S. In the literature, two approaches for modeling λ_0 have been proposed. In the first approach, in order to cover the space for alternative hypothesis, set of other speakers can be used. The drawback of this approach is large number of hypothesized speakers where each requires its own background speaker set. In the second approach, λ_0 is modeled thorough a pool of several speakers and it is further applied to train a single model. From the population of speakers expected during recognition, a collection of speech signals for all speakers is used to train a single model for alternative hypothesis. We have applied BGMM for modeling alternative hypothesis and in literature, it is termed as universal background model (UBM) [29, 134].

2.8.1.3 Adaptation of Speaker Model with BGMM

In our proposed BGMM-UBM system, hypothesized speaker model is derived by adapting the parameters of UBM by applying training speech of the speaker and maximum a posteriori (MAP) as described for GMM in [29, 141]. The basic idea in adaptation approach is to obtain the hypothesized speaker's model by updating the parameters in the UBM via adaptation. The adaptation of speaker model has two steps like EM algorithm in the estimation process. In the first step, we estimate the sufficient statistics parameters of speaker's training data for each mixture in the UBM. In the second step, these new sufficient statistics are combined with old sufficient statistics from UBM mixture parameters [29, 131]. Let $\mathscr{Y} = \{\vec{Y}_1, ..., \vec{Y}_L\}$ represents the sequence of feature vectors obtained from training data of the hypothesized speaker. Given a UBM and speakers training data $\mathscr{X}$, we compute the posterior probability for the training data with respect to the components of mixtures of UBM. For the jth component in the UBM, posterior probability is computed as

follows:

$$p(j|\vec{Y}_l) = \frac{p(\vec{Y}_l|\xi_j)p_j}{\sum_{j=1}^{K} p(\vec{Y}_l|\xi_j)p_j} \tag{2.47}$$

In the next step, $p(j|\vec{Y}_l)$ and training data of the speaker is used to compute the sufficient statistics for mixing weight, mean and covariance parameters using BGMM as follows:

$$N_j = \sum_{l=1}^{L} p(j|\vec{Y}_l) \tag{2.48}$$

$$E_j(Y) = \frac{1}{N_j} \sum_{l=1}^{L} p(j|\vec{Y}_l)\vec{Y}_l \tag{2.49}$$

$$E_j(Y^2) = \frac{1}{N_j} \sum_{l=1}^{L} p(j|\vec{Y}_l)\vec{Y}_l\vec{Y}_l^T \tag{2.50}$$

The maximum a posteriori adaptation update equations for mixing weight, mean and covariance are given as follows:

$$\hat{p}_j = [\alpha_j N_j/L + (1-\alpha_j)p_j]\beta \tag{2.51}$$

$$\hat{\mu}_j = \alpha_j E_j(Y) + (1-\alpha_j)\mu_j \tag{2.52}$$

$$\hat{\Sigma}_j = \alpha_j E_j(Y^2) + (1-\alpha_j)(\Sigma_j + \mu_j\mu_j^T) - \hat{\mu}_j\hat{\mu}_j^T \tag{2.53}$$

The scaling factor β in Eq. (2.51) is computed over all adapted mixture weights to ensure that they sum to unity. The variable α_j is represented as:

$$\alpha_j = \frac{N_j}{N_j + r} \tag{2.54}$$

where r is relevance factor. This parameter controls the adaptation parameters of BGMM in order to affect the hypothesized test speaker. In literature, it has been presented that only adaptation of mean vectors is most effective. In adaptation of speaker model, the posterior probability is computed with respect to BGD and UBM which is trained with BGMM. The rest of the adaptation equation are followed from the procedure explained in [29, 131].

2.8.2 Experiments and Results

2.8.2.1 Design of Experiments

In this section, we present experiments and results using our proposed BGMM-UBM system. We have conducted our experiments with TIMIT and TSP speech databases [91, 143]. The first step is front-end processing and we have performed feature extraction for background data, enrollment data and test data. Before feature extraction, voice activity detection (VAD) is used to distinguish between speech and non-speech parts of speech signals. The main reason of applying VAD is to assure that training process is not inferred with non-speech parts of data. For feature extraction, Mel Frequency Cepstral Coefficients (MFCCs) have been used. MFCCs has been widely used for speech recognition and we have used 39 dimensional MFCC features for front-end processing. The next step is to train the UBM using BGMM with the large part of data set (termed as background data in Fig. (2.38)) selected for UBM modeling. The next step is speaker enrollment and in this step, hypothesized speaker is adapted with BGMM-UBM. In the adaptation, first of all posterior probability (Eq.(2.47)) is computed with BGD by taking the parameters of the trained UBM model. Posterior probability is further used estimate the sufficient statistics (Eqs. (2.48-2.50)) required to update the parameters (Eqs. (2.51-2.53)) of the adapted speaker based on enrollment data. In the last step, a recognition score is calculated for test speaker, using likelihood ratio test (Eq. (2.46)) with parameters of hypothesized speaker and UBM. For the development of this framework, we have employed TIMIT and TSP speech corpora and several experiments are performed to examine the viability of proposed framework.

2.8.2.2 Experimental Framework and Results

The speaker verification based on our proposed approach is evaluated using TIMIT and TSP speech databases. TIMIT speech database consists of 6300 speech utterances having 630 speakers. Each speaker has spoken 10 speech utterances. The data set contain 4620 speech utterances (462 speakers) for training and 1680 speech utterances (168 speakers) for testing. The TSP speech database consists of 1378 speech utterances spoken by 23 speakers (11 male, 12 female). For 22 speakers, database has 60 speech utterances for each speaker whereas one speaker has 58 speech utterances.

In order to test our framework, we have created different experimental scenarios with both databases. First of all, we have taken the speech data of TSP database and 18 speakers (1080) are used to train the UBM via BGMM. For enrollment, 5 speakers are selected and 10 speech utterances are used for each speaker model and 10 speech utterances are employed for testing. The purpose of using this small database for this task is to check rapid response of proposed approach

Table 2.27: 5 Speakers confusion matrix using TSP database.

(a) BGMM-UBM

	S1	S2	S3	S4	S5
S1	7	1	0	1	1
S2	1	6	2	0	1
S3	0	1	8	1	0
S4	1	0	0	9	0
S5	1	1	0	1	7

(b) GMM-UBM

	S1	S2	S3	S4	S5
S1	6	1	0	2	1
S2	1	6	1	1	1
S3	0	1	7	1	1
S4	1	1	0	8	0
S5	1	0	1	1	7

in speaker verification. We have trained UBM via BGMM with different numbers of components of mixture model (2, 4, 8, 16, 32, 64, 128, 256) and in the enrollment step, 5 separate speaker models are adapted from trained UBM. To test each hypothesized speaker, we have 10 speech utterances and likelihood ratio is computed with respect to the hypothesized speaker model and it is also computed with respect to the remaining speaker models. The same process is repeated for all test speakers with respect to all speaker models. The likelihood ratio is compared with threshold value in order to accept or reject the particular speaker with respect to all speaker models. The verification results are computed for UBM trained for different number of components of mixture model and it is observed that speaker verification results are not changing much after UBM trained for 64 components. The speaker verification results when UBM was trained with 64 mixture components are given in Table (2.27). In the confusion matrix, $[S1,...,S5]$ are 5 speakers used for enrollment and testing. A comparison of our proposed BGMM-UBM framework with GMM-UBM is also performed with similar settings and detection results are provided in Table (2.27). From the comparison of detection rate, it is observed that application of BGMM in UBM has improved the recognition rate for speaker verification.

In the next experiment, TIMIT speech corpus is employed for the evaluation of proposed approach. For training the UBM via BGMM, 6200 speech utterances for 620 speakers are selected. In this experiment, we want to train the UBM with maximum available data and rest of 10 speakers are used for enrollment and testing. For each speaker having 10 speech utterances, 5 speech utterances are used for enrollment and rest of the 5 utterances are selected for test. The UBM is trained with the data selected for background model and it is trained for different mixture components (2, 4, 8, 16, 32, 64, 128, 256, 512) as in the experiment for TSP data set. In the enrollment step, 10 speaker models are adapted with trained UBM using 5 speech utterances for each speaker. The next step is to compute the likelihood ratio for all 10 speakers with respect to hypothesized speaker models in same manner as we have described for TSP data set. The likelihood ratio is further used to accept or reject the test speaker based on a threshold. The selection of threshold value is very critical in speaker verification because it can change the detection results. In this

Table 2.28: 10 Speakers confusion matrix using TIMIT database.

(a) BGMM-UBM

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	5	0	0	0	0	0	0	0	0	0
S2	1	4	0	0	0	0	0	0	0	0
S3	0	0	3	1	0	0	1	0	0	0
S4	0	0	0	4	0	0	0	1	0	0
S5	0	0	0	0	4	0	0	0	0	1
S6	0	0	0	0	0	5	0	0	0	0
S7	0	0	0	1	0	0	4	0	0	0
S8	0	0	0	0	0	0	0	5	0	0
S9	0	0	0	0	0	0	0	0	5	0
S10	1	0	0	0	0	0	0	0	0	4

(b) GMM-UBM

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	4	0	0	0	0	0	0	1	0	0
S2	0	4	0	0	0	1	0	0	0	0
S3	0	1	3	0	0	0	0	0	0	1
S4	0	0	1	4	0	0	0	0	0	0
S5	0	0	0	0	5	0	0	0	0	0
S6	0	0	0	1	0	4	0	0	0	0
S7	0	0	0	0	0	0	4	0	0	1
S8	0	0	0	0	0	0	0	5	0	0
S9	1	0	0	0	0	0	0	0	4	0
S10	0	1	0	0	0	0	0	0	0	4

case, lowest likelihood ratio is chosen for detection of speaker with respect to all speaker models. Detection results are computed with models having training and adaptation for different components of mixture model and it is observed that detection results are not changing much after 128 components fo mixture model. Speaker detection results computed when the UBM is trained with 128 components of mixture models are given in Table (2.28). In confusion matrix, $[S1,...,S10]$ are 10 speakers used for enrollment and testing in this experiment. A comparison of our proposed approach is also performed with GMM-UBM for 10 speaker verification and it is observed that our proposed approach has outperformed. However the trend of this improvement is not high and we have observed that since only 5 speakers are used both for enrollment and testing and they are not enough to clearly examine the improvement. In our future work, we are planning to use a large data set to clearly examine the performance of proposed model. In Table (2.28), detection results for BGMM-UBM and GMM-UBM are provided for a comparison.

Table 2.29: 5 Speakers confusion matrix, TSP database (Combined training)

(a) BGMM-UBM

	S1	S2	S3	S4	S5
S1	8	0	0	1	1
S2	1	9	0	0	0
S3	2	0	7	0	1
S4	0	1	1	8	0
S5	1	0	1	0	8

(b) GMM-UBM

	S1	S2	S3	S4	S5
S1	7	0	1	1	1
S2	1	7	0	2	0
S3	1	1	8	0	0
S4	1	1	1	7	0
S5	1	0	1	0	8

In the next experiment, both data sets are combined for training the UBM via BGMM. From TIMIT speech corpus, 6200 speech utterances (620 speakers) are selected and from TSP speech data set, 1080 speech utterances (18 speakers) are used in the training process for UBM. Although this combined data set is not well balanced, but the purpose of combining it for training is to get the trained model on maximum available data. Training of UBM is performed for different number of mixture components (2, 4, 8, 16, 32, 64, 128, 256, 512) and for adaptation and testing, two scenarios given in above two experiments are selected separately. The trained UBM is used for adaptation in 5 speakers detection (TSP) and 10 speaker detection (TIMIT). In this experiment, optimized number of components of mixture model for UBM training is observed as 128. The detection results for 10 speakers using TIMIT speech corpus have not changed in this experiments and they are not reported again separately for this experiment. The reason for having no change in detection results is that since there is not much difference in the training data as compared to the second experiment. However, training data are much increased by combining both data sets as compared to first experiment and for 5 speakers from TSP data set, detection results are improved and they are reported in Table (2.29). Proposed BGMM-UBM is compared with GMM-UBM and results are reported for both frameworks. It is observed that by employing BGMM in the training of UBM, recognition is improved.

From the above experiments, it is observed that BGMM has effectively demonstrated its viability in speaker verification for the training of UBM and further adaptation of speaker model via BGMM based UBM. The detection results of our proposed framework are better than GMM-UBM in each experiment. It is also observed that by increasing the training data for background model, detection rate is improved.

2.8.3 Discussion about BGMM-UBM

In this work, we have proposed BGMM for speaker verification system. In the proposed approach, UBM is developed by BGMM and speaker adaptation is performed by the trained UBM. The proposed framework is applied to the speaker verification task and two speech corpora (TIMIT & TSP) are employed for the development of experiments in order to examine the performance of proposed approach. Three different experimental scenarios are created for the validity of proposed framework and it is observed that there is clear improvement in detection results for all experiments and application of BGMM in speaker verification system has outperformed GMM for the similar setting. Future works could be devoted to the consideration of a larger data sets and the consideration of different mixture models within the same framework.

Chapter 3

Multivariate Bounded Support Laplace Mixture Model

In this chapter, bounded Laplace mixture model (BLMM) is proposed. The parameters of proposed model are estimated by maximum likelihood approach via expectation maximization (EM) and Newton Raphson algorithm. The model is proposed for data modeling to perform clustering using synthetic data for uni-variate and multivariate examples and real datasets of different medical experiments. BLMM is validated through correctness of estimated parameters for synthetic data and clustering accuracy of medical datasets. A new modeling scheme is also introduced for wavelet coefficients which is based on BLMM. It is applied to image clustering and content based image retrieval (CBIR) for feature extraction in wavelet domain. For feature extraction in this application, each image is decomposed into a set of wavelet subspaces and BLMM with two components is adopted to model the statistical characteristics of the wavelet coefficients for each wavelet subspace. The model parameters adapted from BLMM, represent the image features in wavelet domain for each subspace and selected to formulate the feature space which is further used in clustering and CBIR. In the framework for clustering and image retrieval, features extracted in wavelet domain are further modeled through BLMM to categorize images into different groups and trained model is adopted for CBIR. In order to perform image retrieval with trained model via BLMM, City-block distance, posterior probability and Kullback-Leibler divergence are introduced. We also propose a novel solution to compute Kullback-Leibler divergence which is very effective for image retrieval due to its low computational complexity and high retrieval rate. The effectiveness and viability of BLMM in texture image clustering and CBIR is demonstrated through UIUC, KTH-TIPS, DTD, STex and Kylberg databases. Different experiments are performed in the chosen applications and from the results, BLMM has demonstrated its effectiveness in modeling synthetic data, real datasets from medical experiments, feature extraction in wavelet

domain, image clustering, and CBIR.

In the above introduced method, features are extracted via bounded Laplace mixture model (BLMM) in wavelet domain. Due to nature of wavelet coefficients that can be modeled accurately with Laplace distribution, it is also proposed to apply classifiers based on this distribution, which leads us to introduce Naive Bayes classifier with Laplace distribution for image categorization. The proposed approach is validated through experiments on different texture image datasets and it has shown very good results as compared to the model based on Gaussian distribution. The generalized Gaussian distribution is a generalization of both Laplace and Gaussian distributions, thus we have introduced also Naive Bayes classifier with generalized Gaussian distribution to achieve better performance as compared to the above two models. The proposed approach is also validated through extensive experiments. Classification results are presented by different performance metrics to ensure the effectiveness of proposed algorithms in texture image classification.

3.1 Introduction

Unsupervised learning plays an important role in pattern recognition and finite mixture models as unsupervised learning approach are considered as flexible and powerful tool in statistical pattern recognition, for modeling one-dimensional and multi-dimensional data. Furthermore, they have been successfully applied in various interesting applications in computer vision, speech and image processing, pattern recognition and machine learning [30, 31]. Mixture models are built on the idea that data can be represented as a mixture of multiple probability distributions. Gaussian mixture model (GMM) has become quintessential model that is widely used for several applications. It is a well-known fact that Gaussian distributions use quadratic distance between the data points and their means for data clustering, which makes the clustering process sensitive to outliers [52, 77]. Choosing the distributions for data modeling with $L1$ norm distance is one viable solution to this problem [144]. One of the ways to handle this problem is by applying k-median algorithm in data clustering which handles the outliers by taking the sum of the absolute distances between the data points and its class centroid. As generalization to k-median algorithm, a mixture of Laplace distributions was introduced which can be used in clustering applications primarily to handle outliers and this approach proves to be very effective in many data modeling applications where density of data is more close to Laplace distribution [52–54]. Laplace mixture model (LMM) has been used in many successful applications such as blind source separation, feature selection and feature representation in image processing [52, 145–148].

Finding an appropriate model that is able to approximate the data without over-fitting is essential to handle more complex and challenging tasks in data engineering. This has been a booming

field of research in recent years. Various models have been proposed with different distributions like Gaussian mixture models, student's t mixture model, asymmetric Laplace mixture model and generalized Gaussian mixtures models [16, 47, 51, 149, 150]. These models prove to be an efficient choice for modeling the data in multiple applications. In spite of that, it is notable that these models consider the data to run from $(-\infty, +\infty)$ i.e. they are unbounded. Hence in this chapter we introduce bounded Laplace mixture model (BLMM) as a substitute to unbounded mixtures for modeling data. When it comes to parameter estimation, we use the maximum likelihood approach which is a norm as it has proved to be very efficient and for optimization we have used EM algorithm and Newton-Raphson method.

In order evaluate the proposed model, it is applied to many data modeling applications including synthetic and real datasets. As first test, uni-variate and multivariate synthetic data are generated with known parameters of Laplace distributions and proposed model is applied to model these data. The correctness and closeness of learned parameters to the actual parameters validate the effectiveness of introduced model. For second test, LMM is applied to healthcare databases due to the complex nature and the size of these heterogeneous databases, where extracting useful knowledge from these sources is usually a hectic task and difficult to achieve with traditional methods (e.g., SQL queries). Statistical models are methodical for this type of tasks involving categorizing patients based on their symptoms, diagnosis of a disease, etc. Analysis of this information provides useful insights for clinical decision support [151]. Obviously, when it comes to model-based clustering using mixture models, Gaussian mixture model is inevitable. For example, [152] demonstrates an efficient implementation using GMM to cluster patients using a systematic prediction of the surviving probability based on the duration of their stays in the hospital. In our case, we use BLMM for medical data analysis. BLMM is applied to 10 different datasets and the evaluation successfully demonstrated the efficiency of our model to learn the distinct patterns in the data which help to predict different diseases.

In addition to the healthcare datasets, we analyze the efficiency of our model for image clustering and extend it for image retrieval tasks as well because they are considered to be the most essential part of image analysis tasks like object recognition, security and categorization of medical images [153–158]. We have seen that wavelet transform can be implied efficiently for image and video processing [159]. A lot of research has been done to apply wavelet transform for feature extraction from images in wavelet domain, which is further used for image clustering and CBIR [159–164]. The wavelet coefficients are heavily tailed marginal distributions as they have very sparse data due to their energy packing property [159, 162, 165–167]. The literature suggests that using mixture models in the wavelet domain gives good results for image clustering and CBIR [159, 162, 163, 168]. In these cases, GMM has been used to track the peaks of the

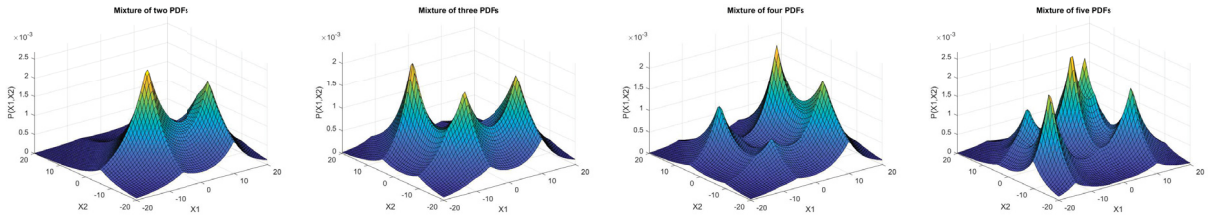


Figure 3.1: Example demonstrating mixtures of different number of Laplace distributions for two dimensional data

distributions. However, it is more efficient to use LMM for applications related to wavelet domain. This has been used as a worthy model for feature extraction for image and video clustering and retrieval [159]. In this chapter we introduce BLMM as a preprocessing model for multiple computer vision tasks such as image clustering and classification, categorization of databases and CBIR. This is done by using BLMM for modeling the wavelet domain representation of image to extract features. The features being derived from adapted parameters of BLMM for each image at different levels of decomposition, it would be an interesting idea to use BLMM for clustering and CBIR. Our proposed model hence uses BLMM to represent the feature space and then uses this representation to perform image clustering and retrieval. The proposed framework is validated through several experiments on texture data for feature extraction, image clustering and retrieval and detection results have demonstrated effectiveness of BLMM as compared to LMM for similar settings. The experiments are performed by using 5 texture images datasets and three different similarity measures are introduced for image retrieval, where a closed form solution is provided for one of the similarity measures in the context of our model.

3.2 Bounded Support Laplace Mixture Model

In this section, BLMM is presented which is an extension of LMM to improve the modeling capabilities of mixture model based on Laplace distribution. The idea behind bounded support mixture model is the fact that data in most of the applications exist in bounded support range and it is more appropriate to introduce a model with bounded support distributions. The parameters estimation for BLMM is performed by maximum likelihood approach, with EM and Newton Raphson method for optimization of estimated parameters. Before going further to explain the proposed model, basic formulation of mixture of Laplace distributions is presented.

3.2.1 Mixture of Laplace Distributions

LMM was introduced as generalization to k-median algorithm, which exhibits mixture based clustering with distributions relying on the median [52–54]. If a uni-variate random variable X, follows a K component mixture distribution which is represented by Eq. (1.1), then Laplace distribution for each component of mixture model can be represented as follows:

$$f(X|\xi_j) = \frac{1}{2b_j} \exp\left[-\frac{|X-\mu_j|}{b_j}\right] \tag{3.1}$$

where μ_j and b_j are mean and scale parameters of Laplace distribution for each component of mixture model. For estimation of parameters in Laplace mixture model, ML approach via EM algorithm gives a closed for solution for all parameters of mixture model as follows:

$$\hat{\mu}_j = \frac{\sum_{i=1}^{N} \frac{p(j|X_i)X_i}{|X_i-\mu_j|}}{\sum_{i=1}^{N} \frac{p(j|X_i)}{|X_i-\mu_j|}} \tag{3.2}$$

$$\hat{b}_j = \frac{\sum_{i=1}^{N} p(j|X_i)\,|X_i-\mu_j|}{\sum_{i=1}^{N} p(j|X_i)} \tag{3.3}$$

$$\hat{p}_j = \frac{1}{N}\sum_{i=1}^{N} p(j|X_i) \tag{3.4}$$

where $p(j|X_i)$ is posterior probability estimated for LMM and N represents the total number of observations of data. Some examples of data modeling via Laplace mixture for two dimensional data for number of components of mixture model are shown in Fig. (3.1).

3.2.2 Mixture of Bounded Laplace Distributions for Multidimensional Data

For BLMM, the term $p(\vec{X}|\xi_j)$ in Eq. (1.1) represents the bounded Laplace distribution (BLD), which is introduced to improve the data modeling capabilities associated with unbounded support range in Laplace distribution. The introduced BLD has the ability to model different shapes of observed data. An indicator function is presented which serves the purpose to define the boundary conditions for BLD. Bounded support region ∂ is presented in $\mathbb{R}$ and applying this indicator function in unbounded distribution defines the bounded support distribution. For each component j in the mixture model, indicator function is defined as $\mathrm{H}(\vec{X}|j)$ similar to its uni-variate counterpart given in Eq. (1.4). If we apply the indicator function $\mathrm{H}(\vec{X}|j)$, in unbounded support distribution (Laplace distribution in this case), the term $p(\vec{X}|\xi_j)$ in Eq. (1.1) is referred as bounded support

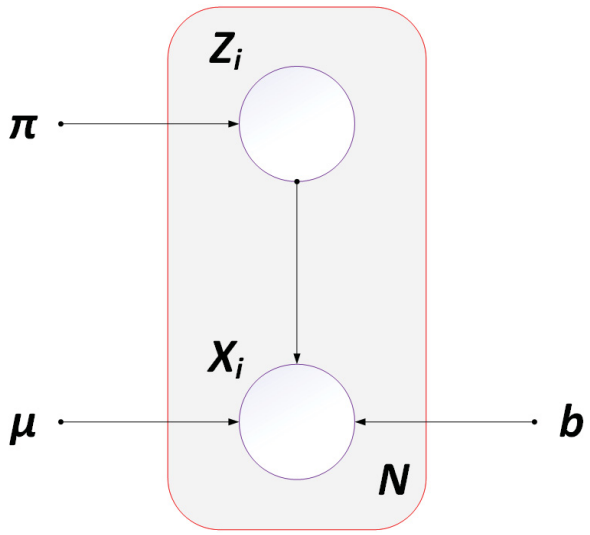


Figure 3.2: Graphical representation of Laplace mixture model

distribution (BLD in this case), for the vector $\vec{X}$ and according to Eq. (1.6), it is defined as:

$$p(\vec{X}|\xi_j) = \frac{f(\vec{X}|\xi_j)\mathrm{H}(\vec{X}|j)}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}} \tag{3.5}$$

where the term $f(\vec{X}|\xi_j)$ is regarded as Laplace distribution for $D-$dimensional vector $\vec{X}$:

$$f(\vec{X}|\xi_j) = \prod_{d=1}^{D} \frac{1}{2b_{jd}} \exp\left[-\frac{|X_d - \mu_{jd}|}{b_{jd}}\right] \tag{3.6}$$

In Eq. (3.5), $\xi_j = (\vec{\mu}_j, \vec{b}_j)$ represents the set of parameters of Laplace distribution with $\vec{\mu}_j = (\mu_{j1}, ..., \mu_{jD})$ and $\vec{b}_j = (b_{j1}, ..., b_{jD})$ as mean and scale parameters of D-dimensional bounded Laplace distribution, respectively [159]. The term $\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}$ presented in Eq. (3.5) defines the normalization constant that indicates the share of $f(\vec{X}|\xi_j)$ which belongs to the support region ∂. Let the input be set of features of data represented as $\mathscr{X} = (\vec{X}_1, ..., \vec{X}_N)$. With a mixture of K BLDs, the probability of data $\mathscr{X}$ can be modeled by a mixture of K BLDs as given in Eq. (1.2), where Θ represents the parameters of mixture model having K classes as $\Theta = (\xi_1, \xi_2, \xi_3)$, with $\xi_1 = (\vec{\mu}_1, ..., \vec{\mu}_K)$, $\xi_2 = (\vec{b}_1, ..., \vec{b}_K)$, and $\xi_3 = (p_1, ..., p_K)$. Missing group indicator vectors, $\vec{Z}_i = (Z_{i1}, ..., Z_{iK})$ can be introduced in complete likelihood of the data, where one vector is dedicated for each observation of data. These missing group vectors are also termed as membership vectors, which are used to encode the membership of each data vector for relative component of mixture model. For each membership vector, the unobserved variable Z_{ij} is equal to 1 if $\vec{X}_i$ belong to class j and 0, otherwise. The complete data likelihood after introducing the missing group

indicator variable is given below.

$$p(\mathscr{X},\mathscr{Z}|\Theta)=\prod_{i=1}^{N}\prod_{j=1}^{K}\Big(p(\vec{X}_i|\xi_j)p_j\Big)^{Z_{ij}} \tag{3.7}$$

The missing group variable Z_{ij} can be substituted by its expectation, where the expectation is termed as posterior probability. The expectation of unobserved variable Z_{ij} means that ith observation of data arises from jth component of the mixture model and it can be written as:

$$\hat{Z}_{ij}=p(j|\vec{X}_i)=\frac{p(\vec{X}_i|\xi_j)p_j}{\sum_{j=1}^{K}p(\vec{X}_i|\xi_j)p_j} \tag{3.8}$$

The complete set of vectors defining the membership of each observation of data into different components of mixture model is represented as: $\mathscr{Z}=\{\vec{Z}_1,...,\vec{Z}_N\}$.

3.2.2.1 Parameters Learning

In a mixture model, the parameter estimation is considered to be a very important step and in BLMM, parameters are estimated by maximum likelihood approach. The maximization of log-likelihood is similar to maximization of likelihood and for mathematical convenience, we consider the log-likelihood function. For parameter estimation using this approach, we suppose that we know the number of components (K) of mixture model. The maximum likelihood approach is to get the parameters of mixture model that maximizes the log-likelihood function given as:

$$\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)=\sum_{i=1}^{N}\sum_{j=1}^{K}\hat{Z}_{ij}\log\Big(p(\vec{X}_i|\xi_j)p_j\Big) \tag{3.9}$$

$$\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)=\sum_{i=1}^{N}\sum_{j=1}^{K}\hat{Z}_{ij}\times\left\{\log p_j+\log f(\vec{X}_i|\xi_j)+\log \mathrm{H}(\vec{X}_i|\partial_j)-\log\int_{\partial_j}f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right\} \tag{3.10}$$

For the computation of the parameters of mixture, log-likelihood of data is required to be maximized with respect to each parameter of mixture model. It is achieved by taking the derivatives of the log-likelihood with respect to p_j, μ_j, and b_j separately and equating them to zero for getting the estimated values of the parameters. The estimation of mixing parameter is provided in Section 2.2.3.1. The parameter estimation for mean and scale parameters of BLMM is presented in the following subsections.

3.2.2.2 Mean parameter estimation

In order to estimate the mean μ_{jd}, log-likelihood function given in Eq. 3.10 is considered which is differentiated with respect to mean parameter to achieve maximization of log-likelihood with respect to $\vec{\mu}_j$ as given Appendix B.1. For the estimated value of mean parameter, derivative of log-likelihood is set to zero as given below:

$$\frac{\partial \mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta)}{\partial \mu_{jd}} = 0 \tag{3.11}$$

The estimation of mean parameter yields a closed form solution (Appendix B.2) and the term for estimated value of mean parameter is represented as follows:

$$\hat{\mu}_{jd} = \frac{\sum_{i=1}^{N} \hat{Z}_{ij} \left\{ \left[\frac{\mathrm{X}_{id}}{b_{jd}|\mathrm{X}_{id}-\mu_{jd}|} \right] - \frac{\int_{\partial_j} \left(f(\vec{\mathrm{u}}|\xi_j) \left[\frac{(\mathrm{u}-\mu_{jd})}{b_{jd}|\mathrm{u}-\mu_{jd}|} \right] \right) d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \right\}}{\sum_{i=1}^{N} \left[\frac{\hat{Z}_{ij}}{b_{jd}|\mathrm{X}_{id}-\mu_{jd}|} \right]} \tag{3.12}$$

The term $\int_{\partial_j} f(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})/b_{jd}\left|\mathrm{u}-\mu_{jd}\right| d\mathrm{u}$ in Eq. (3.12) represents the expectation of term $(\mathrm{u}-\mu_{jd})/b_{jd}\left|\mathrm{u}-\mu_{jd}\right|$ under the probability distribution $f(\mathrm{u}|\xi_j)$, which is approximated as:

$$\int_{\partial_j} \left(f(\vec{\mathrm{u}}|\xi_j) \left[\frac{(\mathrm{u}-\mu_{jd})}{b_{jd}\left|\mathrm{u}-\mu_{jd}\right|} \right] \right) d\mathrm{u} \approx \frac{1}{M} \sum_{m=1}^{M} \left[\frac{(\mathrm{s}_{m_{jd}}-\mu_{jd})}{b_{jd}\left|\mathrm{s}_{m_{jd}}-\mu_{jd}\right|} \right] \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \tag{3.13}$$

where $\mathrm{s}_{m_{jd}} \sim f(\mathrm{u}|\xi_j)$ is a set of random variables, which is drawn from the Laplace distribution for the particular component j of the mixture model. The set of data with random variables drawn from Laplace distribution have M vectors with D dimensions. M is a large integer chosen to generate the set of random variables. Similarly, the term $\int_{\partial_j} f(\mathrm{u}|\xi_j) d\mathrm{u}$ in Eq. (3.12) can be approximated as:

$$\int_{\partial_j} f(\mathrm{u}|\xi_j) d\mathrm{u} \approx \frac{1}{M} \sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \tag{3.14}$$

By replacing the approximated values from Eqs. (3.13 & 3.13) in Eq. (3.12), we get the following expression for estimated value of mean parameter.

Algorithm 3 Model Learning for BLMM

Input:Dataset $\mathscr{X} = \{\vec{X}_1, \ldots, \vec{X}_N\}$, t_{min}.
Output: Θ, $\mathscr{Z}$.
{Initialization}:
  K-Means Algorithm (Computation of $\vec{\mu}_1, \ldots, \vec{\mu}_K$ & cluster assignment)
    for all $1 \leq j \leq K$ do
      Computation of p_j
      Computation of $\vec{b}_K$
    end for
{Expectation Maximization}:
while relative change in log-likelihood $\geq t_{min}$ do
  {[**E Step**]}:
    for all $1 \leq j \leq K$ do
      Compute $p(j|\vec{X}_i)$ for $i = 1, \ldots, N$. using Eq. (3.8).
    end for
  {[**M step**]}:
    for all $1 \leq j \leq K$ do
      Estimation of mixing parameter p_j using Eq. (2.12).
      Estimation of mean $\vec{\mu}_j$ using Eq. (3.12).
      Estimation of scale parameter $\vec{b}_j$ using Eq. (3.20).
    end for
end while

3.2.2.3 Scale parameter estimation

In order to estimate the scale parameter $\hat{b}_{jd}$, log-likelihood function given in Eq. 3.10 is considered, which is differentiated with respect to scale parameter for achieving an expression for scale parameter estimate via maximum likelihood. The first derivative does not provide a closed form solution for the estimate of scale parameter and we need to apply Newton-Raphson method for computation of scale parameter which also require the second derivative of the log-likelihood with respect to scale parameter. The procedure to compute the first and second derivative of the log-likelihood with respect to scale parameter is given in Appendix B.3 & B.4 and final expressions are represented as follows:

$$\frac{\partial \mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta)}{\partial b_{jd}} = \sum_{i=1}^{N} \hat{Z}_{ij} \times \left\{ \left[\frac{-1}{b_{jd}} + \frac{|\mathrm{X}_{id} - \mu_{jd}|}{b_{jd}^2} \right] - \frac{\int_{\partial_j} \left(\frac{-1}{b_{jd}} f(\vec{\mathrm{u}}|\xi_j) + \frac{|\mathrm{u} - \mu_{jd}|}{b_{jd}^2} f(\vec{\mathrm{u}}|\xi_j) \right) d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \right\} \tag{3.15}$$

$$\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial b_{jd}^2} = \sum_{i=1}^{N} \hat{Z}_{ij} \left\{ \left[\frac{1}{b_{jd}^2} - \frac{2\left|\mathrm{X}_{id}-\mu_{jd}\right|}{b_{jd}^3} \right] \right. \tag{3.16}$$

$$- \frac{\int_{\partial_j} \left(\frac{1}{b_{jd}^2} f(\vec{\mathrm{u}}|\xi_j) - \frac{1}{b_{jd}} f(\vec{\mathrm{u}}|\xi_j) \frac{(\mathrm{u}-\mu_{jd})}{b_{jd}|\mathrm{u}-\mu_{jd}|} \right) d\mathrm{u}}{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right)}$$

$$+ \frac{\left(\int_{\partial_j} \frac{-1}{b_{jd}} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right) \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) \frac{(\mathrm{u}-\mu_{jd})}{b_{jd}|\mathrm{u}-\mu_{jd}|} d\mathrm{u}}{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right)^2} -$$

$$\frac{\left(\int_{\partial_j} \frac{-2|\mathrm{u}-\mu_{jd}|}{b_{jd}^3} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} + \int_{\partial_j} \left(\frac{-|\mathrm{u}-\mu_{jd}|}{b_{jd}^3} f(\vec{\mathrm{u}}|\xi_j) + \frac{|\mathrm{u}-\mu_{jd}|^2}{b_{jd}^4} f(\vec{\mathrm{u}}|\xi_j) \right) d\mathrm{u} \right)}{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right)}$$

$$\left. + \frac{\left(\int_{\partial_j} \frac{|\mathrm{u}-\mu_{jd}|}{b_{jd}^2} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right) \int_{\partial_j} \left(\frac{-1}{b_{jd}} f(\vec{\mathrm{u}}|\xi_j) + \frac{|\mathrm{u}-\mu_{jd}|}{b_{jd}^2} f(\vec{\mathrm{u}}|\xi_j) \right) d\mathrm{u}}{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right)^2} \right\}$$

In Eq. (3.16), the term $\int_{\partial_j} f(\mathrm{u}|\xi_j)\left|\mathrm{u}-\mu_{jd}\right|^2 dx$ can be approximated as below:

$$\int_{\partial_j} f(\mathrm{u}|\xi_j)\left|\mathrm{u}-\mu_{jd}\right|^2 dx \approx \frac{1}{M} \sum_{m=1}^{M} \left|\mathrm{s}_{m_{jd}}-\mu_{jd}\right|^2 \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \tag{3.17}$$

where $\mathrm{s}_{m_{jd}} \sim f(\mathrm{u}|\xi_j)$ is a set of random variables drawn from the Laplace distribution for particular component j of the mixture model. The rest of the approximations are followed from the estimation of mean. After applying these approximations, first and second derivatives are represented as follows:

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial b_{jd}} = \sum_{i=1}^{N} Z_{ij} \left\{ \left[\frac{-1}{b_{jd}} + \frac{\left|\mathrm{X}_{id}-\mu_{jd}\right|}{b_{jd}^2} \right] - \right. \tag{3.18}$$

$$\left. \frac{\sum_{m=1}^{M} \frac{-1}{b_{jd}} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) + \sum_{m=1}^{M} \frac{\left|\mathrm{s}_{m_{jd}}-\mu_{jd}\right|}{b_{jd}^2} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j)}{\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j)} \right\}$$

$$\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial b_{jd}^2} = \sum_{i=1}^{N} Z_{ij} \left\{ \left[\frac{1}{b_{jd}^2} - \frac{2\left|\mathrm{X}_{id}-\mu_{jd}\right|}{b_{jd}^3} \right] \right. \tag{3.19}$$

$$- \frac{\left(\sum_{m=1}^{M} \frac{1}{b_{jd}} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) - \sum_{m=1}^{M} \frac{(\mathrm{s}_{m_{jd}}-\mu_{jd})}{b_{jd}^2\left|\mathrm{s}_{m_{jd}}-\mu_{jd}\right|} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \right)}{\left(\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \right)}$$

$$+ \frac{\left(\sum_{m=1}^{M} \frac{-1}{b_{jd}} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \right) \left(\sum_{m=1}^{M} \frac{(\mathrm{s}_{m_{jd}}-\mu_{jd})}{b_{jd}\left|\mathrm{s}_{m_{jd}}-\mu_{jd}\right|} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \right)}{\left(\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \right)^2}$$

$$- \frac{\left(\sum_{m=1}^{M} \frac{-2\left|\mathrm{s}_{m_{jd}}-\mu_{jd}\right|}{b_{jd}^3} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) + \sum_{m=1}^{M} \frac{-\left|\mathrm{s}_{m_{jd}}-\mu_{jd}\right|}{b_{jd}^3} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \right)}{\left(\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \right)}$$

$$- \frac{\left(\sum_{m=1}^{M} \frac{\left|\mathrm{s}_{m_{jd}}-\mu_{jd}\right|^2}{b_{jd}^4} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \right)}{\left(\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \right)} + \frac{\left(\sum_{m=1}^{M} \frac{\left|\mathrm{s}_{m_{jd}}-\mu_{jd}\right|}{b_{jd}^2} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \right)}{\left(\sum_{m=1}^{M} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \right)^2} \times$$

$$\left. \left(\sum_{m=1}^{M} \frac{-1}{b_{jd}} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) + \sum_{m=1}^{M} \frac{\left|\mathrm{s}_{m_{jd}}-\mu_{jd}\right|}{b_{jd}^2} \mathrm{H}(\mathrm{s}_{m_{jd}}|\partial_j) \right) \right\}$$

It is observed from Eq. (3.15), that first derivative is non-linear and does not provide any closed form in the maximization of log-likelihood, we apply Newton-Raphson method for the estimation of $\hat{b}_j$ as follows:

$$\hat{b}_{jd} \simeq b_{jd} - \left[\left(\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)]}{\partial b^2{}_{jd}} \right)^{-1} \left(\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial b_{jd}} \right) \right] \tag{3.20}$$

The complete learning of BLMM is given in Algorithm 3, where t_{min} is minimum threshold used to monitor the convergence criteria in each iteration. In the initialization phase, K-Means is applied for computation of mean and data assignment in each cluster. This information is further used for computation of scale parameter and mixing weights during initialization phase.

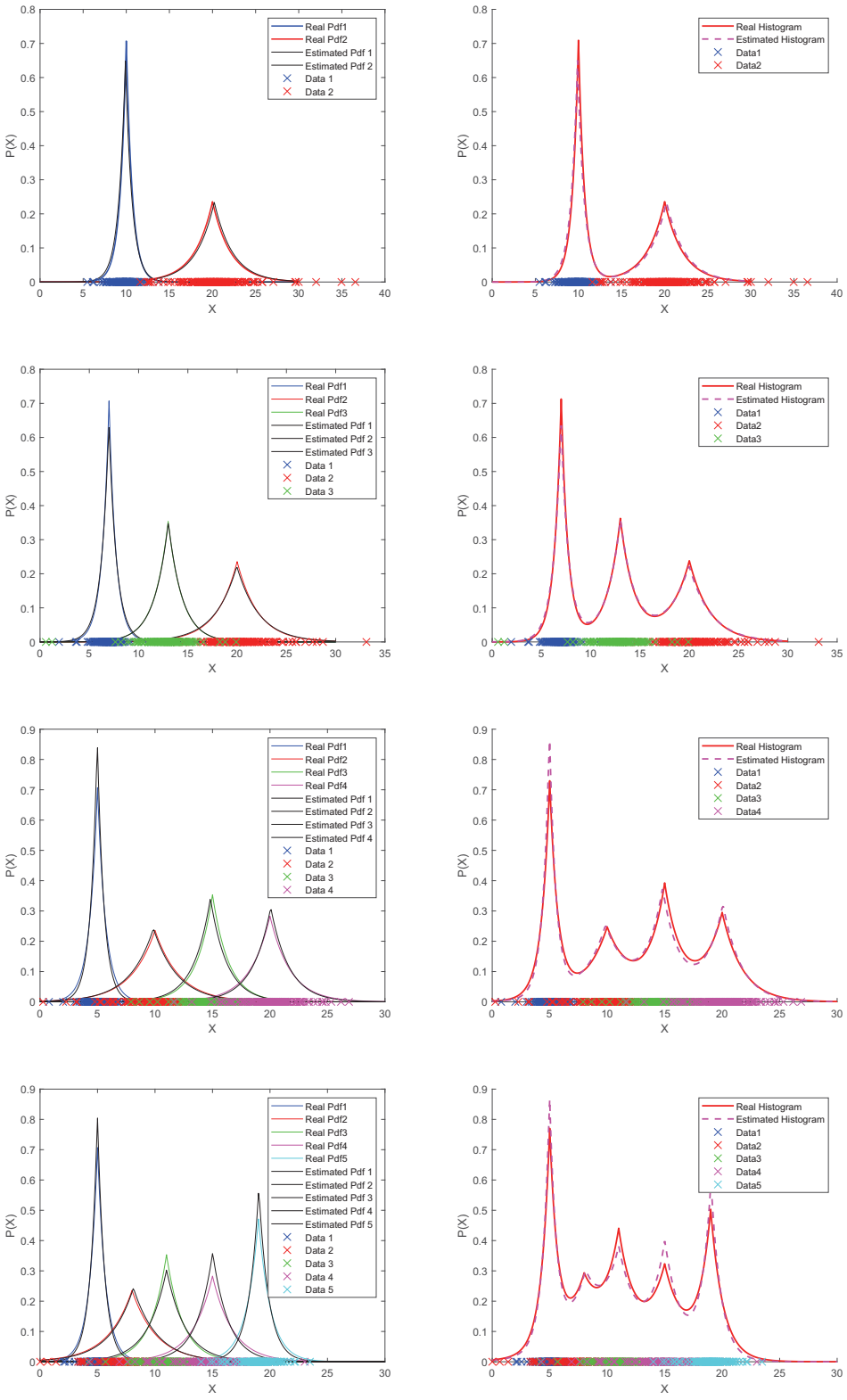


Figure 3.3: Examples demonstrating real and estimated components of mixtures of Bounded Laplace distributions via one dimensional artificial histograms

3.3 Proof of concept through experiments on Synthetic Data Clustering

In this section, the proposed algorithm is validated to perform clustering on synthetic data. The consideration of synthetic data for clustering is very important to examine the performance of proposed model because the distribution of generated data is known and it can be observed after learning through the proposed model. In the first subsection, experiments on one dimensional synthetic data are presented whereas in subsection two, experiments on multi-dimensional synthetic data are provided. The data are generated with different parameters using Laplace with 2,3,4 and 5 components for uni-variate and multi-variate case.

3.3.1 One-dimensional data

We generate single dimensional data from an artificial mixture model based on Laplace distribution and evaluate the performance of our model on this data. 4 different datasets are used with increasing number of components starting from 2 to 5. The first set contains 2 components with 200 data points each and rest of the datasets have 150 data points belonging each of its components. Table 3.1 shows the real values of the parameters and the ones estimated by our model. We can clearly see that the estimated parameters are so close to the real values. Our model was able to maintain its accuracy even with increasing number of components. This is also depicted in Fig. (3.3) which shows the histograms of the real and estimated histograms of the PDFs. The similarities between the two histograms depict the efficiency of our model. It is to be noted that we use five components only for ease of representation. Our model was capable of achieving similar accuracies with more components as well.

3.3.2 Multidimensional data

Like our experiment with one-dimensional data, we create two-dimensional datasets to test the performance of our model for the multivariate case. This experiment involves four datasets with 2, 3, 4 and 5 components respectively with each component having 200 data points assigned to them. Table 3.2 shows the results we obtained with these datasets comparing the real parameter values of each dimension with the estimated parameters. The results reflect the fact that our model is very stable even in the multivariate case. It is also evident from the histograms presented in Figs. (3.4 & 3.5) that the estimation of the parameters by our model is quite accurate.

Table 3.1: Real and estimated parameters of different datasets. N denotes the total number of data points, N_j denotes the number of data points in the cluster j. Here μ_j, b_j and π_j are the real parameters and $\hat{\mu}_j, \hat{b}_j$ and $\hat{\pi}_j$ are the parameters estimated by our proposed model.

Data set	N_j	j	μ_j	b_j	π_j	$\hat{\mu}_j$	$\hat{b}_j$	$\hat{\pi}_j$
D1	200	1	10	0.7071	0.5	10.04	0.7216	0.5094
($N = 400$)	200	2	20	2.1213	0.5	19.92	2.1327	0.4906
D2	150	1	7	0.7071	0.33	6.988	0.7324	0.3373
($N = 450$)	150	2	20	2.1213	0.33	19.965	2.22	0.3340
	150	3	13	1.4142	0.33	13.015	1.36	0.3286
	150	1	5	0.7071	0.25	4.9802	0.715	0.2431
D3	150	2	10	2.1213	0.25	9.8625	2.0658	0.2477
($N = 600$)	150	3	15	1.4142	0.25	14.8156	1.4623	0.2516
	150	4	20	1.7678	0.25	20.0641	1.695	0.2576
	150	1	5	0.7071	0.20	4.9927	0.7233	0.2112
D4	150	2	8	2.1213	0.20	8.0509	2.1516	0.2159
($N = 750$)	150	3	11	1.4142	0.20	11.004	1.3877	0.2043
	150	4	15	1.7678	0.20	15.0124	1.7096	0.1679
	150	5	19	1.0607	0.20	19.025	0.966	0.2007

Table 3.2: Real and estimated parameters of different datasets. N denotes the total number of data points, N_j denotes the number of data points in the cluster j. Here $\mu_{j1}, \mu_{j2}, b_{j1}, b_{j2}$ and π_j are the real parameters and $\hat{\mu}_{j1}, \hat{\mu}_{j2}, \hat{b}_{j1}, \hat{b}_{j2}$ and $\hat{\pi}_j$ are the parameters estimated by our proposed model.

Data set	N_j	j	μ_{j1}	μ_{j2}	b_{j1}	b_{j2}	π_j	$\hat{\mu}_{j1}$	$\hat{\mu}_{j2}$	$\hat{b}_{j1}$	$\hat{b}_{j2}$	$\hat{\pi}_j$
D1	200	1	2	1	1.4142	1.1314	0.50	2.04	1.05	1.3570	1.1079	0.4992
($N = 400$)	200	2	-2	-3	0.7071	0.7071	0.50	-1.98	-2.95	0.7277	0.7105	0.5008
D2	200	1	2	1	1.4142	1.1314	0.33	1.9995	1.0157	1.3543	1.0855	0.3304
($N = 600$)	200	2	-2	-3	1.0607	0.7071	0.33	-1.917	-3.048	0.9629	0.7212	0.3230
	200	3	-4	-4	1.0607	0.7071	0.33	-4.0454	-4.0595	0.9148	0.7109	0.3466
	200	1	2	1	1.4142	1.1314	0.25	1.9756	1.0281	1.3216	1.0163	0.2431
D3	200	2	-2	-3	1.0607	0.7071	0.25	-2.0518	-3.0490	0.9151	0.7012	0.2406
($N = 800$)	200	3	-4	-4	1.0607	0.7071	0.25	-4.0803	-3.9656	0.9213	0.7280	0.2592
	200	4	4	-3	0.7071	0.7071	0.25	3.9339	-2.9354	0.7220	0.7197	0.2570
	200	1	2	1	1.4142	1.1314	0.20	1.9790	1.0140	1.2655	1.1215	0.1942
D4	200	2	-2	-3	1.0607	0.7071	0.20	-1.9671	-2.9136	0.9411	0.7349	0.2190
($N = 1000$)	200	3	-4	4	1.0607	0.7071	0.20	-3.9218	3.9118	0.9652	0.7299	0.1975
	200	4	4	-3	0.7071	0.7071	0.20	3.9829	-3.0609	0.7185	0.6803	0.2025
	200	5	3	-1.5	0.7071	0.7071	0.20	3.0590	-1.5165	0.7279	0.6719	0.1868

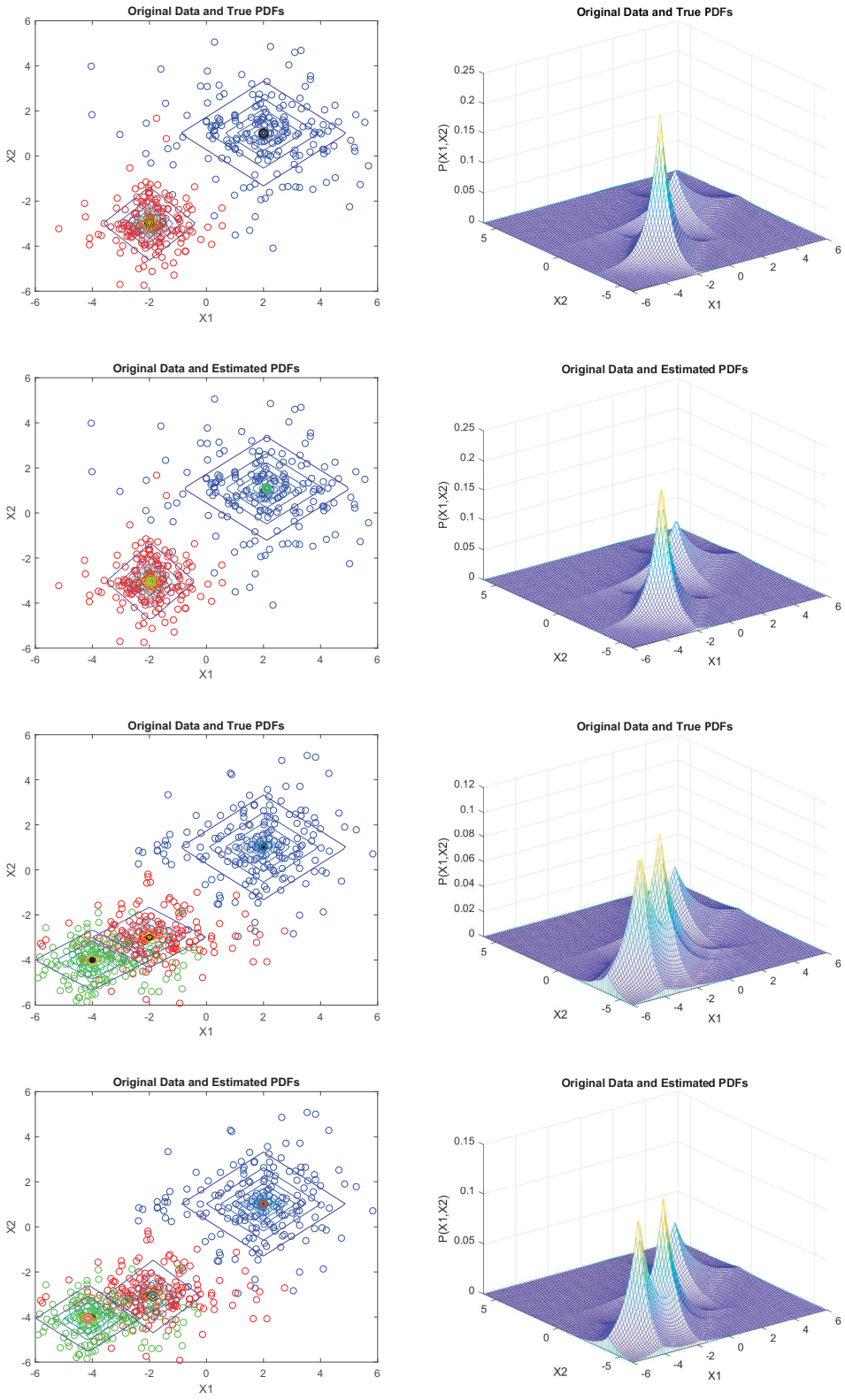


Figure 3.4: Examples demonstrating real and estimated components of mixtures of Bounded Laplace distributions via two dimensional artificial histograms

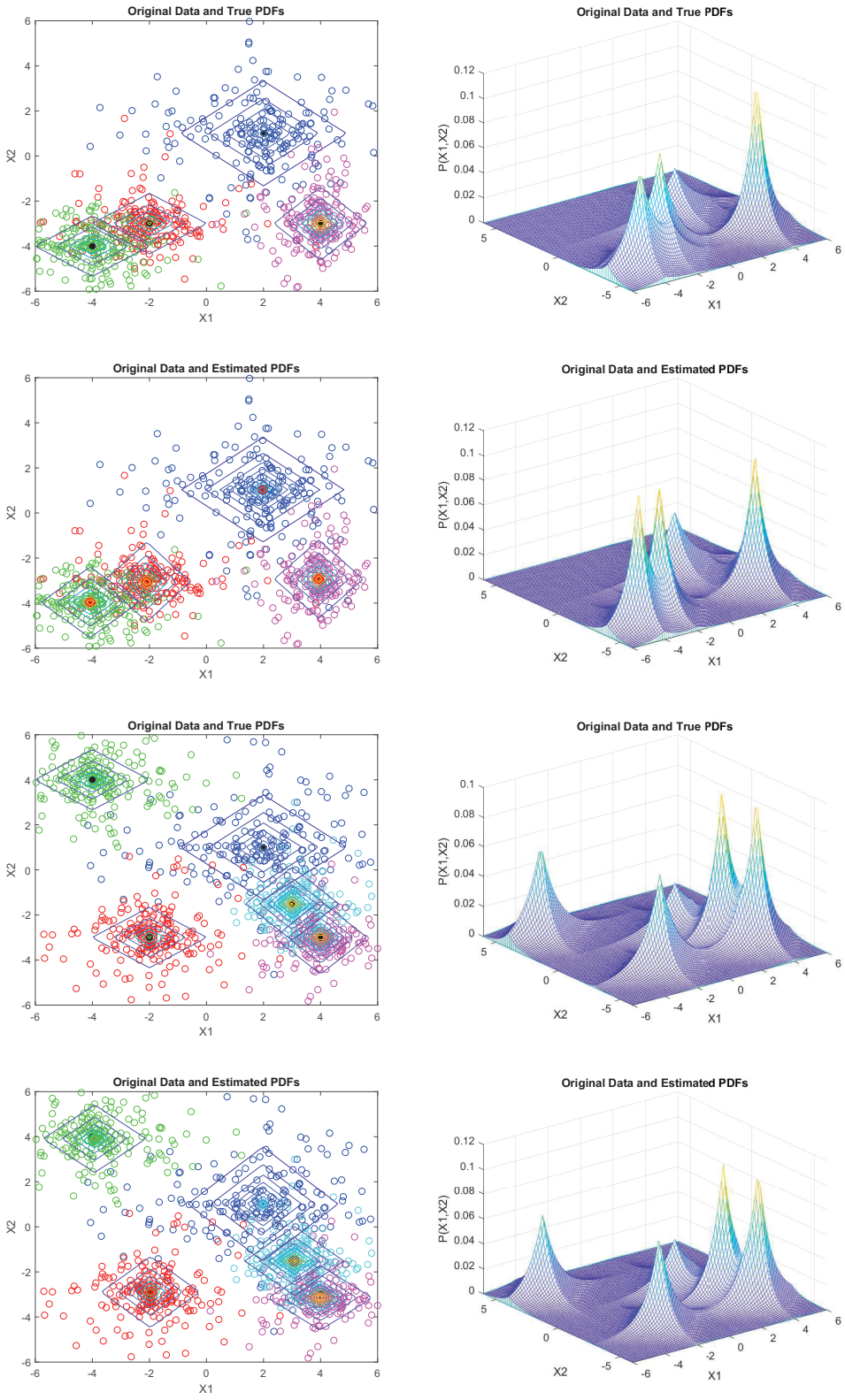


Figure 3.5: Examples demonstrating real and estimated components of mixtures of Bounded Laplace distributions via two dimensional artificial histograms (continued)

3.4 Proof of concept through experiments on medical data clustering

Clustering of medical data tends to be a promising application as it helps in medical diagnosis. We use our model on some real medical datasets. First, we list out the different datasets we evaluate our model upon and then we discuss the performance of our model.

The Cryotherapy dataset comprises data collected form a dermatology clinic in Mashhad with data from patients reporting for warts. The dataset has 7 features and 90 instances. The objective is to identify if the treatment was effective or not. Statlog (Heart) is a dataset which contains sample data from 270 patients. The dataset has 13 attributes and the purpose here is to predict if the patient has a heart disease or not. The Parkinsons dataset consists of data obtained from the speech signals of 31 people among which 23 suffer from Parkinson's disease. The dataset has 195 instances with 23 features. The aim is to differentiate between the voices of patients with Parkinson's disease and the ones who are healthy. Haberman's survival dataset contains data recorded for 306 patients who have undergone surgery for breast cancer. It has 3 attributes and the goal is to cluster between patients who survived for more than 5 years and less than 5 years. The breast cancer Coimbra dataset includes 116 data samples comprising 64 samples from patients with breast cancer and 52 samples from healthy people. Our objective is to distinguish Cancer patients from the healthy people. This dataset has 10 attributes and is one of the latest datasets in the field. The Immunotherapy dataset is part of the data collected from the dermatology clinic in Mashhad as well, except that the treatment method followed here is immunotherapy and the number of attributes here is 8. Like the Cryotherapy dataset the intent is to identify if the treatment was effective or not. Mammographic mass dataset consists of 961 instances and 6 attributes and is used to predict the severity of the breast cancer in the patient. The two classes are benign and malignant. Hence our model must cluster the data points between these two classes. The Blood transfusion service center dataset contains data collected from 748 people from the transfusion center in Hsin-Chu city in Taiwan. The data has 5 attributes with the target being identification of whether the person donated blood or not. The Fertility dataset comprises data samples taken from 100 volunteers. The dataset has 10 attributes which are mostly external features like, fevers, trauma, smoking, etc rather than medical analysis and the intention is to identify if the person is fertile. SPECTF heart dataset contains data obtained from Single Proton Emission Computed Tomography (SPECT) images from 267 samples. The data has 44 dimensions and our aim is to cluster between the normal and abnormal classes [122–130].

We compare the results we obtained with our model, with Laplace Mixture Models (LMM) and K-means which is a robust clustering model. Table 3 shows the comparison between the

Table 3.3: Clustering Accuracy for different Medical Datasets

Data set	Dimension	Samples	Classes	Accuracy (%)		
				BLMM	LMM	K-Means
Cryotherapy	7	90	2	**90:00**	86.67	78.89
Statlog (Heart)	13	270	2	**79.63**	76.30	61.85
Parkinsons	23	195	2	**74.36**	73.33	69.23
Haberman	3	306	2	**74.83**	61.76	50:00
Breast Cancer	10	116	2	**60.34**	53.45	50.86
Immunotherapy	8	90	2	**72.22**	65.56	53.33
Mammographic	6	961	2	**76.99**	74.10	68.55
Transfusion	5	748	2	**73.93**	66.98	61.10
Fertility	10	100	2	**62:00**	60:00	57:00
SPECTF Heart	44	267	2	**68.54**	67.79	62.55

models. Our model outperformed the other models by a very good margin in most of the datasets. For example, in Haberman dataset, the increase in accuracy when compared to LMM and K-means is around 13 and 24 percent respectively; Similarly, in the breast cancer dataset, the increase is around 7 and 10 percent respectively. The overall accuracy with our model is higher with all the datasets when compared to LMM and K-means.

3.5 Application of BLMM in Image Clustering and CBIR

3.5.1 Proposed Framework for Image Clustering and CBIR

Image clustering is the process of categorizing images into one of the predefined groups which is further applied in many important applications [169]. Texture features in images provide very interesting information which plays an important role in image categorization in many computer vision and image processing applications [170]. Several applications of texture categorization include material classification, object recognition, scene classification analyzing biomedical images for computer aided diagnostics [170–172]. In order to validate the performance of proposed BLMM, it is applied to feature extraction for texture images in wavelet domain, texture image clustering and CBIR in texture images databases. The performance of BLMM is compared with LMM in all the these scenarios of texture images. In the following subsections, the proposed feature extraction for texture images in wavelet domain, image categorization and CBIR is explained. The proposed model and application framework are validated through a set of experiments in each scenario. For CBIR, City-block distance, posterior probability and Kullback-Leibler Divergence are applied and closed form solution for Kullback-Leibler Divergence is proposed for CBIR. In order

to demonstrate the contribution of BLMM for this application, complete framework is presented in Fig. (3.6).

3.5.2 Discrete Wavelet Transform

We present the properties of the discrete wavelet transform which are appropriate to image processing, so we will talk about two-dimensional (2-D) discrete wavelet transform. The 2-D wavelet transform is an extension of 1-D wavelet transform using separable wavelet filters. In 1-D wavelet transform, a signal is passed through a lowpass and highpass filters, respectively, and then down sampled by a factor of two. The same process is repeated for each level of decomposition through wavelet transform and multiple levels also called scales are achieved by duplicating the filtering and decimation on the lowpass branch of the output only. This process is performed only for finite number of levels and the resulting coefficients are termed as wavelet coefficients. The 2-D transform is computed by applying the above described 1-D transform on all rows of the input and then redoing it on all the columns. After applying 2-D transform, an image is decomposed into four sub-bands representing the scale-down low resolution approximation of the image and horizontal, vertical and diagonal information. More details on discrete wavelet transform can be found in [173]. The wavelet transform has an important property of energy compaction of input into relatively small number of wavelet coefficients [173, 174]. After wavelet domain representation of images, much of the energy is concentrated into scale-down low resolution approximation of the original image. In addition, the energy in high frequency bands is also concentrated into a relatively small number of coefficients [173, 174] and it can be observed by the histogram representation of high frequency sub-bands in Fig. (3.7). From the studies and Fig. (3.7), it has been observed that distributions of wavelet coefficients in high frequency sub-bands have a Laplacian-like density [159, 173, 174]. In Fig. (3.7), histogram is shown for diagonal, horizontal and vertical subspaces of wavelet domain and HH means that highpass filter is applied horizontally and vertically, whereas HL means that highpass filter is applied horizontally and lowpass filter is applied vertically. Due to this peaky nature of distribution of wavelet coefficients, GMM and LMM have been proposed for modeling the data in wavelet domain [159, 162, 163]. We propose the application of BLMM in the same fashion for modeling the wavelet coefficients as described in literature for GMM and LMM.

3.5.3 Feature Extraction via BLMM from Wavelet subspaces

In the proposed feature extraction, we focus on multi-resolution representation of image feature in wavelet domain. Each image from a database is decomposed via 2-D discrete wavelet transform in

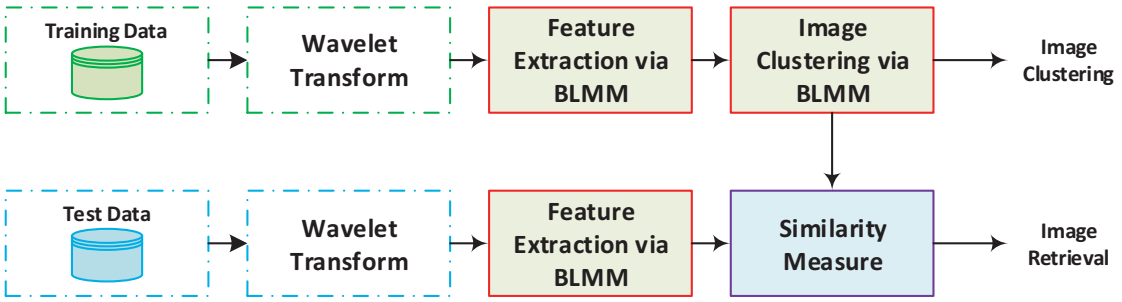


Figure 3.6: Framework for Feature Extraction, Image Clustering & Content Based Image Retrieval via BLMM

four wavelet subspaces at each level of decomposition. The wavelet coefficients in these subspaces, represent the image texture information and it is very important to apply an appropriate statistical model to represent this information in the feature extraction step. We apply BLMM for modeling the wavelet coefficients in high frequency sub-bands by using two components mixture model centered at 0. The parameters of BLMM obtained after modeling the wavelet coefficient are used as features for each image. The dimension of these feature vectors is very low, which make image clustering and retrieval less time consuming and it further enhances the user experience in the system where clustering and image retrieval is being used. If we assume that each wavelet subspace has N coefficients, and each coefficient is represented by W, then model with two components can represented as:

$$p(\mathscr{W}|\Theta) = \prod_{i=1}^{N} [p(W_i|0,b_s)p_s + p(W_i|0,b_l)p_l] \tag{3.21}$$

where p_s & p_l are mixing coefficients that sum to one and $p(W_i|0,b_s)$ & $p(W_i|0,b_l)$ is BLD defined by Eq. (3.5) with zero mean and scale parameters (b_s & b_l). In the modeling of wavelet coefficients, Θ is complete set of parameters to characterize each wavelet subspace defined as: $\Theta = (b_s, b_l, p_s, p_l)$. Since distributions are centered at 0, the shape of bounded Laplace distribution is determined by scale parameter b. The parameter estimation is performed in the same fashion as described in Section (3.2) and EM algorithm is applied to estimate the parameters. In the E-Step, posterior probability for each component is computed with respect to each wavelet coefficient in each subspace. The posterior probability is computed as follows:

$$P(s|W_i) = \frac{p(W_i|0,b_s)p_s}{p(W_i|0,b_s)p_s + p(W_i|0,b_l)p_l} \tag{3.22}$$

$$P(l|W_i) = \frac{p(W_i|0,b_l)p_l}{p(W_i|0,b_s)p_s + p(W_i|0,b_l)p_l} \tag{3.23}$$

The mixing parameters (p_s & p_l) and scale parameters (b_s & b_l) are computed in the M-Step along with Newton-Raphson method as follows:

$$\hat{p}_s = \frac{1}{N}\sum_{i=1}^{N} P(s|W_i) \;\; \& \;\; \hat{p}_l = \frac{1}{N}\sum_{i=1}^{N} P(l|W_i) \tag{3.24}$$

$$\hat{b}_s \simeq b_s - \left[\left(\frac{\partial^2 \log[p(\mathscr{W},\mathscr{Z}|\Theta)]}{\partial {b^2}_s}\right)^{-1}\left(\frac{\partial \log[p(\mathscr{W},\mathscr{Z}|\Theta)]}{\partial b_s}\right)\right] \tag{3.25}$$

$$\hat{b}_l \simeq b_l - \left[\left(\frac{\partial^2 \log[p(\mathscr{W},\mathscr{Z}|\Theta)]}{\partial {b^2}_l}\right)^{-1}\left(\frac{\partial \log[p(\mathscr{W},\mathscr{Z}|\Theta)]}{\partial b_l}\right)\right] \tag{3.26}$$

where derivative are computed following the Eqs. (3.18 & 3.19) with assumption that mean 0. In the next stage, model parameters of all wavelet subspaces are integrated to construct feature space for each image. All the images in the dataset are decomposed via 2-D discrete wavelet transform. The parameters for each detailed sub-band (horizontal, vertical and diagonal) at each wavelet scale are computed using BLMM via EM algorithm. Therefore, we will have four parameters represented by $[p_s, p_l, b_s, b_l]$, for every wavelet subspace. A scaling subspace is also generated at coarsest scale using 2-D wavelet transform, besides wavelet subspaces. The scaling subspace is a low-frequency approximation of the original image and mean value of its coefficients is also taken as a feature. Thus, the integrated feature space via BLMM for each image is expressed as follows:

$$\mathscr{F} = [F_{1H}, F_{1V}, F_{1D}, S_1, ..., F_{jH}, F_{jV}, F_{jD}, S_j] \tag{3.27}$$

where F is the feature set $[p_s, p_l, b_s, b_l]$ of wavelet subspaces and S is the mean value of the coefficients in the scaling subspace. The subscripts H,V, & D express horizontal, vertical and diagonal directions, respectively, at each scale and subscript j represents the number of decomposition scales in the image. The features vectors are composed of different dynamic ranges because they express different physical quantities. In the similarity calculation, the features with higher value will overshadow the features with lower values and therefore, features are normalized according to the procedure defined in [159, 175]. After the normalization each component of the feature vector will be emphasized equally. For the normalization, it is assumed that features are generated by Gaussian distribution and we compute the mean μ and standard deviation σ for each feature vector $\mathscr{F}$. Each feature vector is normalized as follows:

$$\mathscr{F}_i = \frac{\mathscr{F}_i - \mu}{\sigma} \tag{3.28}$$

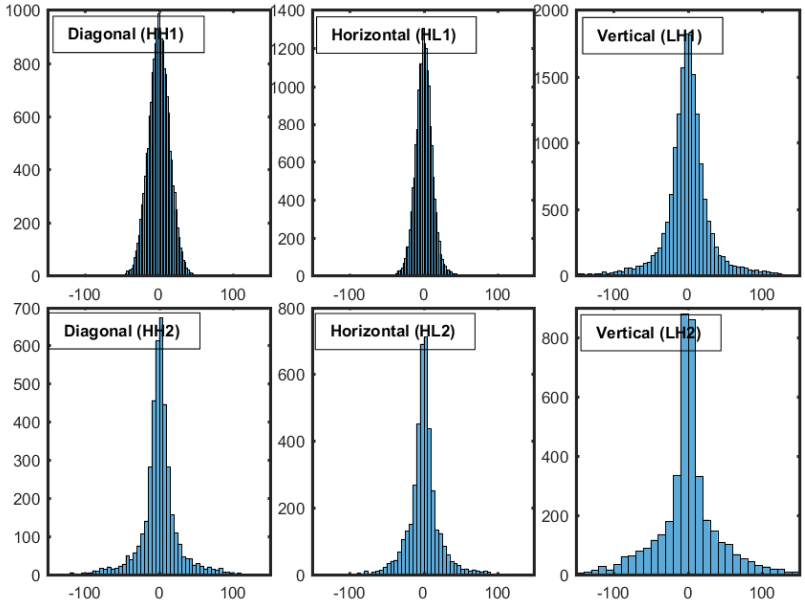


Figure 3.7: Wavelet coefficient at 2nd level of decomposition

Most of the values in the feature vectors will be mapped in the range $[-1, 1]$. Normalization is important to avoid biasing due to a few abnormal values occurring in the feature vectors [159, 175].

3.5.4 Image Clustering

The objective of image clustering is to assign a particular class to the unlabeled images, which can eventually categorize the images of a database into different classes and it can be extended to efficient CBIR [160, 161]. Texture image clustering is chosen to examine the performance of BLMM in data clustering. In the proposed framework for image clustering, texture images are transformed into feature space through wavelet domain modeling of images via BLMM before going further to actual image clustering task which is also achieved by BLMM. In this setting of image clustering framework, it is possible to examine the performance of BLMM for feature extraction and image clustering as described in Fig (3.6). The feature extraction process transforms each image into a D-dimensional feature vector, which can be applied to the image clustering stage to be modeled through BLMM described in Section (3.2). If we assume that a texture images dataset is composed of N images, where each is represented by a D-dimensional vector in the feature space, then image clustering task will require to estimate the parameters of multivariate BLMM with EM algorithm using the Eqs. (2.12,3.12 & 3.20) and estimating the cluster assignment from posterior probability via Eq. (3.8). The complete approach of image clustering via BLMM is described in Algorithm 3. In our application to demonstrate the effectiveness of proposed algorithm, texture image clustering

and CBIR are connected to each other, where CBIR adopts the trained model from clustering stage to perform retrieval which is described in the following subsection.

3.5.5 Content Based Image Retrieval

During the image clustering stage, BLMM categorizes the images into different groups called clusters and this trained model can be further used to perform image retrieval. This task relies on the trained model that characterizes and learn the primitive features of images used in the training process. These features are composed of information about shape, color and texture. In a broad manner, these features can be composed of visual and textual descriptors. Texture is a powerful and an important visual characteristic which is very difficult to define and even harder to model [159]. In order to perform CBIR, texture images databases are selected for demonstration of BLMM in data modeling. The images are categorized by applying clustering strategy as described in Subsection (3.5.4). For CBIR, a mean of the feature vectors of each cluster is computed and a similarity measure is computed with respect to feature vector of query image and mean of feature vectors of each cluster. Different similarity measures are presented below which perform the image retrieval via learned model based on BLMM.

3.5.5.1 Texture Image Retrieval via City-block distance

The City-block distance which is also termed as Manhattan distance can be used for computing the similarity measure in CBIR and it is computed as follows:

$$d(\vec{v}_1, \vec{v}_2) = \sum_{d=1}^{D} |v_{1d} - v_{2d}| \tag{3.29}$$

where $\vec{v}_1$ & $\vec{v}_2$ are mean of the features in each cluster and feature vector for the query image, respectively where D represents the dimension of each feature vector. The city block distance has very low computational complexity and it is robust to outliers which make it very good choice to be used in many applications [176].

3.5.5.2 Texture Image Retrieval via Posterior Probability

In Bayesian approach to categorize data into different groups and classes, the objective is to find the most probable set of group or class descriptions using the data and prior information [177]. Another method for image retrieval is to apply Bayesian classification and clustering criteria. The idea of Bayesian classification is to assign the most likely class to the given feature vector based on posterior probability [178]. Bayesian classification plays an important role in many practical

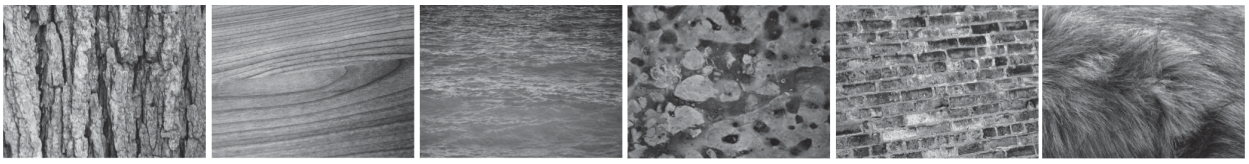


Figure 3.8: Different Texture Images from UIUC dataset

applications such as text classification, medical analysis, etc. With the parameters estimated in the clustering process, given an input feature vector we should be able to find the posterior probability with respect to each of the clusters. We can then retrieve the images from the cluster with the closest posterior probability. If we have a test image, that has a feature vector $\mathscr{I}$ extracted via BLMM in wavelet domain, then posterior probability can be computed for the feature vector of this image using the parameters learned via BLMM as follows:

$$p(j|\mathscr{I}) \propto p(\mathscr{I}|\xi_j)p_j \tag{3.30}$$

where ξ_j and p_j are the learned parameters of mixture model.

Table 3.4: Performance Metrics for UIUC dataset in feature extraction and clustering

Performance Metrics (%)	Feature Extraction			
	BLMM		LMM	
	Clustering Models		Clustering Models	
	BLMM	LMM	BLMM	LMM
Accuracy	**74.25**	71.25	**72.50**	67.50
Sensitivity	**74.25**	71.25	**72.50**	67.50
Specificity	**97.14**	96.81	**96.94**	96.39
Precision	**75.30**	72.28	**73.76**	68.67
FPR	**02.86**	03.19	**03.06**	03.61
F1-Score	**74.33**	71.32	**72.60**	67.64
MCC	**71.75**	68.41	**69.88**	64.31
G-Mean 1	**74.77**	71.76	**73.13**	68.08
G-Mean 2	**84.93**	83.05	**83.84**	80.66

3.5.5.3 Texture Image Retrieval via Kullback-Leibler Divergence

The City-block distance uses the Minkowski distance to compare two feature vectors. The distance function using Minkowski distance is not very effective because some feature values can be very large as compared to the rest of features in the vector and these features can jeopardize the entire distance score. The Kullback-Leibler (KL) divergence provides an effective way to compute the similarity between two distributions. If we assume that wavelet coefficients of two images in a

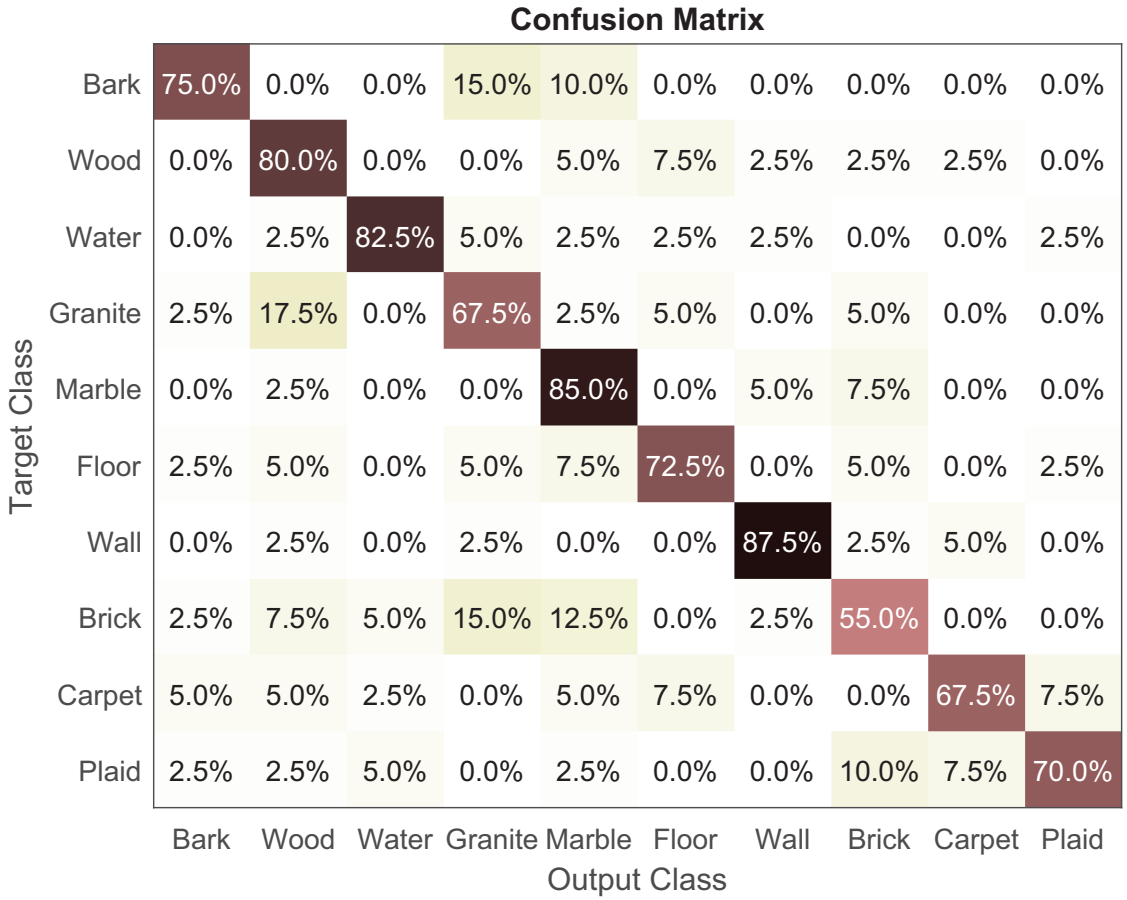


Figure 3.9: Confusion matrix of UIUC dataset with BLMM for feature extraction and clustering

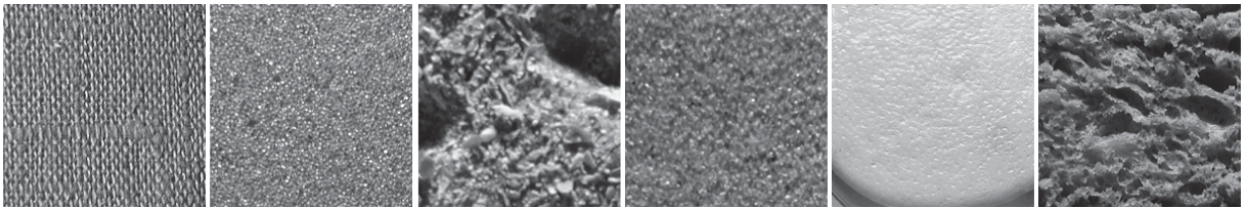


Figure 3.10: Different Texture Images from KTH-TIPS dataset

wavelet subspace are represented by two PDFs $p(x)$ and $q(x)$. The KL divergence for these two PDFs can be computed as follows [163, 164]:

$$d(p(x),q(x)) = \int p(x) \ln \frac{p(x)}{q(x)} dx \tag{3.31}$$

For the similarity measurement between an image and query, KL divergence is required to be computed between Laplace mixture distributions for each decomposed wavelet subspace and them

Table 3.5: Performance Metrics for KTH-TIPS dataset in feature extraction and clustering

Performance Metrics (%)	Feature Extraction			
	BLMM		LMM	
	Clustering Models		Clustering Models	
	BLMM	LMM	BLMM	LMM
Accuracy	**75.83**	72.67	**73.50**	69.17
Sensitivity	**75.83**	72.67	**73.50**	69.17
Specificity	**97.31**	96.96	**97.06**	96.57
Precision	**76.63**	73.46	**74.45**	69.50
FPR	**02.69**	03.04	**02.94**	03.43
F1-Score	**75.87**	72.78	**73.59**	69.18
MCC	**73.41**	69.92	**70.89**	65.85
G-Mean 1	**76.23**	73.06	**73.98**	69.33
G-Mean 2	**85.91**	83.94	**84.46**	81.73

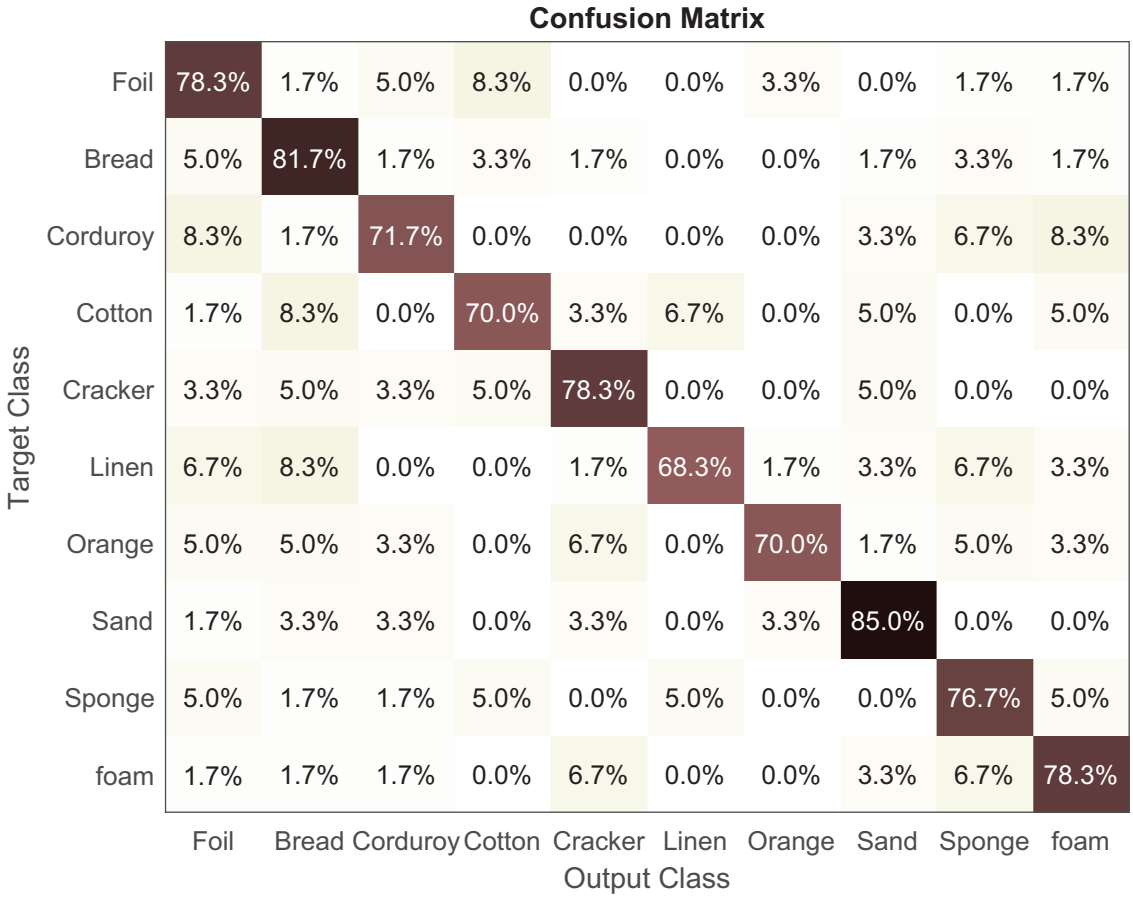


Figure 3.11: Confusion matrix of KTH-TIPS dataset with BLMM for feature extraction and clustering

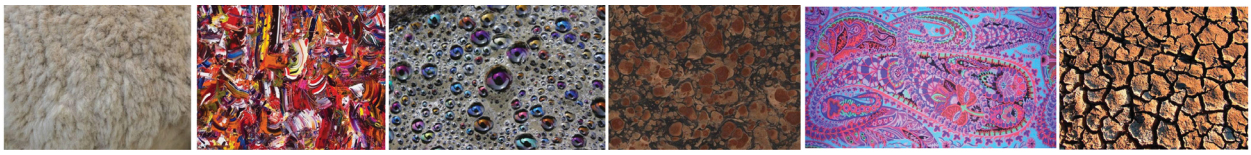


Figure 3.12: Different Texture Images from DTD dataset

Table 3.6: Performance Metrics for DTD dataset in feature extraction and clustering

Performance Metrics (%)	Feature Extraction			
	BLMM		LMM	
	Clustering Models		Clustering Models	
	BLMM	LMM	BLMM	LMM
Accuracy	**76.37**	73.12	**73.87**	70.63
Sensitivity	**76.38**	73.12	**73.87**	70.63
Specificity	**97.38**	97.01	**97.10**	96.74
Precision	**76.60**	73.49	**74.15**	71.26
FPR	**02.63**	02.99	**02.90**	03.26
F1-Score	**76.35**	73.18	**73.91**	70.71
MCC	**73.81**	70.27	**71.07**	67.59
G-Mean 1	**76.49**	73.31	**74.01**	70.94
G-Mean 2	**86.24**	84.23	**84.69**	82.66

Table 3.7: Performance Metrics for Stex dataset in feature extraction and clustering

Performance Metrics (%)	Feature Extraction			
	BLMM		LMM	
	Clustering Models		Clustering Models	
	BLMM	LMM	BLMM	LMM
Accuracy	**75.25**	71.75	**73.50**	68.75
Sensitivity	**75.25**	71.75	**73.50**	68.75
Specificity	**97.25**	96.86	**97.06**	96.53
Precision	**75.53**	72.09	**74.01**	69.07
FPR	**02.75**	03.14	**02.94**	03.47
F1-Score	**75.24**	71.68	**73.42**	68.61
MCC	**72.58**	68.69	**70.69**	65.33
G-Mean 1	**75.39**	71.92	**73.75**	68.91
G-Mean 2	**85.55**	83.37	**84.46**	81.46

up to get overall distance for the image [163, 164]. From Eqs. (3.7 & 3.31), we can get KL divergence for single wavelet subspace for Laplace mixture model as follows:

$$d(p_1(W), p_2(W)) = \int (p(W|0, b_{s1})p_{s1} + p(W|0, b_{l1})p_{l1}) \ln \frac{(p(W|0, b_{s1})p_{s1} + p(W|0, b_{l1})p_{l1})}{(p(W|0, b_{s2})p_{s2} + p(W|0, b_{l2})p_{l2})} dW \tag{3.32}$$

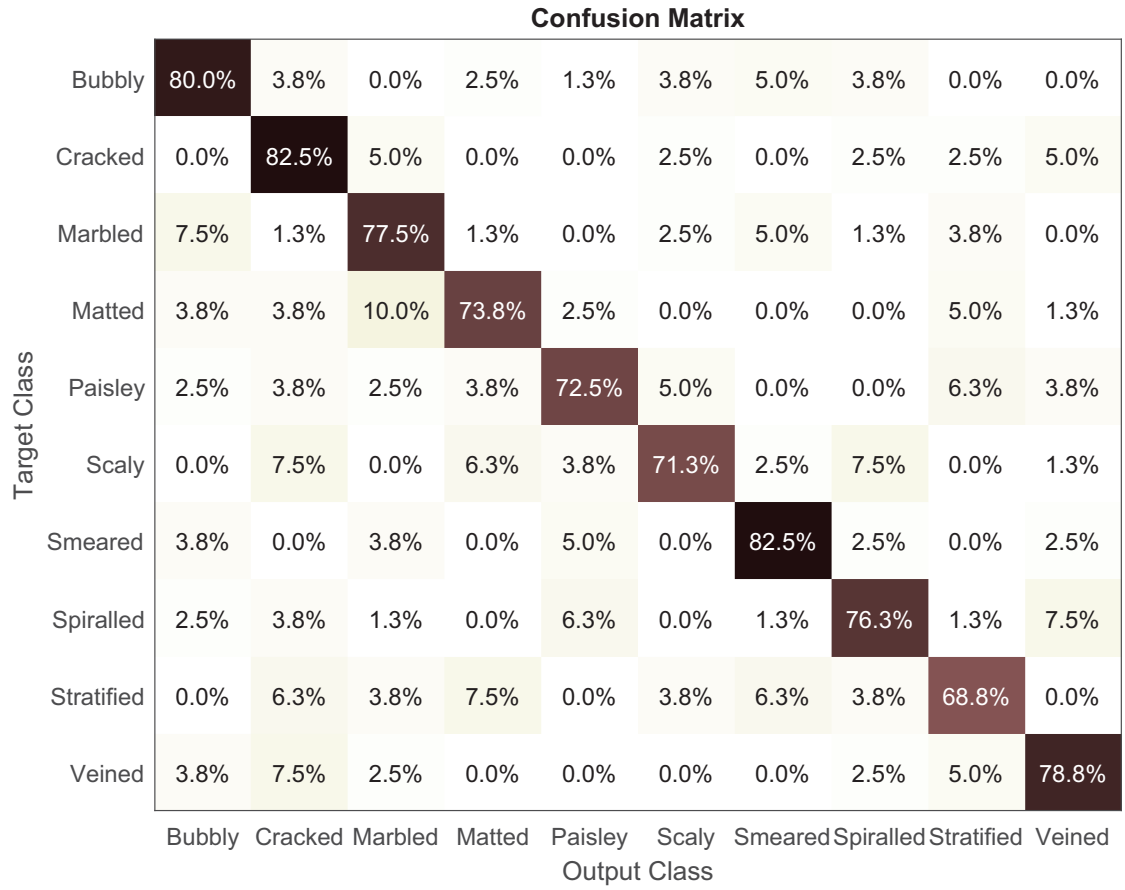


Figure 3.13: Confusion matrix of DTD dataset with BLMM for feature extraction and clustering

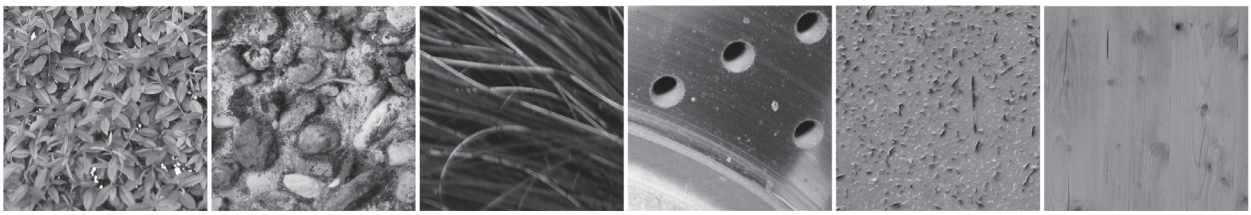


Figure 3.14: Different Texture Images from Stex dataset

where $p_1(W)$ is the Laplace mixture distribution of reference image and $p_2(W)$ is the Laplace mixture distribution of query image. However there is no closed form solution for KL divergence given in Eq. (3.32) and the only possibility is its numerical computation. The computational complexity is so high that it is not considered to be feasible for CBIR. The alternative to avoid this computational complexity is to divide the Laplace mixture distribution into two separate Laplace distributions and it is observed that a closed form solution can be easily computed for each separate Laplace distribution. Based on the above discussion, a new KL divergence based similarity for

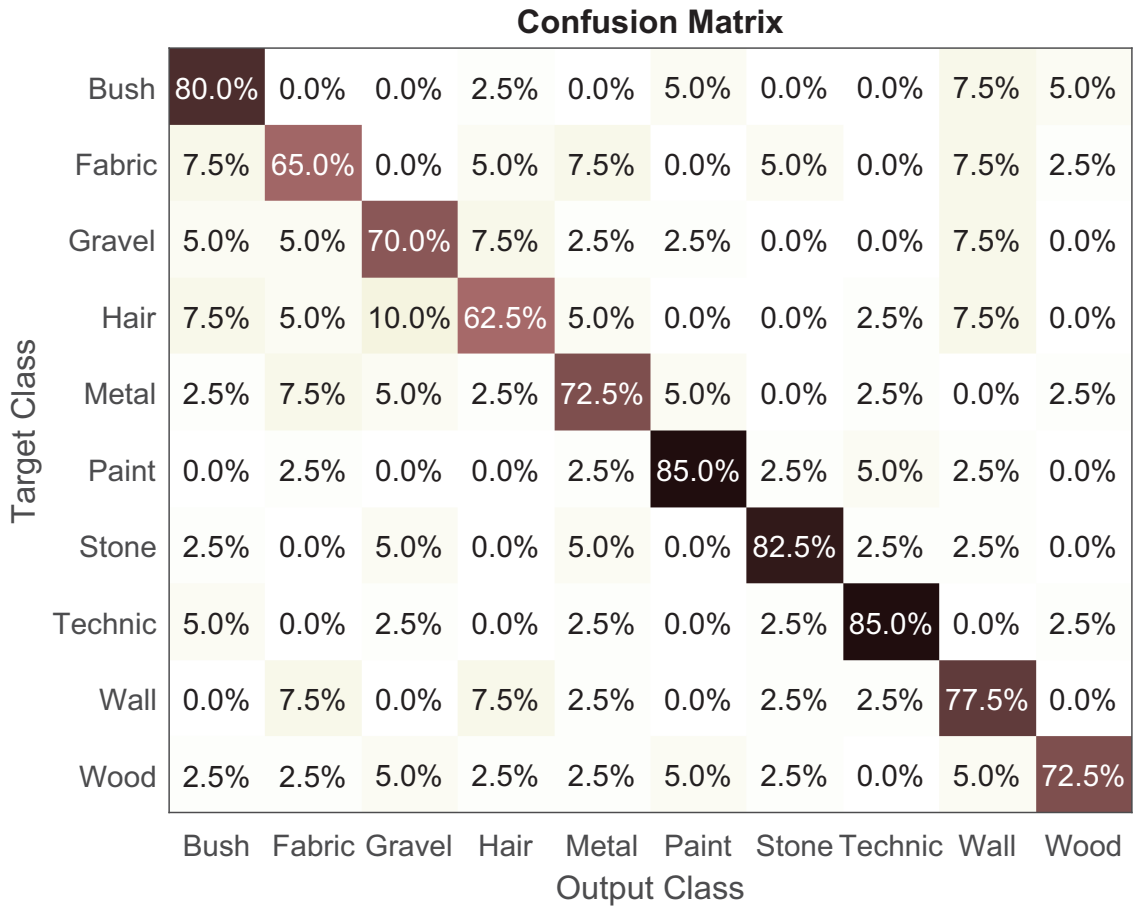


Figure 3.15: Confusion matrix of STex dataset with BLMM for feature extraction and clustering

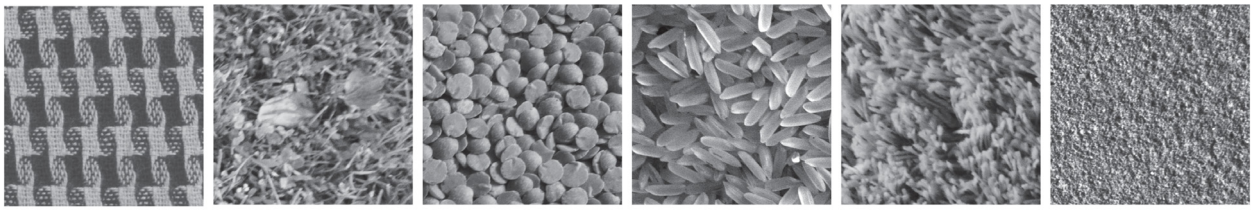


Figure 3.16: Different Texture Images from Kylberg dataset

Laplace mixture model is presented as follows:

$$d(p_1(W), p_2(W)) = F_s(W) + F_l(W) \tag{3.33}$$

$$F_s(W) = \int p(W|0, b_{s1}) p_{s1} \ln \frac{p(W|0, b_{s1}) p_{s1}}{p(W|0, b_{s2}) p_{s2}} \tag{3.34}$$

$$F_l(W) = \int p(W|0, b_{l1}) p_{l1} \ln \frac{p(W|0, b_{l1}) p_{l1}}{p(W|0, b_{l2}) p_{l2}} \tag{3.35}$$

Table 3.8: Performance Metrics for Kylberg dataset in feature extraction and clustering

Performance Metrics (%)	Feature Extraction			
	BLMM		LMM	
	Clustering Models		Clustering Models	
	BLMM	LMM	BLMM	LMM
Accuracy	**76.75**	72.50	**74.37**	70.38
Sensitivity	**76.75**	72.50	**74.37**	70.37
Specificity	**97.42**	96.94	**97.15**	96.71
Precision	**77.16**	73.16	**74.95**	71.07
FPR	**02.58**	03.06	**02.85**	03.29
F1-Score	**76.81**	72.55	**74.41**	70.48
MCC	**74.32**	69.67	**71.72**	67.34
G-Mean 1	**76.96**	72.83	**74.66**	70.72
G-Mean 2	**86.47**	83.84	**85.00**	82.50

After the integral calculations, the two separate KL divergences $F_s(W)$ and $F_l(W)$ have simplified closed form as follows:

$$F_s(W) = p_{s1} \ln \frac{p_{s1} b_{s2}}{p_{s2} b_{s1}} + p_{s1} \left(\frac{b_{s1}}{b_{s2}} - 1 \right) \tag{3.36}$$

$$F_l(W) = p_{l1} \ln \frac{p_{l1} b_{l2}}{p_{l2} b_{l1}} + p_{l1} \left(\frac{b_{l1}}{b_{l2}} - 1 \right) \tag{3.37}$$

It is observed from the closed form solutions that this new KL divergence based similarity measure can be efficiently computed using LMM parameters [163, 164]. For computation of the closed form, Laplace distributions are used in Eqs. (3.34 & 3.35), but once the closed form is achieved which represents only parameters of mixture model as presented in Eqs. (3.36 & 3.37), we use the parameters obtained by BLMM which has been proven very effective in parameter estimation as described in Section 3.3 & 3.4. It is worth mentioning that since the solution requires only estimated parameters of mixture model in wavelet domain to compute the KL divergence, we use the mean of feature vector for each reference class instead of reference image and KL divergence is computed between mean of feature vectors of each class and query image. The computation complexity of our solution is retained at the same level as other conventional similarity measures using Minkowski [163, 164].

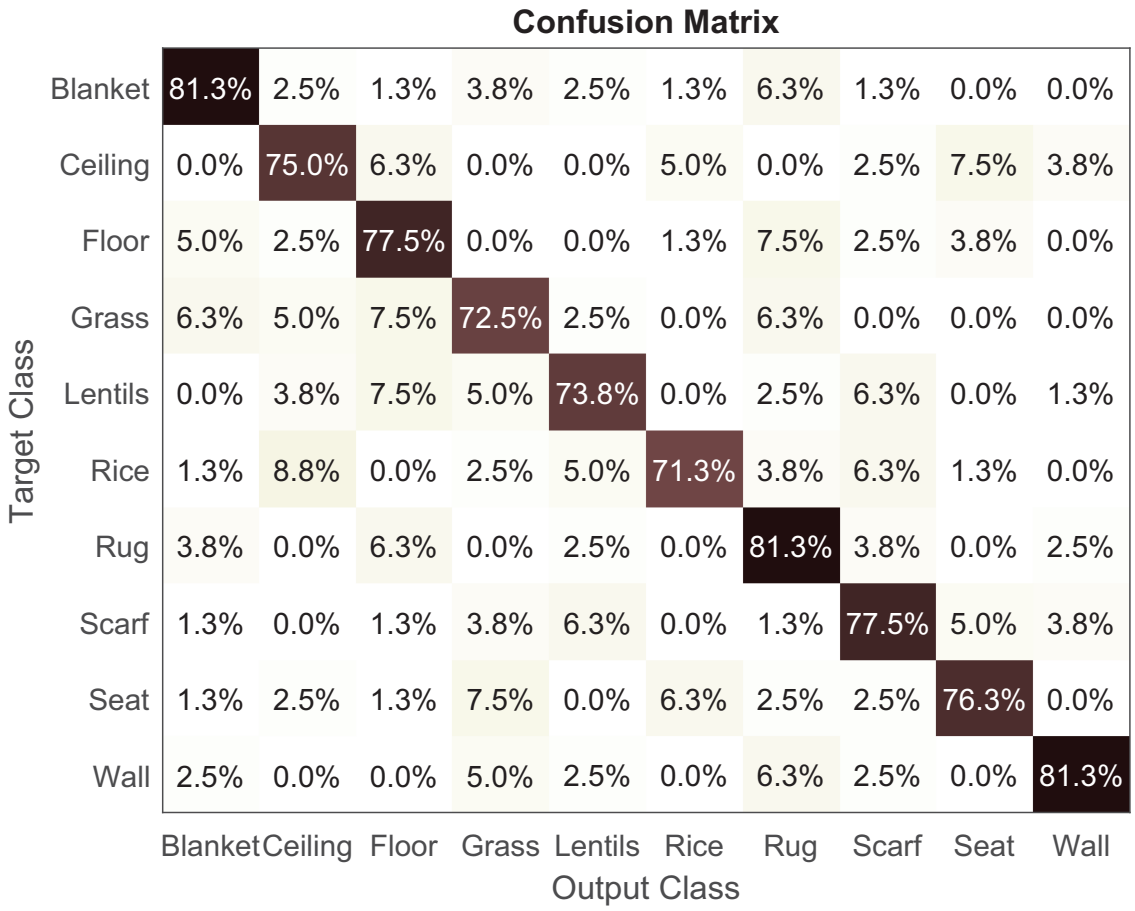


Figure 3.17: Confusion matrix of Kylberg dataset with BLMM for feature extraction and clustering

3.5.6 Experiments and Results

3.5.6.1 Design of Experiments

In this application, BLMM is applied to feature extraction, texture image categorization in unsupervised manner and CBIR and demonstration of effectiveness of this model requires set of experiments for each stage. We have conducted several set of experiments to validate the performance of BLMM with UIUC, KTH-TIPS, DTD, STex and Kylberg databases [179–183]. Selected parts of these datasets are used in this application and experimental framework is designed in a manner that highlight the performance of proposed model at each stage of this application very effectively. The first stage is feature extraction, where image in wavelet domain is modeled through BLMM to represent the images in feature space as described in Subsection (3.5.3). In the experiments presented in this work, Haar wavelet filter is used for decomposition of images. In order to model the wavelet coefficients with BLMM, 3-level decomposition is adopted for feature extraction. In the feature representation phase, 2-components mixture model with zero mean is applied and complete model

description and learning is given in Section (3.2). During the modeling, wavelet coefficients are represented as uni-variate data for model learning. Once the data are represented in feature space, feature normalization is applied as Eq. (3.28), which helps to avoid biasing from abnormal values in data. After feature space representation of texture images, data are modeled with BLMM for image categorization in an unsupervised manner and learning is followed from Section (3.2) for multivariate data. The data are used without labels and clustering provides label to each feature vector by assigning it to a particular class and this clustering information achieved in this stage also serves as index to CBIR. The mean of feature vectors in each cluster is computed and used to find the similarity between query image and each cluster of the data, in the image retrieval phase. The effectiveness of propose approach is validated through different performance measures which are mentioned in [106–109] and whole experimental framework is also modeled with LMM to have a comparison between BLMM and LMM. In the proposed framework, image clustering task also validates the feature extraction and all the datasets are considered in these experiments. For image retrieval, KTH-TIPS, DTD, STex and Kylberg databases are adopted for experimental framework. UIUC dataset is only used in image clustering experiments and it is not adopted for image retrieval due to limited number of images per class.

3.5.6.2 Experimental Framework for Image Clustering and Results: UIUC Dataset

UIUC dataset is adopted as a starting point in our experiments for BLMM. In UIUC dataset, there are 25 categories of texture images and it has 40 images in each class. In this experiment for texture image clustering, data from 10 different categories from this dataset are selected which make 400 images available for the task. The first step in this application is feature extraction, where 3-level decomposition of texture images is used to transform the images into wavelet domain which is further transformed into feature space by applying two-component BLMM. It is noteworthy that training of wavelet coefficients with BLMM is performed by considering data as uni-variate. Once the texture images are transformed into feature space, next step is to apply BLMM for categorizing the data into different clusters. In order to perform clustering, feature vectors of texture images are trained with 10-component BLMM. In order the validate the performance of this experiment and effectiveness of BLMM, it is very important to have a comparison with an existing model in a similar setting and same experiments are also performed with LMM. Since proposed model is applied at feature extraction phase and image clustering phase in the same task and it is also compared with LMM, it will create four scenarios for comparison at both stages. Sample images from UIUC dataset are provided in Fig. (3.8) and results of this experiment are given in Table (3.4). For the evaluation, different performance measures are adopted and results are presented to separately compare the BLMM at feature extraction level and in image clustering with LMM. The first column

in Table (3.4) provide the performance metrics considered in this experimental framework. Second and third column provide the clustering results with features extracted via BLMM and fourth and fifth columns provide the clustering results with features extracted via LMM. For a comparison of proposed feature extraction via BLMM with LMM, column 5 provides the results with feature extraction via LMM and column 3 provides the results for feature extraction via BLMM. Clustering in both experiments is performed by LMM, which gives us a chance to examine the effectiveness of feature extraction with BLMM and compare it with LMM. It is observed that image clustering is significantly improved when features are extracted with BLMM as compared to LMM. A similar comparison for feature extraction is also possible with the results of columns 2 and 4 where feature extraction is performed with BLMM and LMM, respectively and image clustering is performed by BLMM in both experiments. The results in column 2 and 4 also indicates the effectiveness of BLMM in feature extraction to represent the texture images in wavelet domain and modeled with BLMM. For the comparison of proposed model in texture images clustering, results provided in column 4 and 5 are considered where features are extracted with LMM in both cases and clustering is done by BLMM and LMM, respectively. The comparison of results indicates the effectiveness of BLMM in modeling the texture images to perform clustering. A similar comparison is evident from the results of columns 2 and 3 where clustering is performed by BLMM and LMM, respectively and feature extraction is done by BLMM in both cases. This comparison also highlights the effectiveness of BLMM in texture image clustering. The confusion matrix for the results of column 2 is provided in Fig. (3.9), where both stages are (feature extraction and image clustering) performed by BLMM which indicates the best performance of this experiment on UIUC dataset.

3.5.6.3 Experimental Framework for Image Clustering and Results: KTH-TIPS Dataset

Our next experiment on image clustering is performed on KTH-TIPS dataset which serves the purpose to examine the performance at feature extraction level and in image clustering. KTH-TIPS dataset consists of images of 10 different categories, where each category has 81 images. This dataset is part of both image clustering and retrieval experiments, that's why it is divided into two parts for making it possible to be used in both experiments. In this experiment, 60 images are selected from each category which make 600 images available for image clustering task. The experimental framework is prepared in a similar manner for feature extraction and image clustering as described in Section (3.5.6.2) for UIUC dataset. The features are extracted by modeling the wavelet coefficients with two-component mixture model whereas image clustering is performed by 10-component mixture model. The sample images from KTH-TIPS dataset are given in Fig. (3.10) and experimental results are provided in Table (3.5). The evaluation of experiments for KTH-TIPS dataset is done by performance metrics provided in the first column of Table (3.5). The results of

this experimental framework are provided in columns 2, 3, 4 and 5 which provide the comparison of at both stages of images clustering framework in a similar manner as described in Section (3.5.6.2). From the set of experiments with KTH-TIPS dataset and evaluation of results, it is observed that BLMM has performed significantly well in feature extraction phase and image clustering. The performance is achieved when both stages (feature extraction and image clustering) are modeled through BLMM as compared to LMM and confusion matrix to demonstrate this performance is given in Fig. (3.11).

3.5.6.4 Experimental Framework for CBIR and Results: KTH-TIPS Dataset

In the experiments for image retrieval, clustering framework developed in the previous experiments is adopted to perform image retrieval. In this experimental framework, image retrieval is performed on KTH-TIPS dataset and it is an extension of image clustering framework presented in Section (3.5.6.3). In the clustering framework, 60 images were selected for training the model and 20 images were chosen for testing on image retrieval. The query images from testing data are processed through feature extraction in wavelet domain via BLMM and normalized. During the clustering phase, we get cluster assignment for all the images used in the training process and we compute mean of feature vectors for each cluster to be used for image retrieval. In the next step, a similarity measure is computed for each query image with respect to mean of feature vectors of each cluster. In the image retrieval framework, we have proposed City-block distance, posterior probability and KL-Divergence for computing the similarity index for query images with respect to each cluster formed in the image clustering phase. Similarity measures are computed as described in the Section (3.5.5) with Eqs. (3.29, 3.30, 3.33). For image retrieval using KTH-TIPS dataset, 200 images are used as query images from the test data. In the image retrieval experiments, feature extraction is performed by BLMM and models are trained with BLMM and LMM for comparison which will lead to two comparison scenarios for each similarity measure. Image retrieval results for KTH-TIPS dataset with respect to all similarity measures are given in Table (3.9). The first column in the table, represents the performance metrics used for the evaluation of this set of experiments. In the columns 2 and 3, image retrieval results for city block distance are presented which is modeled with BLMM and LMM during the training phase. From the comparison of these results, it is evident that model trained with BLMM has better retrieval capability as compared with LMM. In column 4 and 5, retrieval results computed with posterior probability are presented where training was done via BLMM and LMM and a comparisons shows that BLMM has better modeling capabilities which improves the image retrieval results. A similar conclusion is achieved by examining the results for KL-Divergence presented in column 6 and 7 which show the effectiveness of BLMM in image retrieval. The best score is achieved when model is trained

with BLMM and similarity measure is computed with KL-Divergence as presented in column 6 in the Table (3.9) and confusion matrix for this result is presented in Fig. (3.18) which also explains the classification performance for each class. A comparison for different similarity measures is also possible from these set of experiments and it is also observed that posterior probability has show better retrieval capability as compared to City-block distance and KL-divergence has shown the best performance in all the three similarity measures.

3.5.6.5 Experimental Framework for Image Clustering and Results: DTD Dataset

Our third experiment on texture image clustering is performed on DTD datset, which comprises of texture images from 47 different categories and 120 images in each category of dataset. The experiments are conducted to examine the performance of BLMM for feature extraction and image clustering similar to experiments conducted in Section (3.5.6.2). DTD dataset is used for both image clustering and retrieval framework and it is divided into two parts to be used for both experimental frameworks. For image clustering framework, 10 categories of texture images are selected, where 80 images from each category are chosen which make it possible to have 800 images for experiments on image clustering which in turn also validate the feature extraction phase. In the feature extraction phase 2-component mixture model is employed whereas in the clustering phase 10-component mixture model is used. Sample images from DTD dataset are provided in Fig. (3.12). Experimental results for the experiments conducted to demonstrate the feature extraction and image clustering with BLMM are provided in Table (3.6). In the first column, performance metrics used in this experimental framework are given and rest of the columns provide the experimental results with DTD dataset to demonstrate the effectiveness of BLMM at feature extraction and image clustering phase. From Table (3.6), a comparison for feature extraction and image clustering is possible in a similar manner as provided in Section (3.5.6.2) and from evaluation of results and by observing the whole experimental framework, it is evident that BLMM has performed better than LMM at both stages of this application and best performance is observed in the results given in column 2 of the Table (3.6), where both stages are modeled via BLMM. The results of experiment with both stages performed via BLMM are provided in Fig. (3.13) which demonstrates the effectiveness of BLMM in this task as compared to LMM.

3.5.6.6 Experimental Framework for CBIR and Results: DTD Dataset

The second experiment on CBIR is performed using DTD dataset and this experimental framework is an extension of image clustering task presented in Section (3.5.6.5), where 80 images from each class of texture images are used in the training process and 40 images are reserved for image retrieval experiments. For experiments in this framework, 400 images from 10 categories are used.

Experimental framework for the retrieval task with DTD dataset is prepared in a similar manner as discussed in Section (3.5.6.4), where feature extraction phase is modeled with BLMM and training process is accomplished by BLMM and compared with LMM. Image retrieval is performed by considering the test images as query images and computing the similarity measure between query image and mean of each cluster. For the similarity measure, City block distance, posterior probability and KL-Divergence are considered as explained in Section (3.5.5). Experimental results for image retrieval using all the similarity measures and models trained with BLMM and LMM are presented in Table (3.10). In the first column, all the performance measures are presented and in column 2 and 3, retrieval results for city block distance are presented where training process is performed by BLMM and compared with LMM. In column 4 and 5, results for posterior probability and in column 6 and 7, image retrieval results with KL-divergence are presented. From the set of experiments, it is observed that BLMM has better modeling capabilities and better performance in image retrieval as compared to LMM for all experiments with different similarity measures. It is also concluded that KL-Divergence has show better performance in the image retrieval task. Best performance in the image retrieval experiment with DTD dataset is achieved when model is trained with BLMM and similarity measure is computed with KL-Divergence as presented in column 6 of Table (3.10). The confusion matrix for the best performance is presented in Fig. (3.19), which also present the classification accuracy for each class.

3.5.6.7 Experimental Framework for Image Clustering and Results: STex Dataset

STex texture image dataset is chosen to perform experiments for our fourth experimental framework on image clustering. From STex dataset, images from 10 categories are selected where 14 images are chosen for our experiments. In this dataset, each image is composed of 1024×1024 pixels and we have divided these high quality images into sub-images to make it possible to be used in our experiments. It is worthy to note that one image (1024×1024) can be divided into four images having 512×512 dimensions. This modification of images results in 56 images in each class. Since STex dataset is used in the experiments for both image clustering and retrieval, it divided to into two subsets to be used in both frameworks. For the clustering, we have used texture images from 10 different classes where each class is composed of 40 images, resulting in 400 images. Sample images from STex dataset are provided in Fig. (3.14). In the image clustering framework, feature extraction and clustering model learning is performed in a similar manner as described in Section (3.5.6.2). Experimental results are provided in Table (3.7), which demonstrate the effectiveness of proposed model in feature extraction and image clustering. Column 1 in Table (3.7), provide the performance metrics used for experiments and column 2, 3, 4 and 5

provide the results for image clustering with different models at feature extraction and image categorization phase. These experimental results clearly demonstrate the role of this experiment to see the effectiveness of BLMM. From the set of experiments modeled with BLMM and LMM and comparing the results of image clustering, it is evident that BLMM has significantly improved the clustering performance based on its application in feature extraction and modeling the texture data represented in feature space for image categorization. The best is performance is achieved when BLMM is applied in both stages of image clustering pipeline which is given in column 2 of Table (3.7) and the confusion matrix for this result is given in Fig. (3.13) which also demonstrate the classification performance for each class.

3.5.6.8 Experimental Framework for CBIR and Results: STex Dataset

STex dataset is employed to perform third experiment on image retrieval task and it is an extension of image clustering framework presented in Section (3.5.6.7). In STex dataset, 40 images from each class are used in the training process to perform clustering and 16 images per class are chosen as test data for image retrieval experiments. In this task, 160 texture images from 10 categories of STex dataset are selected to perform experiments. The experimental framework for retrieval using STex dataset is designed in a similar manner as discussed in Section (3.5.6.4) and three similarity measure are used to perform the image retrieval on a trained model. The retrieval results are presented in Table (3.11), where first column represent the performance measures used to examine the performance of this experiment and rest of the columns demonstrate the retrieval results for different similarity measure computed on models trained with BLMM and LMM. By examining the performance metrics for image retrieval task, it is observed that BLMM has performed significantly better than LMM for all settings of different similarity measures. It is also noted that posterior probability is an improvement on City-block distance in this task and KL-Divergence is better than all of similarity measures and best performance is achieved when BLMM is applied in the training process and KL-Divergence is used as similarity measure for image retrieval as depicted in column 6 of Table (3.11). The confusion matrix to represent this result is given in Fig. (3.20), which also demonstrate the performance for each class.

3.5.6.9 Experimental Framework for Image Clustering and Results: Kylberg Dataset

In the fifth experiment on texture image clustering, we have employed Kylberg dataset which possesses images from 28 different classes and each class consists of 160 images. Since Kylberg dataset is employed for experiments in both frameworks (image clustering and retrieval), it is divided into two parts. For our experiments on image clustering and retrieval, 120 images are selected from each class, where 80 texture images per class are selected for clustering framework

and 40 images per class are selected for experiments on image retrieval. In our experiments on image clustering, 800 images from 10 different classes are used. The sample images from Kylberg dataset are presented in Fig (3.16). Image clustering framework is designed to examine the performance of BLMM in feature extraction and image categorization phase and both stages are modeled as described in Section (3.5.6.2). The results of experimental framework on image clustering with Kylberg dataset are presented in Table (3.8), where first column presents the performance metrics considered in the experiments and rest of the columns provide the image clustering results with selection of different model at feature extraction and image clustering stage. From the set of experiments conducted on Kylberg dataset and comparing the results, it is evident that BLMM has better performance in feature extraction and image categorization which demonstrate the modeling capabilities of our proposed approach. In the experimental framework with Kylberg dataset, best performance is achieved with both stages performed via BLMM and it is given in second column of Table (3.8) and confusion matrix to demonstrate this performance is given in Fig. (3.17) which also represent the classification performance at each class level.

3.5.6.10 Experimental Framework for CBIR and Results: Kylberg Dataset

Kylberg texture image dataset is used in fourth experiment to validate the proposed image retrieval framework based on BLMM and different similarity measures. This experimental framework is an extension of texture image clustering framework described in Section (3.5.6.9) and it is designed in a similar manner as explained in Section (3.5.6.4) with image clustering phase modeled via BLMM and retrieval phase performed by the three similarity measures separately. For a comparison, image clustering phase is also modeled via LMM and rest of the experimental framework remains the same as with BLMM. For the experiment, model is trained with data from 10 categories (80 images per class) as described in Section (3.5.6.9) and 40 images per class are dedicated for retrieval task which make it possible to have 400 texture images in this experiment using Kylberg dataset. The results of image retrieval task from this experimental framework are presented in Table (3.12), which clearly demonstrate the effectiveness of BLMM and similarity measures for image retrieval. From the set of experiments, it is observed that if model is trained via BLMM, the retrieval framework will have better results with all similarity measures. It is also observed that posterior probability and KL-Divergence has performed better than City-block distance and best performance is achieved with KL-Divergence when clustering stage is modeled with BLMM. The results of best performance are presented in Fig. (3.21), which also explain the results for each class.

In this application BLMM is applied on different texture images databases and proposed in the feature extraction phase and image clustering phase. From the set of experiments performed

Table 3.9: Performance Metrics for KTH-TIPS dataset in CBIR

Performance Metrics (%)	City-block		Posterior Probability		KL-Divergence	
	BLMM	LMM	BLMM	LMM	BLMM	LMM
Accuracy	**78.00**	74.50	**79.50**	76.00	**81.00**	77.50
Sensitivity	**78.00**	74.50	**79.50**	76.00	**81.00**	77.50
Specificity	**97.56**	97.17	**97.72**	97.33	**97.89**	97.50
Precision	**78.25**	75.13	**80.05**	76.94	**81.41**	78.46
FPR	**02.44**	02.83	**02.28**	02.67	**02.11**	02.50
F1-Score	**77.94**	74.54	**79.52**	76.16	**81.00**	77.65
MCC	**75.61**	71.88	**77.40**	73.69	**79.02**	75.35
G-Mean 1	**78.13**	74.82	**79.77**	76.47	**81.20**	77.98
G-Mean 2	**87.23**	85.08	**88.14**	86.01	**89.04**	86.93

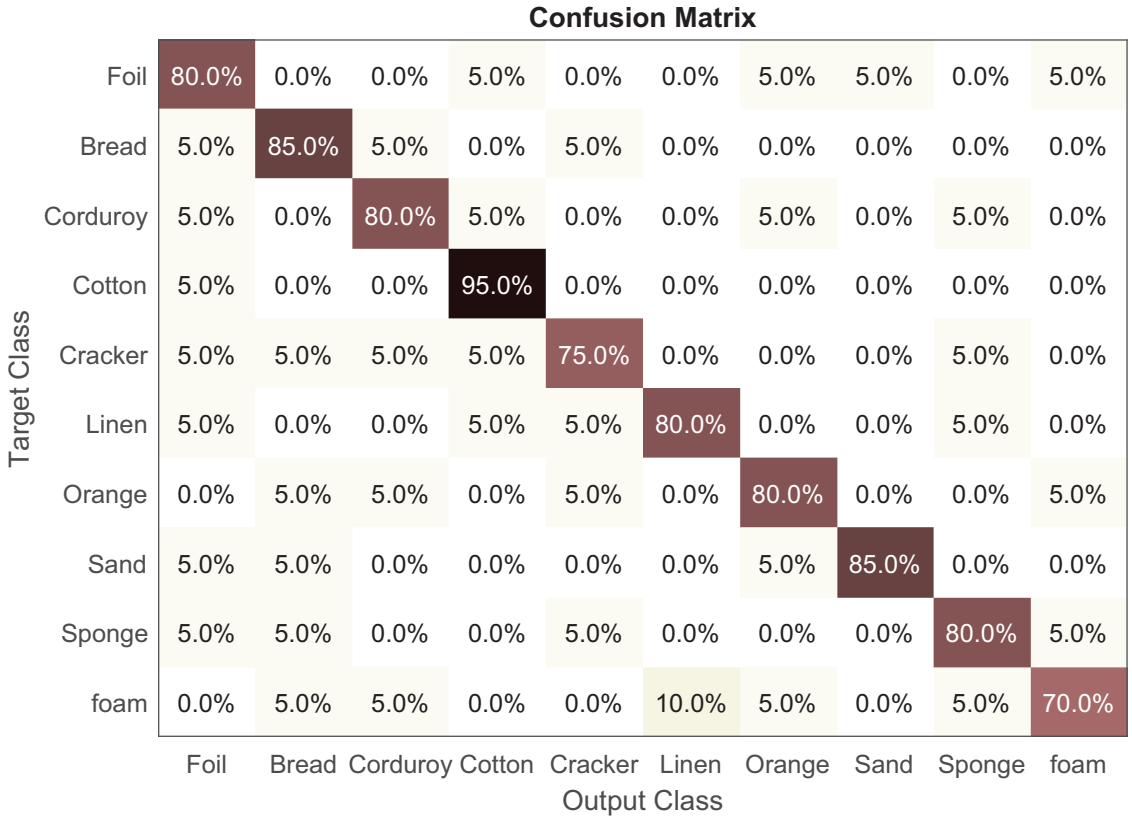


Figure 3.18: Confusion matrix of KTH-TIPS dataset CBIR

Table 3.10: Performance Metrics for DTD dataset in CBIR

Performance Metrics (%)	City-block		Posterior Probability		KL-Divergence	
	BLMM	LMM	BLMM	LMM	BLMM	LMM
Accuracy	**78.75**	73.25	**79.75**	74.50	**81.25**	75.75
Sensitivity	**78.75**	73.25	**79.75**	74.50	**81.25**	75.75
Specificity	**97.64**	97.03	**97.75**	97.17	**97.92**	97.31
Precision	**79.37**	73.94	**80.07**	75.23	**81.64**	76.30
FPR	**02.36**	02.97	**02.25**	02.83	**02.08**	02.69
F1-Score	**78.77**	73.28	**79.71**	74.53	**81.23**	75.76
MCC	**76.59**	70.51	**77.59**	71.91	**79.28**	73.23
G-Mean 1	**79.06**	73.60	**79.91**	74.86	**81.44**	76.02
G-Mean 2	**87.69**	84.30	**88.29**	85.08	**89.19**	85.85

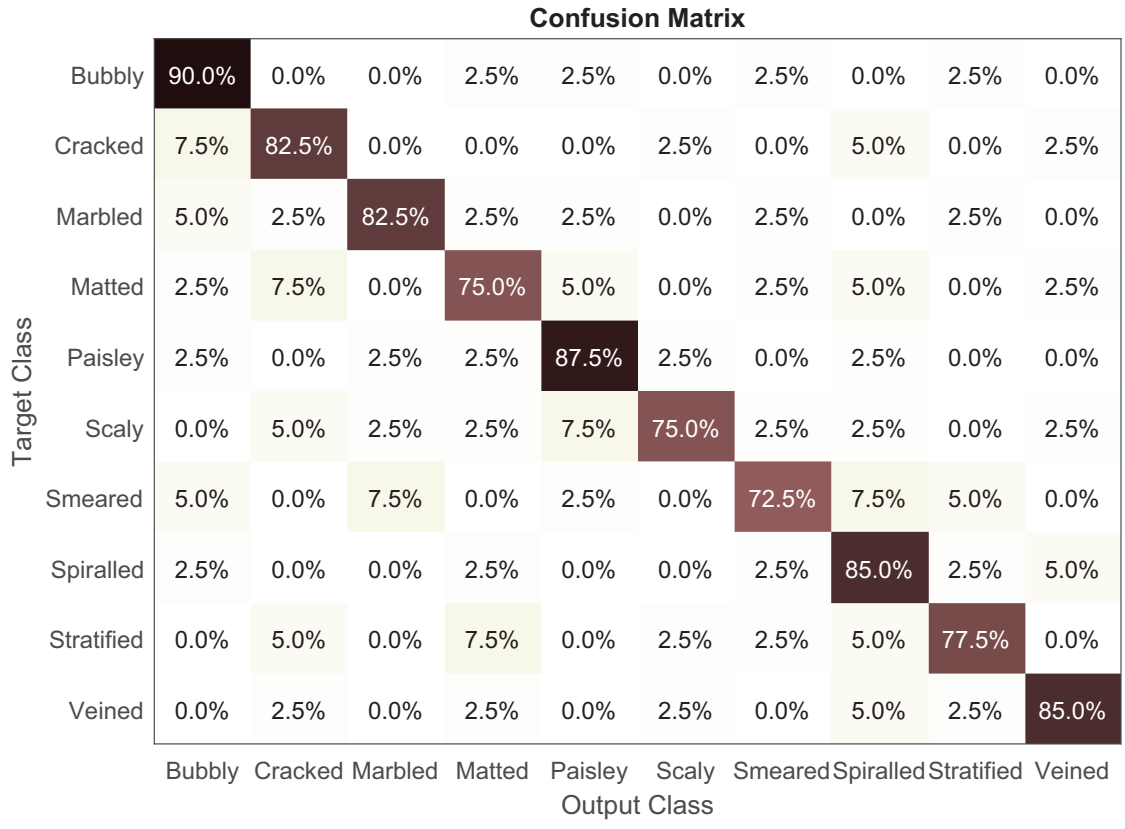


Figure 3.19: Confusion matrix of DTD dataset CBIR

Table 3.11: Performance Metrics for STex dataset in CBIR

Performance Metrics (%)	City-block		Posterior Probability		KL-Divergence	
	BLMM	LMM	BLMM	LMM	BLMM	LMM
Accuracy	**76.25**	70.63	**78.13**	72.50	**81.25**	75.00
Sensitivity	**76.25**	70.63	**78.13**	72.50	**81.25**	75.00
Specificity	**97.36**	96.74	**97.57**	96.94	**97.92**	97.22
Precision	**76.65**	71.21	**78.53**	73.35	**81.37**	75.75
FPR	**02.64**	03.26	**02.43**	03.06	**02.08**	02.78
F1-Score	**76.15**	70.57	**77.92**	72.44	**81.03**	74.98
MCC	**73.70**	67.52	**75.74**	69.68	**79.12**	72.45
G-Mean 1	**76.45**	70.92	**78.33**	72.92	**81.31**	75.38
G-Mean 2	**86.16**	82.66	**87.31**	83.84	**89.19**	85.39

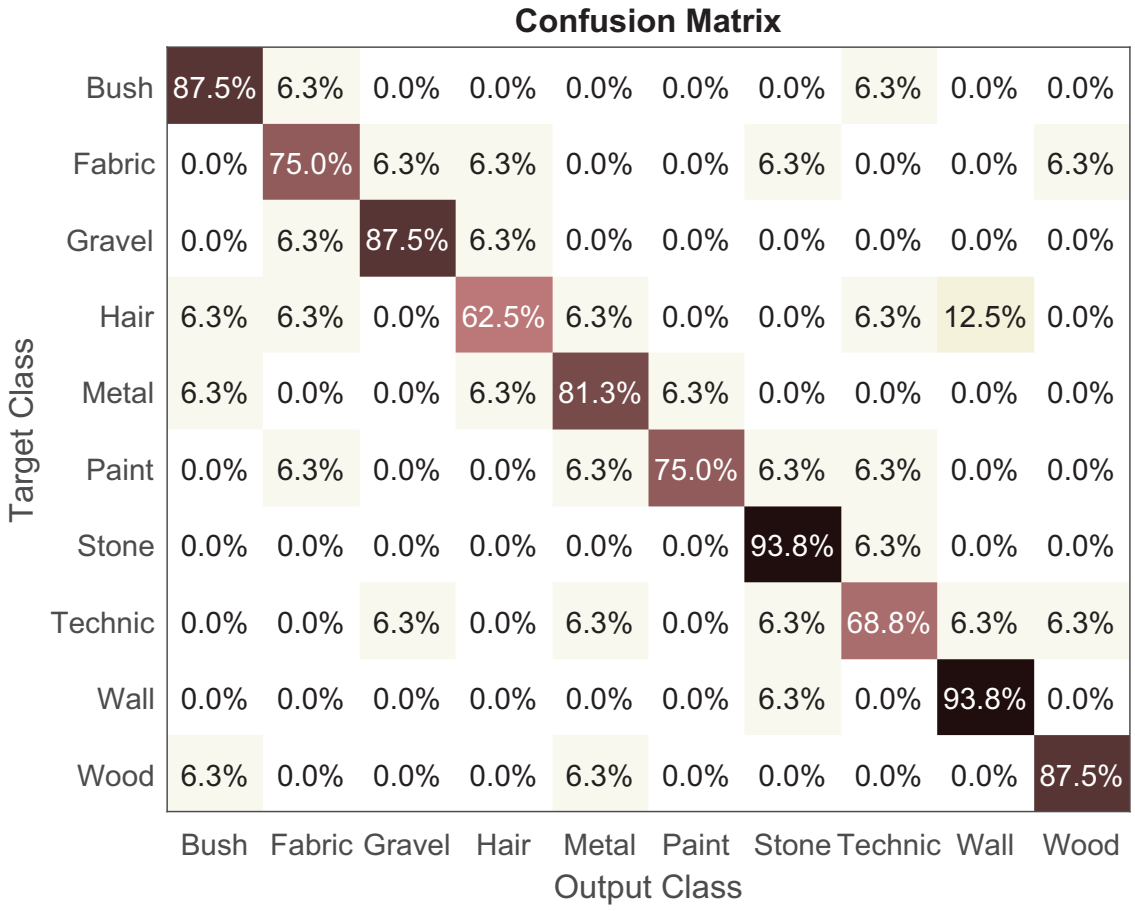


Figure 3.20: Confusion matrix of STex dataset CBIR

Table 3.12: Performance Metrics for Kylberg dataset in CBIR

Performance Metrics (%)	City-block		Posterior Probability		KL-Divergence	
	BLMM	LMM	BLMM	LMM	BLMM	LMM
Accuracy	**79.25**	74.00	**80.50**	75.25	**82.00**	76.75
Sensitivity	**79.25**	74.00	**80.50**	75.25	**82.00**	76.75
Specificity	**97.69**	97.11	**97.83**	97.25	**98.00**	97.42
Precision	**79.31**	74.46	**80.39**	75.68	**82.15**	76.97
FPR	**02.31**	02.89	**02.17**	02.75	**02.00**	02.58
F1-Score	**79.17**	74.00	**80.33**	75.23	**81.86**	76.69
MCC	**76.93**	71.26	**78.24**	72.63	**08.00**	74.21
G-Mean 1	**79.28**	74.23	**80.45**	75.46	**82.08**	76.86
G-Mean 2	**87.99**	84.77	**88.74**	85.55	**89.64**	86.47

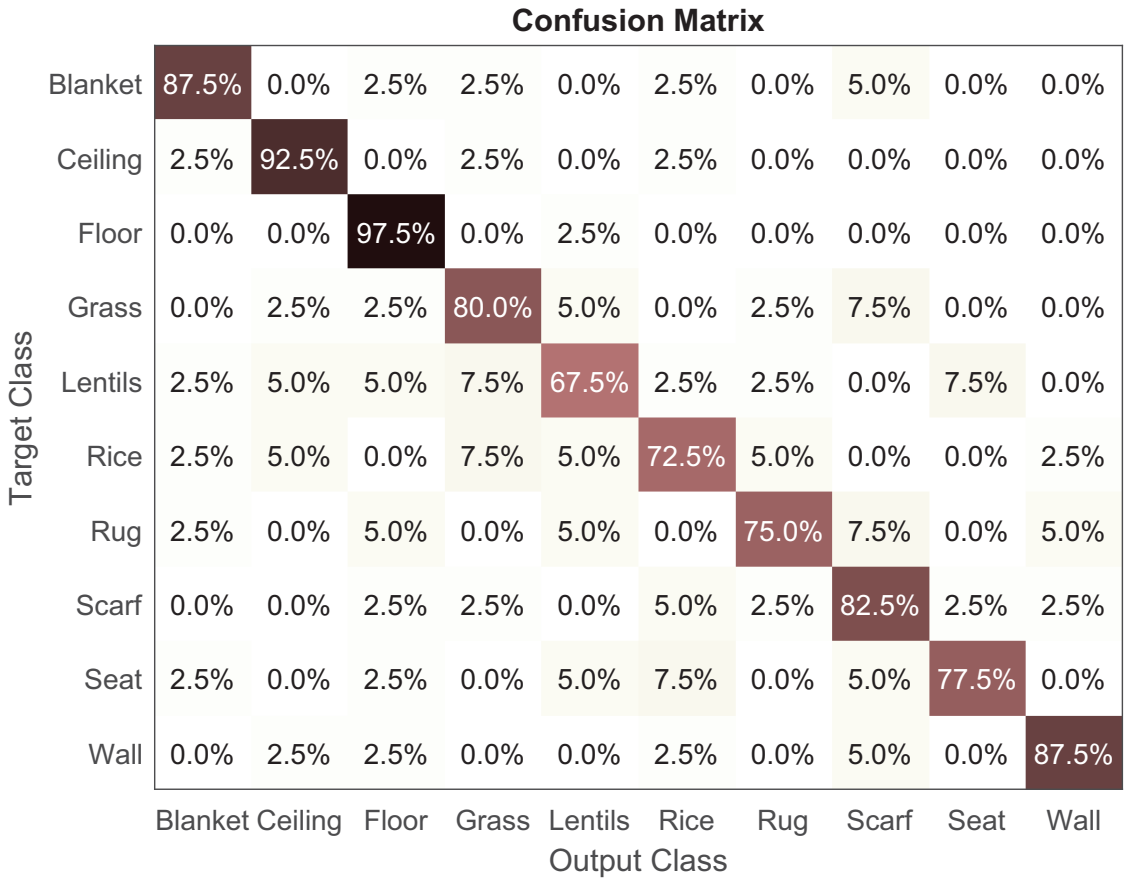


Figure 3.21: Confusion matrix of Kylberg dataset CBIR

on different datsets, presented in Sections (3.5.6.2, 3.5.6.3, 3.5.6.5, 3.5.6.7 & 3.5.6.9) demonstrate the effectiveness of BLMM in feature extraction and texture image clustering. To validate our approach, different performance measure are adopted to demonstrate the success of proposed approach as compared to LMM in a similar set of experiments.

The image clustering framework based on BLMM is further adopted for CBIR using different similarity measures and from the set of experiments presented in Sections (3.5.6.4, 3.5.6.6, 3.5.6.8 & 3.5.6.10), it is observed that trained model based on BLMM also improves the image retrieval for all the similarity measures (City-block distance, posterior probability and KL-Divergence) as compared to model trained with LMM. It is also observed that posterior probability is an improvement on City-block distance in this task and KL-Divergence has better performance in all experiments.

3.6 Discussion about BLMM

In this chapter, a mixture of bounded Laplace distribution is proposed which uses maximum likelihood approach for parameter estimation and optimization of parameters is performed by an EM algorithm with Newtons-Raphson method in an iterative procedure. In order to validate the performance of this model, it is applied to data clustering for synthetic data and several real datasets from different medical experiments. For synthetic data, one dimensional and two dimensional artificial histograms are generated and BLMM is applied to perform clustering on these synthetic datasets. It is observed that BLMM has performed very effectively on these artificial datasets which is depicted from real parameters used to generate these datasets and estimated parameters after the clustering from BLMM. For experiments on medical datasets, BLMM is applied for categorization of data into different classes through clustering and it has demonstrated its success in this task which is depicted through clustering accuracy. These results are also compared with clustering performed by K-Means and LMM and our proposed model also exhibit good clustering accuracy as compared to these algorithms.

In order to extend the experiments, this algorithm is proposed in image processing applications and it is applied to perform feature extraction in wavelet domain, texture image clustering and content based image retrieval. We have defined a strategy to perform feature extraction through BLMM in wavelet domain. Our model is also applied to texture image categorization in the same framework where feature extraction is also achieved by BLMM. Image clustering can be further used for image retrieval and we have introduced three different methods to perform image retrieval. We also have computed a novel closed form solution for KL divergence which is one of the three

methods used to perform image retrieval. For the validation of BLMM and proposed experimental framework for feature extraction, texture image categorization and image retrieval, different texture datasets (UIUC, KTH-TIPS, DTD, STex and Kylberg) are adopted. Different experiments are conducted using these datasets and results of these experiments demonstrate that BLMM has significantly improved the data modeling capabilities in the proposed experimental framework as compared to LMM for the same experimental setup. The experiments on clustering exhibits the success of our proposed approach in feature extraction and image clustering whereas rest of the experiments demonstrate the effectiveness of BLMM and proposed KL divergence in image retrieval.

3.7 Texture Image Categorization in Wavelet Domain via Naive Bayes Classifier Based on Laplace and Generalized Gaussian Distribution

If we assume that data from each class follows a probability distribution and estimate the parameters pertaining to that distribution, it is possible to develop a Naive Bayes classifier. In Naive Bayes classifier, parameters are estimated with respect to distribution of data for each class and new data is assigned to a class based on the learned parameters by maximum value of posterior probability. Naive Bayes classifier has been widely used in the industry for several classification tasks [184–186]. Particularly, Naive Bayes classifiers based on Gaussian distribution has profound influence in a number of medical, industrial and multimedia applications [187–189]. Use of Laplace distribution and generalized Gaussian distribution in many machine learning algorithms has proved to be a good solution which has demonstrated it success in different kind of data modeling applications. If we choose to model the features represented with BLMM in wavelet domain in a supervised learning approach, it can be modeled with Laplace and generalized Gaussian distributions due to the nature of data in wavelet domain [159, 162]. Generalized Gaussian distribution also has the ability to model Gaussian and Laplace distribution, which make the models more robust in data modeling. Hence, we propose a Naive Bayes classifier with these distributions. To test the performance of our proposed model we have chosen a challenging application namely texture classification. Texture classification plays an important role in industries which involves quality check in product manufacturing factories and many other multimedia tasks. In this chapter, we proposed Naive Bayes classifier based on Laplace and generalized Gaussian distributions which is further applied to perform texture image categorization. The classification framework is validated through set of experiments performed on UIUC, KTH-TIPS, and DTD datasets. For validation, different

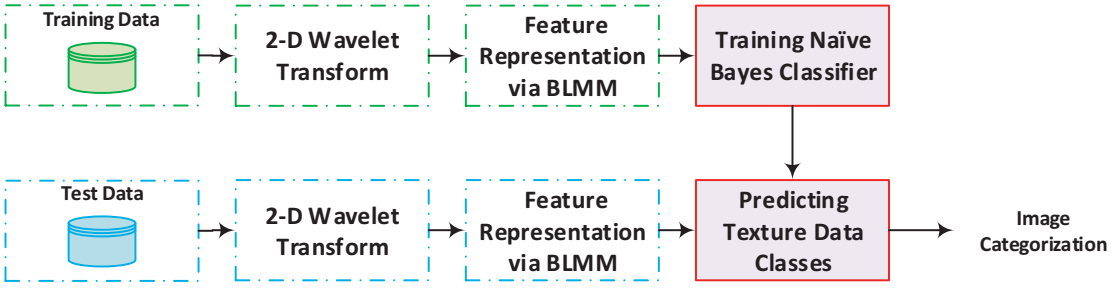


Figure 3.22: Framework for Texture Image Categorization via Naive Bayes Classifier

performance metrics are considered and effectiveness of our proposed approach is examined.

3.8 Proposed Algorithms

In this section proposed Naive Bayes classifier based on Laplace and generalized Gaussian distributions is presented. We have presented general formulation of Naive Bayes classifier and parameter estimation technique and then models based on Laplace and generalized Gaussian distribution are demonstrated with their parameter estimation.

3.8.1 Naive Bayes Classifier

If we consider that $\vec{X}$ having D-dimensions, as feature vector, c_k as the possible class with $k \in \{1,...,K\}$ and $p(\vec{X}|c_k)$ as the probability of $\vec{X}$ belong to class c_k, then general notation for posterior probability using Bayes theorem can be written as follows:

$$p(c_k|\vec{X}) = \frac{p(c_k)p(\vec{X}|c_k)}{p(\vec{X})} \tag{3.38}$$

where $p(c_k)$ is class probability and $p(\vec{X}|c_k)$ is class conditional probability. The objective is to maximize the posterior probability using parameters obtained from training data as follows:

$$\hat{c} = \underset{k\in\{1,...,K\}}{\arg\max} \; p(c_k|\vec{X}) \tag{3.39}$$

where $\hat{c}$ is predicted class label for feature vector $\vec{X}$ [190–192]. In a Naive Bayes classifier, to avoid curse of dimensionality, it is assumed that the features are independent and identically distributed

(i.i.d.), which define the $p(\vec{X}|c_k)$ as follows:

$$p(\vec{X}|c_k) = \prod_{d=1}^{D} p(X_d|c_k) \tag{3.40}$$

Since the denominator in posterior probability is a normalization constant and it can be ignored, which makes the Eq. (3.38) as follows:

$$\begin{aligned} p(c_k|\vec{X}) &\propto p(c_k)p(\vec{X}|c_k) \\ &\propto p(c_k)\prod_{d=1}^{D} p(X_d|c_k) \end{aligned} \tag{3.41}$$

Once the model is defined, next step is to find the parameters of the model from training data. For parameter estimation, maximum likelihood approach is adopted in Naive Bayes classifier. If the complete training dataset is represented as $\mathscr{X} = (\vec{X}_1, ..., \vec{X}_N)$ and $C = c_1, ..., c_K$ unique classes in the class label $\mathscr{Y} = (Y_1, ..., Y_N)$ for each data sample, then likelihood of data can be expressed as follows:

$$p(\mathscr{X}, \mathscr{Y}|\theta_k) = \prod_{i=1}^{N} p(Y_i = c_k) \prod_{d=1}^{D} p(X_{id}|Y_i = c_k) \tag{3.42}$$

where θ_k is set of parameters for Naive Bayes classifier for each class of the data which include class probability and parameters of distribution for each subset of the data belonging to class c_k. By taking the log of likelihood for mathematical convenience, Eq. (3.42) is expressed as follows:

$$\mathscr{L}(\mathscr{X}, \mathscr{Y}|\theta_k) = \log\left\{\prod_{i=1}^{N} p(Y_i = c_k) \prod_{d=1}^{D} p(X_{id}|Y_i = c_k)\right\} \tag{3.43}$$

In parameters estimation using maximum likelihood approach, parameters values are estimated by maximizing log-likelihood as follows:

$$\theta_k = \arg\max_{k\in\{1,...,K\}} \mathscr{L}(\mathscr{X}, \mathscr{Y}|\theta_k) \tag{3.44}$$

The log-likelihood can be further expressed as follows:

$$\begin{aligned} \mathscr{L}(\mathscr{X}, \mathscr{Y}|\theta_k) &= \sum_{i=1}^{N} \log p(Y_i = c_k) + \sum_{i=1}^{N} \log \prod_{d=1}^{D} p(X_{id}|Y_i = c_k) \\ &= \sum_{i=1}^{N} \log p(Y_i = c_k) + \sum_{i=1}^{N}\sum_{d=1}^{D} \log p(X_{id}|Y_i = c_k) \end{aligned} \tag{3.45}$$

In maximum likelihood estimate, for determining the class probability, it is ensured that $p(c_k) \geq 0$ for all classes and $\sum_{k=1}^{K} p(c_k) = 1$ [190–192]. In simple words, we need to find the parameters of model for the data regarding to each class using maximum likelihood approach which is equivalent to finding the parameters. By applying the above mentioned constraint on log-likelihood, the expression for class probability is as follows:

$$p(c_k) = \frac{1}{N} \sum_{i=1}^{N} (Y_i = c_k) \tag{3.46}$$

The detailed computation of class probability by ensuring the above mentioned constraints is given in [193, 194]. The parameters estimation relating to class conditional probability is discussed in following subsections.

3.8.2 Laplace Naive Bayes Classifier

For a feature vector $\vec{X}_i$ with independent and identically distributed features, Laplace distribution is represented as follows:

$$p(\vec{X}_i|\vec{\mu}_k, \vec{b}_k) = \prod_{d=1}^{D} \frac{1}{2b_{kd}} \exp\left[-\frac{|X_{id} - \mu_{kd}|}{b_{kd}}\right] \tag{3.47}$$

where $\vec{\mu}_k$ and $\vec{b}_k$ are mean and scale parameters of Laplace distribution for class c_k of the data, respectively. If we place Laplace distribution in Eq. (3.45), and maximize it with respect to mean and scale parameters, we can estimate the value of these parameters as follows:

$$\hat{\mu}_{kd} = \frac{\sum_{i=1:Y_i=c_k}^{N} \frac{X_{id}}{|X_{id} - \mu_{kd}|}}{\sum_{i=1:Y_i=c_k}^{N} \frac{1}{|X_{id} - \mu_{kd}|}} \tag{3.48}$$

$$\hat{b}_{kd} = \frac{\sum_{i=1:Y_i=c_k}^{N} |X_{id} - \mu_{kd}|}{\sum_{i=1}^{N} (Y_i = c_k)} \tag{3.49}$$

3.8.3 Generalized Gaussian Naive Bayes Classifier

If we consider that $\vec{X}$ having D-dimensions, as feature vector by having the assumption of independent and identically distributed features, then generalized Gaussian distribution can be expressed

as follows:

$$p(\vec{X}_i|\vec{\mu}_k,\vec{\sigma}_k,\vec{\lambda}_k) \tag{3.50}$$
$$= \prod_{d=1}^{D} \frac{\lambda_{kd}\sqrt{\frac{\Gamma(3/\lambda_{kd})}{\Gamma(1/\lambda_{kd})}}}{2\sigma_{kd}\Gamma(1/\lambda_{kd})} \exp\left(-A(\lambda_{kd})\left|\frac{X_{id}-\mu_{kd}}{\sigma_{kd}}\right|^{\lambda_{kd}}\right)$$

$$\text{with} \qquad A(\lambda_{kd}) = \left[\frac{\Gamma(3/\lambda_{kd})}{\Gamma(1/\lambda_{kd})}\right]^{\lambda_{kd}/2} \tag{3.51}$$

where $\vec{\mu}_k, \vec{\sigma}_k$ and $\vec{\lambda}_k$ are mean, standard deviation and shape parameters of generalized Gaussian distribution for class c_k of the data, respectively. By using this distribution in Eq. (3.45), we can estimate the parameters of the Naive Bayes classifier for generalized Gaussian distribution by maximum likelihood estimate as follows:

$$\hat{\mu}_{kd} = \frac{\sum_{i=1:Y_i=c_k}^{N} |X_{id}-\mu_{kd}|^{\lambda_{kd}-2} X_{id}}{\sum_{i=1:Y_i=c_k}^{N} |X_{id}-\mu_{kd}|^{\lambda_{kd}-2}} \tag{3.52}$$

$$\hat{\sigma}_{kd} = \left[\frac{\lambda_{kd}A(\lambda_{kd})\sum_{i=1:Y_i=c_k}^{N} |X_{id}-\mu_{kd}|^{\lambda_{kd}}}{\sum_{i=1}^{N}(Y_i=c_k)}\right]^{1/\lambda_{kd}} \tag{3.53}$$

For the shape parameter, a closed form solution does not exist and it will be estimated using Newton's-Raphson method for each class of the data as follows:

$$\hat{\lambda}_{kd} \simeq \lambda_{kd} - \left[\left(\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Y}|\theta_k)}{\partial {\lambda^2}_{kd}}\right)^{-1}\left(\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Y}|\theta_k)}{\partial \lambda_{kd}}\right)\right]_{Y_i=c_k} \tag{3.54}$$

The computations for derivative of log-likelihood with respect to parameter of generalized Gaussian distributions is given in [47]. In order to get optimized value of parameters, Expectation Maximization (EM) algorithm can be applied to Naive Bayes classifier in a similar way as described in [191, 192]. In the initialization phase, parameter values are set according to the assumption of Gaussian distribution for both algorithms. The scale parameter for Laplace Naive Bayes classifier is computed from standard deviation in initialization phase as $b = \sigma^2/\sqrt{2}$. The value of shape parameter is set to 2 during the initialization phase for generalized Gaussian Naive Bayes classifier. The rest of the parameters are initialized with the Gaussian distribution assumption.

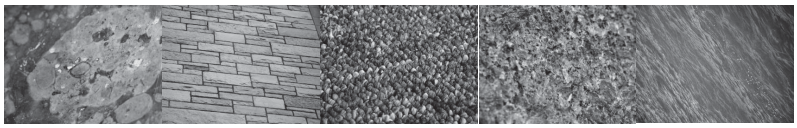


Figure 3.23: Sample images of UIUC dataset

Table 3.13: Performance of UIUC texture data categorization based on different metrics

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
GenGNB	85.00	85.00	98.33	89.46	01.67	85.74	85.04	87.20	91.42
LapNB	83.00	83.00	98.11	88.43	01.89	83.60	83.08	85.67	90.24
GNB	82.00	82.00	98.00	87.55	02.00	82.53	81.97	84.73	89.64

3.9 Texture Image Categorization

In this section, texture image categorization is performed by Naive Bayes classifiers introduced in Section 3.8. In the texture image categorization framework, features are extracted in wavelet domain using bounded Laplace mixture model (BLMM) as introduced in [85]. Once the features are extracted, Naive Bayes classifier can be applied for training and prediction of texture data classes. The whole framework is given in Figure (3.22), which serves the purpose of validation of proposed Naive Bayes classifiers. In [85], feature extraction using BLMM, was applied in image clustering and content based image retrieval in a unsupervised manner, In this chapter, image categorization framework also demonstrate the effectiveness of feature extraction in supervised manner. In the following subsections, BLMM and feature extraction technique are presented.

3.9.1 Feature Extraction in Wavelet Domain via BLMM

The 2-*D* wavelet transform is derived from its 1-*D* counterpart via separable wavelet filters. By applying 2-*D* transform, an image can be decomposed into four sub-bands which represent a scale-down low resolution image and diagonal, vertical and horizontal information [173]. From the studies, it is observed that wavelet coefficients in high frequency sub-bands are distributed in a Laplacian like density [159, 173] and for modeling the wavelet coefficients for representation of texture images, Gaussian mixture model (GMM), LMM and BLMM can be used [85, 159, 162]. In this work, BLMM is adopted for modeling the wavelet coefficients to represent the texture images. In the feature extraction, first step is to apply 2-*D* discrete wavelet transform on each

Table 3.14: Performance of KTH-TIPS texture data categorization based on different metrics

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
GenGNB	86.19	86.19	98.47	87.52	01.53	86.21	85.08	86.85	92.12
LapNB	83.33	83.33	98.15	84.65	01.85	83.22	81.86	83.99	90.44
GNB	80.95	80.95	97.88	81.97	02.12	80.55	79.01	81.46	89.02

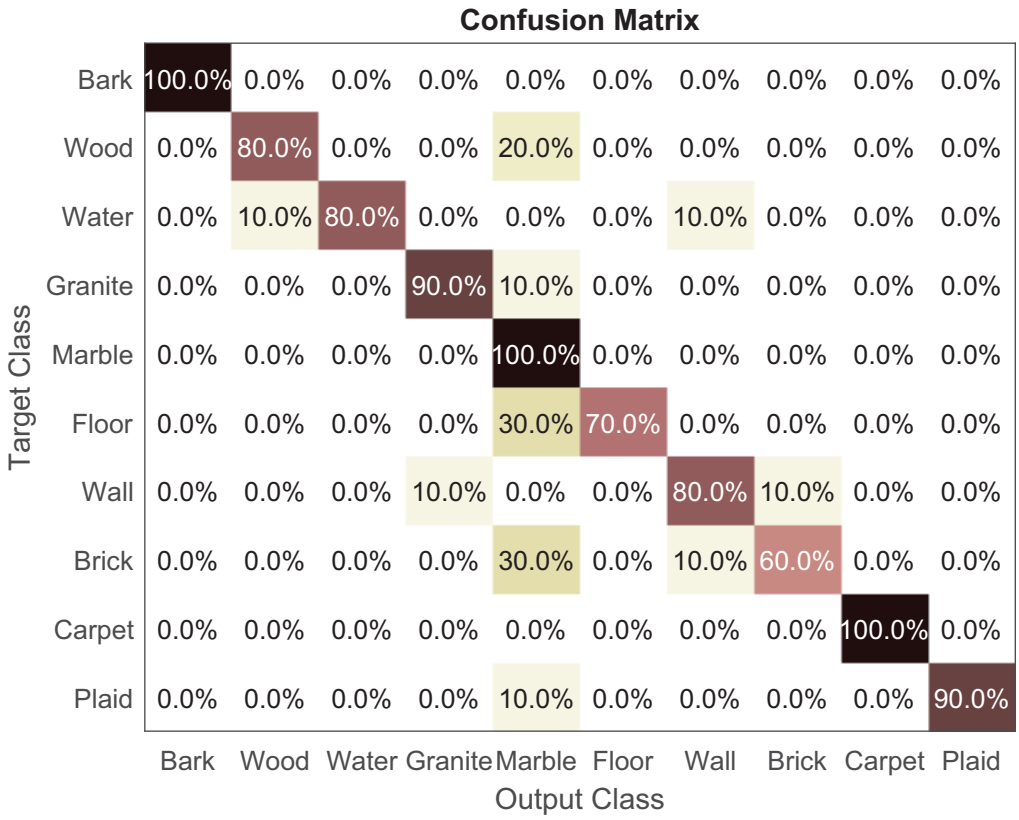


Figure 3.24: UIUC dataset with generalized GNB classifier

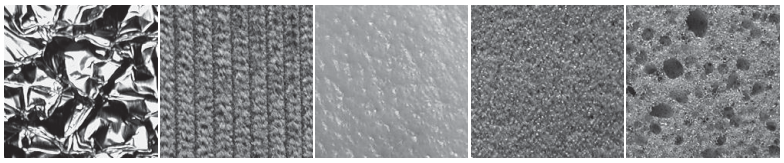


Figure 3.25: Sample images of KTH-TIPS dataset

image from the database which will decompose each image into four wavelet subspaces at each level of decomposition. For feature extraction from high frequency sub-bands, wavelet coefficients are modeled via BLMM with two mixture components centered at 0. The parameters learned from modeling wavelet coefficients are used as features for representing each image. If each wavelet subspace is assumed to have N coefficients, then BLMM can be represented with Eq. (3.7) modeled with two components centered at 0. The parameters learned for each sub-band (diagonal, vertical and horizontal) at each decomposition level are $\Theta = (b_1, b_2, p_1, p_2)$. For scaling subspace, mean is computed for its wavelet coefficients and used as feature along with parameters learned via BLMM.

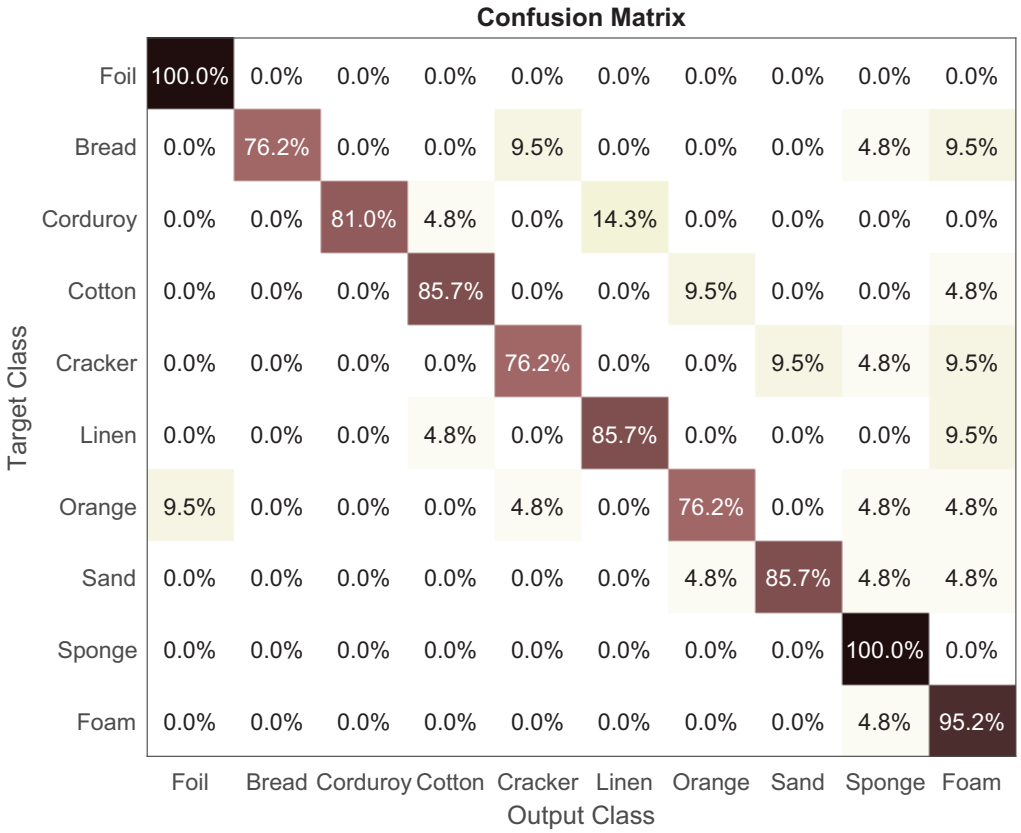


Figure 3.26: KTH-TIPS with generalized GNB classifier

For each image, the integrated feature space learned with BLMM is represented as follows:

$$\mathscr{F} = [F_{1H}, F_{1V}, F_{1D}, S_1, ..., F_{jH}, F_{jV}, F_{jD}, S_j] \tag{3.55}$$

where F represents the feature set $[p_1, p_2, b_1, b_2]$ of wavelet subspaces and S represents the mean value of coefficients in scaling subspace. The subscripts D, V, & H represent diagonal, vertical and horizontal directions, respectively, at each scale and subscript j express the number of decomposition scales in the image [85, 159, 162]. As a last step, each feature vector is normalized to avoid biasing.

Figure 3.27: Sample images of DTD dataset

Table 3.15: Performance of DTD texture data categorization based on different metrics

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
GenGNB	88.75	88.75	98.75	88.94	01.25	88.75	87.56	88.84	93.62
LapNB	84.50	84.50	98.28	84.82	01.72	84.55	82.89	84.66	91.13
GNB	80.25	80.25	97.81	80.64	02.19	80.26	78.18	80.44	88.59

3.10 Experiments and Results

3.10.1 Design of Experiments

In this section, experiments and results for texture image categorization via our introduced Naive Bayes classifier are presented. The experiments are conducted on UIUC, KTH-TIPS and DTD datasets. Parts of these datasets are selected to perform texture image categorization which will validate the performance of proposed Naive Bayes classifiers and demonstrate the effectiveness of feature representation via BLMM in wavelet domain for supervised learning. Haar wavelet filter is used for decomposition of images. For modeling the wavelet coefficients, BLMM is adopted and 3-level decomposition is used in all experiments for feature extraction. Wavelet coefficients are modeled with 2-component mixture of bounded Laplace distributions with zero mean during the feature extraction. The data is divided into testing and training before feature extraction and once the feature are obtained, the training part of the data is used to train the Naive Bayes classifier. The trained model is further applied to predict the classes of test data which will eventually categorize the texture data into desired classes. The complete experimental framework for each dataset, results and discussions are given in the following subsections.

3.10.2 Experimental Framework and Results: UIUC Dataset

UIUC dataset is a collection of texture images of 25 categories with 40 images in each class. For the experiments to categorize texture images via Naive Bayes classifier, we have chosen 10 classes. The dataset is divided into training and testing and 30 images from each class (300 images for 10 classes) are considered for training and 10 images from each class (100 images for 10 classes) are chosen for testing. A few sample images of UIUC dataset are given in Figure (3.23). Feature extraction is performed with BLMM in wavelet domain with 3-level decomposition. As first step, model is trained with Laplace Naive Bayes classifier for validation of modeling capability of

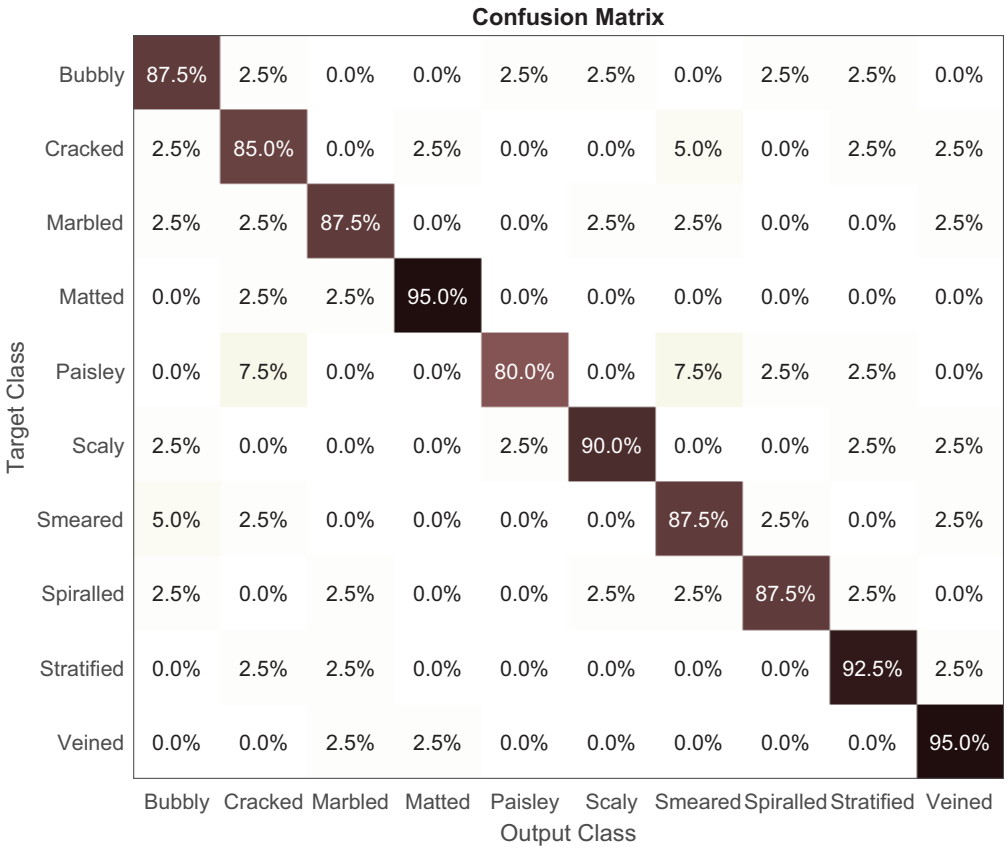


Figure 3.28: DTD dataset with generalized GNB classifier

feature representation in wavelet domain extracted with learning of BLMM. Once Laplace Naive Bayes classifier is trained by using the training part of data, it further adopted to predict the categories of test data. In this experiment, we achieved 83% accuracy in correctly classifying the texture data in particular categories. For a comparison, we also train the model with Gaussian Naive bayes classifier and on testing, the accuracy obtained is 82% for correct classification of texture images. We also have computed several other performance metrics for both models and it is observed by comparison of results of both models that Laplace Naive Bayes classifier has better modeling capability for feature representation via BLMM in wavelet domain. The results of both experiments are presented in Table (3.13). Since generalized Gaussian distribution has the capability to model the data with both Laplace and Gaussian distributions, it is more appropriate to adopt generalized Gaussian distribution for developing a Naive Bayes classifier in this application for texture image categorization. The classification results (85% accuracy) obtained by applying generalized Gaussian Naive Bayes classifier have outperformed both results previously

discussed in this experiment which is a proof of concept that peaky nature of data in wavelet domain can be better modeled by choosing the model with appropriate distribution which is Laplace and generalized Gaussian distribution. Complete comparison of results for all models is presented in Table (3.13) and confusion matrix to present the classification performance in each class using generalized Gaussian mixture Naive Bayes classifier is presented in Figure (3.24).

3.10.3 Experimental Framework and Results: KTH-TIPS Dataset

KTH-TIPS dataset is a collection of texture images of 10 classes with 81 images in each class. For our experiments, whole dataset is chosen and it is divided into training and testing. For training, 60 images from each classes (600 images for 10 classes) are chosen and 21 images from each class (210 images for 10 classes) are selected for testing. A few sample images of KTH-TIPS dataset are provided in Figure (3.25). The features are extracted in a similar manner as described in the experiment for UIUC dataset. Once features are extracted, Laplace Naive Bayes classifier is applied to train the model with training part of dataset and predict the test data into different texture image classes. The classification accuracy in this case 83.33% which is higher than Gaussian Naive Bayes classifier in a similar setting (80.95%). The results are presented in Table (3.14), and it is observed that Laplace Naive Bayes classifier has outperformed Gaussian Naive Bayes classifier in modeling the features represented by BLMM extracted from images in wavelet domain. We further applied generalized Gaussian Naive Bayes classifier for categorization of texture images and 86.19% accuracy is observed. The detailed performance metrics are provided in Table (3.14), it is observed that generalized Gaussian Naive Bayes classifier has performed better than both algorithms previously observed in this experiment. A confusion matrix to present detailed classification performance using generalized Gaussian Naive Bayes classifier is given in Figure (3.26).

3.10.4 Experimental Framework and Results: DTD Dataset

DTD dataset is a collection of texture images with 47 classes and having 120 images in each class. For our experiment, 10 classes are selected with 80 images per class (800 images for 10 classes) for training and 40 images per class (400 images for 10 classes) for training. A few sample images of DTD dataset are presented in Figure (3.27). The feature are extracted in a similar manner described for previous two experiments. Once the features are obtained, training data are used to train the Laplace Naive Bayes classifier and testing data are adopted for examining the performance of trained model in texture image categorization. In this experiment, 84.50% accuracy is achieved and for a comparison to observe the effectiveness of this approach, Gaussian Naive Bayes classifier is also trained and tested in a similar setting for texture image categorization and

80.25% accuracy is observed. The detailed results with other performance metrics are given in Table (3.15), which indicates the effectiveness of Laplace Naive Bayes classifier in texture image categorization. The experiment is further extended to observe the performance of generalized Gaussian Naive Bayes classifier in a similar setting and 88.75% accuracy is observed. From Table (3.15), by observing all the performance metrics, it is evident that our generalized Gaussian Naive Bayes classifier has outperformed other two algorithms in modeling the texture images data represented by BLMM extracted from wavelet domain representation. A confusion matrix obtained by generalized Gaussian Naive Bayes classifier is given in Figure (3.28), which exhibit the detailed classification performance of DTD dataset in this experiment.

From the set of experiments on UIUC, KTH-TIPS and DTD texture datasets, it is observed that our proposed models have preformed better than Gaussian Naive Bayes classifier which is a proof of the concept that modeling the data by choosing the model and probability based on the nature of the data can give better performance. It is also observed that by increasing the size of training data, the performance of our proposed models is increased which is evident from experiments on KTH-TIPS and DTD dataset.

3.11 Discussion about Naive Bayes Classifiers

In this section, Naive Bayes classifiers based on Laplace and generalized Gaussian distribution are introduced. These algorithms are applied in texture image categorization and the deriving force to introduce these algorithms is the nature of data in wavelet domain representation of texture images. In this approach, wavelet domain images are modeled through BLMM for feature extraction and Naive Bayes classifier is applied for image categorization after the feature extraction. In order to validate the proposed approaches, different experiments are conducted on texture datasets. From the set of experiments, it is observed that Naive Bayes classifier with Laplace distribution is definitely a better choice for this application as compared to Naive Bayes classifier with Gaussian distribution. It is also observed that Naive Bayes classifier with generalized Gaussian distribution has the best performance are compared to other two approaches. For the validation of framework, different performance metrics are considered and proposed approaches have shown their effectiveness for texture image categorization with feature extracted in wavelet domain.

Chapter 4

Multivariate Bounded Generalized Gaussian Mixture Model with ICA

In this chapter, we propose bounded generalized Gaussian mixture model (BGGMM) with independent component analysis (ICA). One limitation in ICA is that it assumes the sources to be independent from each other. This assumption can be relaxed by employing a mixture model. In our proposed model, bounded generalized Gaussian distribution (BGGD) is adopted for modeling the data and we have further extended its mixture as an ICA mixture model by employing gradient ascent along with expectation maximization for parameter estimation. By inferring the shape parameter in BGGD, Gaussian and Laplace distributions can be characterized as special cases. In order to validate the effectiveness of this algorithm, experiments are performed on unsupervised keyword spotting, speaker classification, blind source separation (BSS) and BSS as pre-processing to unsupervised keyword spotting. For speaker classification, TSP and TIMIT speech datasets are adopted and keyword spotting framework is developed with TIMIT speech corpus. For BSS, TIMIT, TSP and Noizeus speech corpora are selected and results are compared with ICA. For keyword spotting, recognition results are further compared before and after BSS being applied as pre-processing when speech utterances are affected by mixing of noise or other speech utterances. The mixing of noise or speech utterances with a particular or target speech utterance can greatly affect the intelligibility of a speech signal. The results achieved from the presented experiments on different applications have demonstrated the effectiveness of ICA mixture model in statistical learning.

4.1 Introduction

In machine learning and pattern recognition, effectiveness of an approach or an algorithm is determined by the ability of modeling underlying distribution of observed data [195]. Finite mixture models have been extensively used for statistical modeling in machine learning and pattern recognition and have demonstrated their importance in many speech and image processing applications [19, 20]. Gaussian mixture model (GMM) is well renowned for data clustering. The parameters of GMM can be estimated effectively using expectation maximization (EM) algorithm by maximizing the log-likelihood function [18, 193]. The main problem associated with GMM is sensitivity to outliers [18]. Student's-t mixture model (SMM) has been proposed in order to improve the robustness of Gaussian mixture model for statistical modeling [16, 45, 46]. In SMM, each component has one more parameter, called degree of freedom, as compared to GMM. Cauchy and Gaussian distributions are special cases of student's-t distribution with degree of freedom 1 and ∞, respectively [18]. There have been substantial growth in research for developing mixture models using generalized Gaussian distribution (GGD) [47–51]. This distribution has one extra parameter (shape parameter λ) than Gaussian distribution, which controls the tails of distribution. One problem associated with above mentioned mixture models is unbounded support range $(-\infty,+\infty)$ of their distributions [18]. It is observed that many real application have their data within bounded support regions [60–62]. For speech processing application, bounded Gaussian mixture model (BGMM) has been proposed in [61, 62]. The idea of bounded support mixture is adopted for GGMM and BGGMM has been proposed in [18], which provides a generalization for GMM, Laplace mixture model (LMM), GGMM and BGMM as special cases.

ICA mixture model has been proposed as an extension of Gaussian mixture model in [195–197]. ICA has been successfully applied to problems such as blind source separation and signal analysis describing its ability to model non-Gaussian statistical structures. If the source distributions are assumed to be Gaussian, it is equivalent to principle component analysis (PCA), which assumes that observed data is distributed as a multivariate Gaussian [195]. ICA generalize PCA by modeling the observed data with non-Gaussian distributions and goal is to linearly transform the data structures in such a way that variables after transformation are independent from each other [196]. One limitation in ICA is that it assumes the sources to be independent from each other. This assumption can be relaxed by employing a mixture model. The observed data can be categorized into several mutually exclusive classes by employing a mixture model [198], simply called an ICA mixture model. It can be generalized with the assumption that observed data in each class is produced by a linear combination of independent, non-Gaussian sources as in case of ICA [196]. Hence, in an ICA mixture model, it is assumed that observed data can be categorized into mutually exclusive classes and components of the model are generated by linear combination

of independent sources [199]. Many variations of ICA mixture model have been proposed in the last few years [200–203]. It has been extensively used for statistical modeling in a variety of applications that include segmentation, image enhancement and BSS [196, 197, 204]. In [205], ICA mixture model was proposed with adaptive source densities including generalized Gaussian and Student's t distributions as special cases along with other forms of densities. In this chapter, we are interested in extending the model presented in [195] with BGGD. In [18], BGGMM is formulated for univariate data which is extended here for multivariate data. The parameter estimation for proposed ICA mixture is adopted from [195, 196] using ICA and gradient ascent. The preliminary results obtained by applying the proposed ICA mixture are published in [2, 87]. In this chapter we have extended the applications of ICA mixture in BSS and unsupervised keyword spotting frameworks for more insightful analysis.

Automatic speech recognition (ASR) is considered as a nonlinear transformation from spoken words to text [3, 206], which requires large quantities of annotated data along with the language specific speech and text data, used for training complex statistical acoustic and language models [1, 207, 208]. The problem associated with these techniques is the lack of valuable linguistic information related to majority of the languages (termed as under-resourced), especially if they are not frequently used [1, 209–211]. Many languages of the world are categorized as under-resourced which refers to lack of unique writing system, limited linguistic expertise, unavailability of the electronic resources for language processing and lack of on-line resources [209–211]. There are about 6900 spoken languages in the world [209, 212] and despite a lot of development in the ASR, the availability of only few (50-100) commercial ASR engines leads to the need to develop unsupervised methods that do not require any annotated or labeled data [213]. Keyword spotting task has also been explored for many years and ASR is used to detect the occurrence of a specific keyword in speech data [214]. Keyword spotting is defined as an approach for speech understanding to detect specific keyword(s) that most likely express the intent of a speaker rather than recognition of a whole speech utterance [215]. Hidden Markov models based keyword spotting methods have been proposed widely for supervised and unsupervised settings [216–220]. Dynamic time warping has been used extensively for speech recognition and keyword spotting [26, 221–227]. The use of mixture model in automatic speech recognition and keyword spotting has demonstrated its effectiveness in unsupervised platforms and settings [228, 229]. The proposed model is used for statistical learning from the training data given in TIMIT speech corpus [143]. The trained model is used to decode the keyword example(s) and test utterances in posteriorgrams. The posteriorgrams generated from keywords examples and unseen test utterances are compared using segmental DTW. The distortion scores are further processed to select the best matching candidate for the keyword hits. The TIMIT speech dataset is used to tune the parameters of the described

unsupervised keyword spotting system. The results achieved from the experiment demonstrates the effectiveness and viability of the proposed algorithm in keyword spotting.

Speaker classification is a fundamental component of speaker recognition systems which performs two alternative tasks: speaker identification and verification. The goal of speaker identification is to label an unknown speech file with a speaker identity. The task of speaker verification is to validate and confirm the claim of a speaker about its identity [131, 132]. Speaker classification has been used in human-machine dialog systems, forensics, medical and many other applications. One interesting application of speaker classification is in the speech recognition and keyword spotting as preprocessing to reach the speaker of interest which is further useful in many security applications. Mixture models have been widely adopted to address the speaker classification task [29]. Recently Mixture model have been employed to address the object recognition and classification tasks through clustering in [230, 231]. A two level hierarchical clustering framework based on inverted Dirichlet mixture model is presented in [232] which is selected for object clustering and recognition. In this work, the same hierarchical clustering framework is adapted using bounded generalized Gaussian mixture model (BGGMM) with ICA and employed for speaker classification. In this chapter, gender and 10 speakers classification is performed through the hierarchical clustering framework using ICA mixture model. Bounded generalized Gaussian mixture model with ICA is applied for the statistical learning of the clustering framework. Speaker classification based on supervised hierarchical clustering also serves the purpose to validate the effectiveness of ICA mixture model in speaker recognition and statistical learning. The gender speaker classification is performed on TIMIT and TSP speech databases and 10 speakers classification is conducted on TSP speech database. Both classification frameworks are also implemented using Gaussian mixture model in order to compare the performance of ICA mixture model in statistical learning. It is observed that classification framework based on hierarchical clustering performs well for both classification scenarios and ICA mixture model outperforms the GMM in model learning based on the classification rate. It is also observed that conventional problem of female speaker recognition is improved by employing multi-cluster model instead of classical model during the learning.

Blind source separation has been applied to many signal processing and machine learning problems including speech enhancement, speech recognition, medical signal processing and telecommunications [196]. BSS is defined as a method which reconstruct the unknown sources of observed signals from an unknown mixture [233–236]. BSS was formulated around 1982 and first related contributions appeared around 1985 in [237–241]. The ICA was proposed as general framework for solving blind source separation problems based on statistical independence of the unknown sources in [242] and formalized for linear mixtures in [243, 244]. The limitations associated with ICA were controlled by ICA mixture as proposed in [195, 245] and successfully

applied to BSS [196, 198, 246]. Research for the development of many new approaches for BSS is continued and many interesting algorithms and techniques have been developed [247, 248]. The Expectation-Maximization (EM) algorithm has also been applied to ICA in [249, 250]. In this chapter, we have proposed BGGMM using ICA for the task of BSS. For the evaluation of proposed BSS framework, we have used signal-to-distortion ratio (SDR), signal-to-interference ratio (SIR), signal-to-artifact ratio (SAR) and perceptual evaluation of speech quality (PESQ). The detailed explanation of evaluation metrics is presented in [251–254].

In many real time scenarios, speech signals are mixed with noise or other speech signal which reduces the intelligibility of signals in keyword spotting and speech recognition. In order to improve the detection rate in keyword spotting, speech signal can be pre-processed using BSS before being applied to the trained model for keyword detection or speech recognition. The proposed ICA mixture have demonstrated its effectiveness in BSS as described in Subsection 4.5 and we have proposed the same BSS framework as prepossessing to unsupervised keyword spotting presented in [2]. Due to mixing of speech utterances, two types of problems occur in keyword spotting. In the first case, target keyword will more likely not detected during the keyword spotting, whereas in second case target keyword will be detected in correct speech utterance but it will also get detected in other speech utterances as false alarm. These two problems are explained in detail in Subsection 4.6. In this chapter, we have also proposed BSS as pre-processing to unsupervised keyword spotting as an extension to the keyword spotting framework with ICA mixture described previously.

4.2 Bounded Generalized Gaussian Mixture Model with ICA

In this section, bounded generalized Gaussian mixture model with ICA is presented. In an ICA mixture model, it is assumed that observed data come from a mixture model and it can be categorized into mutually exclusive classes which means that each class of the data is modeled via an ICA [195, 199]. Consider the case where the input is a set of features of the data represented as $\mathscr{X} = (\vec{X}_1, ..., \vec{X}_N)$ and $\vec{X}_i$ is a D-dimensional random variable $\vec{X}_i = [X_{i1}, ..., X_{iD}]^T$. The $\vec{X}_i$ follows a K components mixture distribution if its probability function can be written as Eq. (1.1) provided that $p_j \geq 0$ and $\sum_{j=1}^{K} p_j = 1$. In Eq. (1.1), $p(\vec{X}_i|\xi_j)$ is probability density function, ξ_j represents the set of parameters defining jth component, p_j is mixing proportion, $\Theta = (\xi_1, ..., \xi_K, p_1, ..., p_K)$ is complete set of parameters to characterize the mixture model and $K \geq 1$ is number of components in the mixture model [30–32]. For an ICA mixture model, each data vector $\vec{X}_i$ can be represented as:

$$\vec{X}_i = \mathrm{A}_j \vec{\mathrm{s}}_{j,i} + \vec{\mathrm{b}}_j \tag{4.1}$$

where A_j is $L \times D$ scalar matrix termed as basis functions, $\vec{\mathrm{s}}_{j,i}$ is D-dimensional source vector and $\vec{\mathrm{b}}_j$ is an L-dimensional bias vector for a particular mixture component j [195–197, 199, 202, 203]. In order to define the BGGD for a variable $\vec{X} \in \mathbb{R}$, it is required to provide an indicator function which introduces the boundary conditions. For each component (denoted by j), indicator function $\mathrm{H}(\vec{X}_i|j)$ is defined with bounded support region (∂_j) for each component:

$$\mathrm{H}(\vec{X}_i|j) = \begin{cases} 1 & \text{if} \quad \vec{X}_i \in \partial_j \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

For BGGMM, $\vec{X}_i$ follows a K components mixture represented in Eq. (1.1), where $p(\vec{X}_i|\xi_j)$ is multivariate BGGD as:

$$p(\vec{X}_i|\xi_j) = \frac{f_{ggd}(\vec{X}_i|\xi_j)\mathrm{H}(\vec{X}_i|j)}{\int_{\partial_j} f_{ggd}(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}} \tag{4.3}$$

where term $f_{ggd}(\vec{X}_i|\xi_j)$ represents the multivariate generalized Gaussian distribution (GGD):

$$f_{ggd}(\vec{X}_i|\xi_j) = \prod_{d=1}^{D} \frac{\lambda_{jd}\sqrt{\frac{\Gamma(3/\lambda_{jd})}{\Gamma(1/\lambda_{jd})}}}{2\sigma_{jd}\Gamma(1/\lambda_{jd})} \exp\left(-A(\lambda_{jd})\left|\frac{X_{id}-\mu_{jd}}{\sigma_{jd}}\right|^{\lambda_{jd}}\right) \tag{4.4}$$

with

$$A(\lambda_{jd}) = \left[\frac{\Gamma(3/\lambda_{jd})}{\Gamma(1/\lambda_{jd})}\right]^{\lambda_{jd}/2} \tag{4.5}$$

The term $\int_{\partial_j} f_{ggd}(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}$ is normalization constant that indicates the share of $f_{ggd}(\vec{X}_i|\xi_j)$ which belongs to the support region. Note that $\xi_j = \left\{\vec{\mu}_j, \vec{\sigma}_j, \vec{\lambda}_j, \mathrm{A}_j, \vec{\mathrm{b}}_j\right\}$ is the set of parameters defining jth component, where $\vec{\mu}_j = (\mu_{j1}, ..., \mu_{jD})$, $\vec{\sigma}_j = (\sigma_{j1}, ..., \sigma_{jD})$, $\vec{\lambda}_j = (\lambda_{j1}, ..., \lambda_{jD})$, $\mathrm{A}_j = (\mathrm{a}_1, ..., \mathrm{a}_L)$ and $\vec{\mathrm{b}}_j = (\mathrm{b}_{j1}, ..., \mathrm{b}_{jD})$ are the mean, standard deviation, shape parameters, basis functions and bias vector, respectively. The vectors representing mean, standard deviation, shape parameters and bias are D-dimensional for each component of the mixture model, whereas the basis functions for each component has L number of linear combination with each linear combinations being D-dimensional. For simplicity, number linear combinations (L) is considered to be equal to the number of sources (D) in each observation which makes basis functions a $D \times D$ scalar matrix. With a mixture of K BGGDs, the likelihood of data $\mathscr{X}$ can be defined as Eq. (1.2), where the complete set of parameters of the ICA mixture model having K classes is defined by $\Theta = (\vec{\mu}_1, ..., \vec{\mu}_K, \vec{\sigma}_1, ..., \vec{\sigma}_K, \vec{\lambda}_1, ..., \vec{\lambda}_K, \mathrm{A}_1, ..., \mathrm{A}_K, \vec{\mathrm{b}}_1, ..., \vec{\mathrm{b}}_K, p_1, ..., p_K)$. We introduce the stochastic indicator $Z = \{\vec{Z}_1, ..., \vec{Z}_N\}$, where $\vec{Z}_i = (Z_{i1}, ..., Z_{iK})$ is the label of each observation, such that $Z_{ij} \in \{0, 1\}$, $\sum_{j=1}^{K} Z_{ij} = 1$. The role of these variables is to encode the membership of each observation for a relative component of the mixture model. In other words, Z_{ij}, the unobserved variable in each indicator vector equals 1 if $\vec{X}_i$ belongs to class j and 0, otherwise [32, 56, 193]. The complete

data likelihood is:

$$p(\mathscr{X}, Z|\Theta) = \prod_{i=1}^{N}\prod_{j=1}^{K}\Big(p(\vec{X}_i|\xi_j)p_j\Big)^{Z_{ij}} \tag{4.6}$$

For instance, if we consider that number of mixture components is known, the parameter estimation requires the maximization of log-likelihood function:

$$\mathscr{L}(\Theta, Z, \mathscr{X}) = \sum_{i=1}^{N}\sum_{j=1}^{K} Z_{ij}\log\Big(p(\vec{X}_i|\xi_j)p_j\Big) \tag{4.7}$$

By replacing each Z_{ij} by its expectation, defined as posterior probability that the ith observation belongs to jth component of the mixture model we obtain:

$$\hat{Z}_{ij} = p(j|\vec{X}_i) = \frac{p(\vec{X}_i|\xi_j)p_j}{\sum_{j=1}^{K} p(\vec{X}_i|\xi_j)p_j} \tag{4.8}$$

4.2.1 Parameters Estimation

In a mixture model, the parameters include mixing proportions and parameters of the distribution whereas in case of ICA mixture model each vector of the data is represented as in Eq. (4.1), which also necessitates the estimation of basis functions and bias vectors. The basis functions and bias vectors are further adopted to compute the sources in ICA model. For the parameters mean, standard deviation and mixing proportions, maximization of log-likelihood is obtained by setting the gradient of log-likelihood (with respect to each parameter) to zero. The maximization of log-likelihood for the shape parameters, basis functions and bias vector is performed by using the standard ICA model and gradient ascent. Using Eq. (4.8), each observation can be labeled to one or zero for a particular component of the mixture model which can be further applied to maximize the complete data log-likelihood with respect to the parameters of ICA mixture model. The gradient of log-likelihood with respect to parameters of each component is computed as following:

$$\nabla_{\Theta_j}\mathscr{L}(\Theta, Z, \mathscr{X}) = \nabla_{\Theta_j}\sum_{i=1}^{N}\sum_{j=1}^{K} Z_{ij}\log\Big(p(\vec{X}_i|\xi_j)p_j\Big) \tag{4.9}$$

The ∇_{Θ_j} represents here the gradient with respect to p_j, $\vec{\mu}_j$, $\vec{\sigma}_j$, $\vec{\lambda}_j$, A_j and $\vec{\mathrm{b}}_j$. Eq. (4.9) can be written as:

$$\nabla_{\Theta_j}\mathscr{L}(\Theta, Z, \mathscr{X}) = \nabla_{\Theta_j}\sum_{i=1}^{N}\sum_{j=1}^{K} Z_{ij}\left\{\log p_j + \log f_{ggd}(\vec{X}_i|\xi_j) + \log \mathrm{H}(\vec{X}_i|j) - \log\int_{\partial_j} f_{ggd}(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right\} \tag{4.10}$$

4.2.1.1 Estimation of Mixing Parameter, Mean and Standard Deviation

The estimation of mixing parameter is provided in Section 2.2.3.1. The mean μ_j can be estimated by maximizing the log-likelihood with respect to μ_j. The gradient of the log-likelihood and estimation of μ_j are given in Appendix C.1 & C.2, respectively. The estimated mean $\hat{\mu}_{jd}$ for $d = 1,...,D$ is given by:

$$\hat{\mu}_{jd} = \frac{1}{\sum_{i=1}^{N} \hat{Z}_{ij} \left|X_{id} - \mu_{jd}\right|^{(\lambda_{jd}-2)}} \sum_{i=1}^{N} \hat{Z}_{ij} \times \left\{ \left[\left|X_{id} - \mu_{jd}\right|^{(\lambda_{jd}-2)} X_{id} \right] - \left[\frac{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)\mathrm{sign}\left(\mathrm{u}-\mu_{jd}\right)\left|\mathrm{u}-\mu_{jd}\right|^{\lambda_{jd}-1} d\mathrm{u}}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j) d\mathrm{u}} \right] \right\} \tag{4.11}$$

Note that, in Eq. (4.11), the term $\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)\mathrm{sign}\left(\mathrm{u}-\mu_{jd}\right)\left|\mathrm{u}-\mu_{jd}\right|^{\lambda_{jd}-1} d\mathrm{u}$ is the expectation of function $\mathrm{sign}\left(\mathrm{u}-\mu_{jd}\right)\left|\mathrm{u}-\mu_{jd}\right|^{\lambda_{jd}-1}$ under the probability distribution $f_{ggd}(\mathrm{u}|\xi_j)$ [18, 60, 193], which can be approximated as:

$$\begin{aligned} &\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)\mathrm{sign}\left(\mathrm{u}-\mu_{jd}\right)\left|\mathrm{u}-\mu_{jd}\right|^{\lambda_{jd}-1} d\mathrm{u} \\ &\approx \frac{1}{M}\sum_{m=1}^{M} \mathrm{sign}(\mu_{jd} - s_{j_{md}})\left|\mu_{jd} - s_{j_{md}}\right|^{\lambda_{jd}-1} \mathrm{H}(s_{j_{md}}|j) \end{aligned} \tag{4.12}$$

where $s_{m_{jd}} \sim f_{ggd}(\mathrm{u}|\xi_j)$ is a set of random variables drawn from the bounded generalized Gaussian distribution for the particular component of the mixture model j. The set of data with random variables have M vectors with D dimensions. M is a large integer chosen to generate the set of random variables. Similarly, the term $\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u}$ in Eq. (4.11) can be approximated as:

$$\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u} \approx \frac{1}{M}\sum_{m=1}^{M} \mathrm{H}(s_{m_{jd}}|j) \tag{4.13}$$

From Eqs. (4.12) and (4.13), $\hat{\mu}_j$ can be written as:

$$\hat{\mu}_{jd} = \frac{1}{\sum_{i=1}^{N} \hat{Z}_{ij} \left|X_{id} - \mu_{jd}\right|^{(\lambda_{jd}-2)}} \sum_{i=1}^{N} \hat{Z}_{ij} \times \left\{ \left[\left|X_{id} - \mu_{jd}\right|^{(\lambda_{jd}-2)} X_{id} \right] - \left[\frac{\sum_{m=1}^{M} \mathrm{sign}(\mu_{jd} - s_{j_{md}})\left|\mu_{jd} - s_{j_{md}}\right|^{\lambda_{jd}-1} \mathrm{H}(s_{j_{md}}|j)}{\sum_{m=1}^{M} \mathrm{H}(s_{j_{md}}|j)} \right] \right\} \tag{4.14}$$

with $i = 1,...,N,\ j = 1,...,K,\ d = 1,...,D$ and $m = 1,...,M$. The standard deviation σ_j can be estimated by maximizing the log-likelihood with respect to σ_j and gradient of log-likelihood and estimation of σ_j are given in Appendix C.3 & C.4, respectively. The estimated standard deviation

$\hat{\sigma}_{jd}$ for $d = 1, ..., D$ is given as:

$$\hat{\sigma}_{jd} = \left(\frac{\sum_{i=1}^{N} Z_{ij} \left[A(\lambda_{jd}) \left| X_{id} - \mu_{jd} \right|^{\lambda_{jd}} \lambda_{jd} \right]}{\sum_{i=1}^{N} Z_{ij} \left\{ 1 + \left[\frac{\int_{\partial_{\Omega_j}} f_{ggd}(\mathrm{u}|\xi_j) \left\{ -1 + A(\lambda_{jd}) \left| X_{id} - \mu_{jd} \right|^{\lambda_{jd}} \lambda_{jd} (\sigma_{jd})^{-\lambda_{jd}} \right\} d\mathrm{u}}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j) d\mathrm{u}} \right] \right\} } \right)^{1/\lambda_{jd}} \tag{4.15}$$

Similar to Eq. (4.12), in Eq. (4.15) the term $\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)(-1 + A(\lambda_{jd}) \left| X_{id} - \mu_{jd} \right|^{\lambda_{jd}} \lambda_{jd} (\sigma_{jd})^{-\lambda_{jd}}) d\mathrm{u}$ can be approximated as:

$$\begin{aligned} &\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)(-1 + A(\lambda_{jd}) \left| X_{id} - \mu_{jd} \right|^{\lambda_{jd}} \lambda_{jd} (\sigma_{jd})^{-\lambda_{jd}}) d\mathrm{u} \\ &\approx \frac{1}{M} \sum_{m=1}^{M} (-1 + \lambda_{jd} A(\lambda_{jd}) \left| s_{mjd} - \mu_{jd} \right|^{\lambda_{jd}} (\sigma_{jd})^{-\lambda_{jd}}) \mathrm{H}(s_{mjd}|j) \end{aligned} \tag{4.16}$$

From Eqs. (4.16) and (4.13), $\hat{\sigma}_j$ can be written as:

$$\hat{\sigma}_{jd} = \left(\frac{\sum_{i=1}^{N} Z_{ij} \left[A(\lambda_{jd}) \left| X_{id} - \mu_{jd} \right|^{\lambda_{jd}} \lambda_{jd} \right]}{\sum_{i=1}^{N} Z_{ij} \left\{ 1 + \left[\frac{\sum_{m=1}^{M} (-1 + \lambda_{jd} A(\lambda_{jd}) \left| s_{mjd} - \mu_{jd} \right|^{\lambda_{jd}} (\sigma_{jd})^{-\lambda_{jd}}) \mathrm{H}(s_{mjd}|j)}{\sum_{m=1}^{M} \mathrm{H}(s_{mjd}|j)} \right] \right\} } \right)^{1/\lambda_{jd}} \tag{4.17}$$

with $i = 1, ..., N$, $j = 1, ..., K$, $d = 1, ..., D$ and $m = 1, ..., M$.

4.2.1.2 Parameter Estimation using ICA and Gradient Ascent

For parameter estimation using ICA and gradient ascent, zero mean and unit variance is assumed which is fundamental assumption of the source in ICA. The parameters estimated using ICA with gradient ascent include basis functions, bias vector and shape parameters. The gradient of complete data log-likelihood for the parameters of each class is given below:

$$\nabla_{\Theta_j} \mathscr{L}(\Theta, Z, \mathscr{X}) = \sum_{i=1}^{N} \sum_{j=1}^{K} p(j|\vec{X}_i) \nabla_{\Theta_j} \log \left(p(\vec{X}_i|\xi_j) p_j \right) \tag{4.18}$$

The ∇_{Θ_j} represents here the gradient with respect to basis function, bias vector and shape parameter.

$$\nabla_{\Theta_j} \mathscr{L}(\Theta, Z, \mathscr{X}) = \sum_{i=1}^{N} \sum_{j=1}^{K} p(j|\vec{X}_i) \left(\nabla_{\Theta_j} \log p(\vec{X}_i|\xi_j) + \nabla_{\Theta_j} \log p_j \right) \tag{4.19}$$

The term $\nabla_{\Theta_j} \log p_j$ will become zero while taking gradient with respect to basis functions, bias vector and shape parameter which will lead us to :

$$\nabla_{\Theta_j} \mathscr{L}(\Theta, Z, \mathscr{X}) = \sum_{i=1}^{N} \sum_{j=1}^{K} p(j|\vec{X}_i) \left(\nabla_{\Theta_j} \log p(\vec{X}_i|\xi_j) \right) \tag{4.20}$$

The class log-likelihood $\log p(\vec{X}_i|\xi_j)$ in Eq. (4.20) can be estimated using standard ICA model as follows:

$$\log p(\vec{X}_i|\xi_j) = \log \frac{p(\vec{\mathrm{s}}_{j,i})}{\left|\det \mathrm{A}_j\right|} \tag{4.21}$$

The source can be computed by applying estimated basis function and bias vector in the above equation and log-likelihood of the standard ICA model will become:

$$\log p(\vec{X}_i|\xi_j) = \log p(\mathrm{A}_j{}^{-1}(\vec{X}_i - \vec{\mathrm{b}}_j)) - \log\left|\det \mathrm{A}_j\right| \tag{4.22}$$

Basis Functions Estimation: The adaptation of basis functions for each component of ICA mixture is performed by maximizing the log-likelihood with respect to basis functions A_j for each component of mixture model:

$$\nabla_{\mathrm{A}_j}\mathscr{L}(\Theta, Z, \mathscr{X}) = \sum_{i=1}^{N} p(j|\vec{X}_i)\nabla_{\mathrm{A}_j}\log p(\vec{X}_i|\xi_j) \tag{4.23}$$

The adaptation performed by the gradient ascent with respect to the basis functions is given as:

$$\Delta \mathrm{A}_j \propto p(j|\vec{X}_i)\frac{\partial}{\partial \mathrm{A}_j}\log p(\vec{X}_i|\xi_j) \tag{4.24}$$

The derivative in Eq. (4.24) can be computed using standard ICA learning algorithm given in [196] and it also described in Appendix C.6.

$$\frac{\partial}{\partial \mathrm{A}_j}\log p(\vec{X}_i|\xi_j) = \mathrm{A}_j\left[\mathrm{I} - 2\tanh(\vec{\mathrm{s}}_{j,i})\vec{\mathrm{s}}_{j,i}^{T}\right] \tag{4.25}$$

By using the standard ICA model for log-likelihood, we get:

$$\Delta \mathrm{A}_j \propto p(j|\vec{X}_i)\mathrm{A}_j\left[\mathrm{I} - 2\tanh(\vec{\mathrm{s}}_{j,i})\vec{\mathrm{s}}_{j,i}^{T}\right] \tag{4.26}$$

In the adaptation of basis functions, the gradient of component of the mixture model with respect to basis functions is weighted by $p(j|\vec{X}_i)$. An estimate of the basis functions using gradient ascent is as follows:

$$\hat{\mathrm{A}}_j = \mathrm{A}_j + \alpha\left(p(j|\vec{X}_i)\mathrm{A}_j\left[\mathrm{I} - 2\tanh(\vec{\mathrm{s}}_{j,i})\vec{\mathrm{s}}_{j,i}^{T}\right]\right) \tag{4.27}$$

where α is step size and source is represented as:

$$\vec{\mathrm{s}}_{j,i} = \mathrm{A}_j{}^{-1}(\vec{X}_i - \vec{\mathrm{b}}_j) \tag{4.28}$$

Bias Vectors Estimation: The adaptation of the bias vector can be performed for each component of the mixture model by using the Eq. (4.20).

$$\nabla_{\mathrm{b}_{jd}} \mathscr{L}(\Theta, Z, \mathscr{X}) = \sum_{i=1}^{N} p(j|\vec{X}_i) \nabla_{\mathrm{b}_{jd}} \log p(\vec{X}_i|\xi_j) \tag{4.29}$$

The gradient ascent is used for the adaptation, with the gradient of the component density with respect to bias term b_{jd} for each component of the mixture model:

$$\Delta \mathrm{b}_{jd} \propto p(j|\vec{X}_i) \frac{\partial}{\partial \mathrm{b}_{jd}} \log p(\vec{X}_i|\xi_j) \tag{4.30}$$

Eq. (4.22) can be applied in Eq. (4.30) to adapt the bias term:

$$\Delta \mathrm{b}_{jd} \propto p(j|\vec{X}_i) \frac{\partial}{\partial \mathrm{b}_{jd}} \left[\log p(\mathrm{A}_j{}^{-1}(\vec{X}_i - \vec{b}_j)) - \log \left| \det \mathrm{A}_j \right| \right] \tag{4.31}$$

An approximate method can also be applied for the adaptation of bias vectors instead of applying gradient. For approximate method, maximum likelihood estimate must satisfy the following condition:

$$\sum_{i=1}^{N} p(j|\vec{X}_i) \nabla_{\Theta_j} \log p(\vec{X}_i|\hat{\xi}_j) = 0 \tag{4.32}$$

The bias term b_{jd} can be adapted as follows:

$$\nabla_{\mathrm{b}_{jd}} \mathscr{L}(\Theta, Z, \mathscr{X}) = 0, \quad \Rightarrow \quad \sum_{i=1}^{N} p(j|\vec{X}_i) \nabla_{\mathrm{b}_{jd}} \log p(\vec{X}_i|\xi_j) = 0 \tag{4.33}$$

By substituting Eq. (4.22) into Eq. (4.33), it is clear that gradient of the $\log p(\mathrm{A}_j{}^{-1}(\vec{X}_i - \vec{\mathrm{b}}_j))$ must be zero as given in Eq. (4.34).

$$\nabla_{\mathrm{b}_{jd}} \log p(\mathrm{A}_j{}^{-1}(\vec{X}_i - \vec{\mathrm{b}}_j)) = 0 \tag{4.34}$$

In the adaptation of bias vector, if we assume that we have a large amount of data and that the prior probability distribution function of the source is differentiable and symmetric, then the $\log p(\mathrm{A}_j{}^{-1}(\vec{X}_i - \vec{\mathrm{b}}_j))$ will be symmetric as well and the bias vector $\vec{\mathrm{b}}_j$ will be approximated by the weighted average of data samples as:

$$\vec{\mathrm{b}}_j = \frac{\sum_{i=1}^{N} \vec{X}_i p(j|\vec{X}_i)}{\sum_{i=1}^{N} p(j|\vec{X}_i)} \tag{4.35}$$

Shape Parameter Estimation: For the estimation of parameters in ICA mixture model, unit variance and zero mean is assumed. For the purpose of estimation of shape parameter, same

assumption is adopted and the problem will become the estimation of shape parameter from the data. The gradient ascent is used to estimate the shape parameter by maximizing the log-likelihood:

$$\nabla_{\lambda_{jd}} \mathscr{L}(\Theta, Z, \mathscr{X}) = \sum_{i=1}^{N} p(j|\vec{X}_i) \nabla_{\lambda_{jd}} \log p(\vec{X}_i|\xi_j) \tag{4.36}$$

The gradient ascent is used for the adaptation, with the gradient of the component density with respect to shape parameter vector λ_{jd} for each component of the mixture model.

$$\Delta\lambda_{jd} \propto p(j|\vec{X}_i) \frac{\partial}{\partial \lambda_{jd}} \log p(\vec{X}_i|\xi_j) \tag{4.37}$$

In the adaptation of shape parameter λ_{jd}, the gradient of component of the mixture model with respect to shape parameter is weighted by $p(j|\vec{X}_i)$. An estimate of the shape parameter using gradient ascent is as follows:

$$\hat{\lambda}_{jd} = \lambda_{jd} + \alpha \left(p(j|\vec{X}_i) \frac{\partial}{\partial \lambda_{jd}} \log p(\vec{X}_i|\xi_j) \right) \tag{4.38}$$

The estimation of shape parameter in an ICA mixture model is discussed in [195] and the term $\frac{\partial}{\partial \lambda_{jd}} \log p(\vec{X}_i|\xi_j)$ is computed with the assumption of unit variance and zero mean as follows:

$$\begin{aligned} \frac{\partial}{\partial \lambda_{jd}} \log p(X_{id}|\xi_j) &= \frac{\partial}{\partial \lambda_{jd}} \log \left[\frac{f_{ggd}(X_{id}|\xi_j) \mathrm{H}(X_{id}|j)}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j) d\mathrm{u}} \right] \\ &= h(X_{id}|\xi_j) - \frac{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j) h(\mathrm{u}|\xi_j) d\mathrm{u}}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j) d\mathrm{u}} \end{aligned} \tag{4.39}$$

where the term $h(X_{id}|\xi_j)$ is represented as:

$$\begin{aligned} h(X_{id}|\xi_j) &= \frac{\partial}{\partial \lambda_{jd}} \log f_{ggd}(X_{id}|\xi_j) \\ &= \left[\frac{1}{\lambda_{jd}} + \frac{3}{2\lambda_{jd}} \left[\Psi(1/\lambda_{jd}) - \Psi(3/\lambda_{jd}) \right] \right] - A(\lambda_{jd}) |X_{id}|^{\lambda_{jd}} \log |X_{id}| \\ &- A(\lambda_{jd}) \left(\frac{1}{2} \log \frac{\Gamma(3/\lambda_{jd})}{\Gamma(1/\lambda_{jd})} + \frac{1}{2\lambda_{jd}} \left[\Psi(1/\lambda_{jd}) - 3\Psi(3/\lambda_{jd}) \right] \right) |X_{id}|^{\lambda_{jd}} \end{aligned} \tag{4.40}$$

The term $h(\mathrm{u}|\xi_j)$ also follows the computation presented in Eq. (4.40). The term $\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)$ $h(\mathrm{u}|\xi_j) d\mathrm{u}$ can be approximated similar to Eq. (4.12).

$$\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j) h(\mathrm{u}|\xi_j) d\mathrm{u} \approx \frac{1}{M} \sum_{m=1}^{M} h(s_{j_{md}}|\xi_j) \mathrm{H}(s_{j_{md}}|j) \tag{4.41}$$

The estimation of shape parameter can be expressed as follows:

$$\hat{\lambda}_{jd} = \lambda_{jd} + \alpha \left[p(j|\vec{X}_i) \left\{ h(X_{id}|\xi_j) - - \frac{\sum_{m=1}^{M} h(s_{j_{md}}|\xi_j) \mathrm{H}(s_{j_{md}}|j)}{\sum_{m=1}^{M} \mathrm{H}(s_{j_{md}}|j)} \right\} \right] \tag{4.42}$$

The complete procedure for estimation of shape parameter is discussed in Appendix C.5. The complete learning procedure for BGGMM with ICA is given in Algorithm 4, where t_{min} is minimum threshold used to examine convergence criteria in each iteration.

Algorithm 4 Model Learning with BGGMM with ICA

Input:Dataset $\mathscr{X} = \{\vec{X}_1, \ldots, \vec{X}_N\}$, t_{min}.
Output: Θ.
{Initialization}:
  K-Means Algorithm (Computation of $\vec{\mu}_1, \ldots, \vec{\mu}_K$ & cluster assignment)
    for all $1 \leq j \leq K$ do
      Computation of p_j
      Computation of $\{\vec{\sigma}_j\}$
      Set the $\{(\vec{\lambda}_j = 2\}$
    end for
{Expectation Maximization}:
while relative change in log-likelihood $\geq t_{min}$ do
  $\{$[**E Step**]$\}$:
    for all $1 \leq j \leq K$ do
      Compute $p(j|\vec{X}_i)$ for $i = 1, \ldots, N$. using Eq. (4.8).
    end for
  $\{$[**M step**]$\}$:
    for all $1 \leq j \leq K$ do
      start ICA Algorithm
        Update the basis functions A_j using Eq. (4.27).
        Update the bias vector $\vec{b}_j$ using Eq. (4.35).
        Update shape parameter $\vec{\lambda}_j$ using Eq. (4.42).
      end ICA
      Update the mixing parameter p_j using Eq. (2.12).
      Update the mean $\vec{\mu}_j$ using Eq. (4.14).
      Update standard deviation $\vec{\sigma}_j$ using Eq. (4.17).
    end for
end while

4.3 Unsupervised Keyword Spotting using ICA Mixture Model

In this section, bounded generalized Gaussian mixture model (BGGMM) using independent component analysis (ICA) is applied to an existing unsupervised keyword spotting setting for the generation of posteriorgrams. The ICA mixture model is trained without any transcription information to generate the posteriorgrams which further labels the speech frames of the keyword example(s) and test data. For the detection of occurrence of a specific keyword in the test data, the posteriorgrams of one or more keyword examples are compared with the posteriorgrams of test utterances using the segmental dynamic time warping (DTW). A score fusion method is used to obtain the result of the keyword detection by ranking the distortion scores of all the test utterances. TIMIT speech corpus is used for the evaluation of this unsupervised keyword spotting setting. The keyword detection results demonstrate the viability and effectiveness of the proposed algorithm in unsupervised keyword spotting framework.

4.3.1 Experiments and Results

4.3.1.1 Design of Experiments

In this section, experimental framework and the detection results based on unsupervised keyword spotting framework reported in [1] are presented. However, instead of using independently trained phonetic recognizer or GMM, bounded generalized Gaussian mixture model using ICA is employed for training the model and generation of posteriorgrams. The training process involves directly modeling the speech without any transcription information using the proposed ICA mixture model. The trained model is used to decode the keyword examples and test utterances in posteriorgrams. The segmental DTW is used to compare the posteriorgrams between keyword examples and the test utterances. The distortion scores are ranked for the most reliable warping path to achieve keyword detection [1, 255]. The detailed description on the keyword spotting is provided in [1]. In the experimental setup, the parameters of the keyword spotting framework are chosen exactly the same as given in [1], in order to have fair comparison of the keyword detection results. The unsupervised keyword spotting framework with ICA mixture model is shown in Fig. (4.1).

4.3.1.2 Experimental Framework and Results

The unsupervised keyword spotting framework is evaluated on TIMIT speech corpus. The TIMIT speech corpus consists of 6300 speech utterances which contains 4620 speech utterances for training and 1680 speech utterances for testing. Each speech utterance is segmented into frames of 25

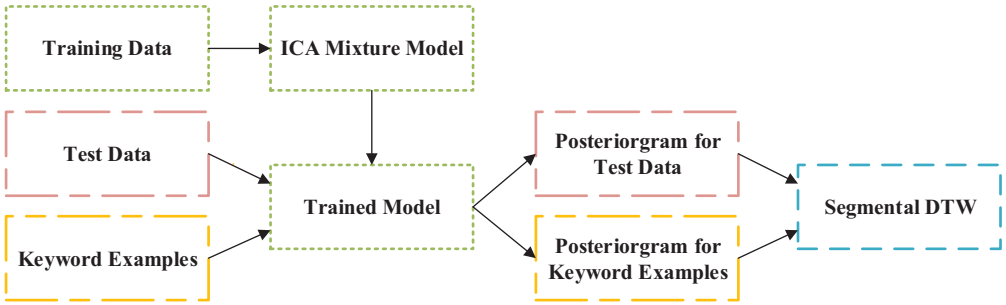


Figure 4.1: Unsupervised Keyword Spotting with ICA Mixture Model [3]

ms with a window shifting of 10 ms, where each frame is represented by 13 MFCCs. The K-Means is used to initialize the parameters of ICA mixture model, with shape parameter set to 2 for each component of the mixture model. The number of components for the mixture model is chosen to be 50 as in [1]. The other parameters for this framework are smoothing factor λ, segmental DTW adjustment window size and score weighing factor α. The smoothing factor is part of discounting based smoothing strategy applied to move small portion of probability mass from non-zero to zero dimensions. The segmental DTW adjustment window size is used to prevent the warping process from going too far or behind in warping path. The score weighing factor α is used to vary the averaging function in the voting based score merging and ranking [1]. The smoothing factor λ, segmental DTW adjustment window size and score weighing factor α are chosen to be 0.00001, 6 and 0.5 respectively, in order to have the same keyword spotting scenario as in [1]. For testing, 10-keywords set presented in [1] is used and given in Table 4.9. Table 4.10 summarizes the keyword detection performance for different number of keyword examples. For the evaluation of keyword detection, three different evaluation matrices reported in [1, 255] are examined, which are defined as: (1) the average precision for the top 10 utterance hits termed as P@10, (2) the average precision for the top N utterance hits termed as P@N, where N is equal to the number of occurrences of the each keyword in the test data, (3) the average equal error rate (EER), where false acceptance rate is equal to false rejection rate. For the P@10 evaluation, 4 keywords from Table 4.9 are considered because only they have occurred more than 10 times both in the training and the test dataset. For P@N and EER evaluations, with one keyword example experiment, all the keywords from Table 4.9 are used. For P@N and EER evaluations, with 5 keyword examples experiment, two keywords are not used because they have occurred less than 5 times in the training set. For P@N and EER evaluations, with 10 keyword examples experiment, 5 keywords are not used because they have occurred less than 10 times in the training set. The average precision for each keyword is calculated first and then mean of average precisions of all keywords for P@10 or P@N is computed.

Table 4.1: TIMIT 10 Keyword List used in [1]

age(3:10)	warm(10:8)	year(11:177)	money(19:17)
artists(7:7)	problem(22:9)	children(18:15)	
surface(3:7)	development(9:8)	organizations(7:7)	

The EER for each keyword is computed based on false acceptance rate (FAR) and false rejection rate (FRR). The EER mentioned in Table 4.10 is the average of EER for all keywords used for that particular case. Table 4.10 indicates considerable improvement in the evaluation matrices from one keyword example to 5 keyword examples. The trend of improvement is slow from 5 to 10 keyword examples. Table 4.3 presents the results ranked on the basis of EER in the 5-example experiment. For the 5-example experiment, 8 keywords are used and the ranking indicates that words with more syllables tended to have better performance. The word "organizations" have 5 syllables and the word "development" have 4 syllables and they present better performance in the recognition. The keyword spotting framework is adapted from [1], but the experimental results do not provide direct comparison since framework presented in [1] has used two databases named as TIMIT and MIT lecture corpus (TIMIT database is employed in this work) and the performance is mainly evaluated with MIT lecture corpus which is not publicly available. Nevertheless, a superficial comparison with the results presented in [1] is possible since the same evaluation matrices are computed. In [1], the P@10 evaluation does not use TIMIT database at all and P@N evaluation is mostly based on MIT lecture corpus. In this work, P@10 matrix is computed with 4 keywords, since only 4 keywords from the list of keywords presented in [1] have occurred more than 10 times in the training and test data. The P@10 performance (64.87%) presented in this work is comparable with their result (68.3%) while they have used 30 keywords from the MIT lecture corpus. For the P@N and EER evaluations, they have used 10 keywords from the TIMIT speech corpus and 30 keywords from the MIT lecture corpus which have occurred more than 100 times for most of the keywords in both training and test data. Although apparently, the results for P@N and EER evaluations are higher from their results (P@N : 58.27% vs. 39.3%, EER : 12.35% vs. 15.8%), but the small database used in this work for P@N and EER evaluations limit the comparison and it is further required to perform this experiment with a larger vocabulary database in order to have fair comparison. However, the keyword detection results based on average precision (P@10 and P@N), EER and the ranking of several keywords based on EER, validate the productiveness and viability of the proposed algorithm for statistical modeling.

Table 4.2: Evaluation matrix for different number of keyword examples

# of Examples	P@10	P@N	EER
1	28.37%	26.43%	29.20%
5	57.75%	51.39%	13.79%
10	64.87%	58.27%	12.35%

Table 4.3: Ranking of Keywords by EER for 5 No. of examples

organizations(6.1%)	development(6.7%)	childern(11.3%)	problem(12.6%)
artists(13.5%)	money(15.8%)	warm(21.4%)	year(22.9%)

4.3.2 Discussion about Keyword Spotting

In this section, BGGMM with ICA is presented as a model for statistical learning and used for unsupervised keyword spotting. In the proposed keyword spotting framework, speech data are modeled using BGGMM and ICA. The proposed model is used for the generation of posteriorgrams for the keyword examples and test data. The segmental DTW is used to compare the posteriorgrams and voting based score merging strategy presented in [1] is employed for determining the detection results. TIMIT speech corpus is used for the evaluation of this keyword spotting framework. The keyword detection results based on average precision (P@10 and P@N), EER and ranking of several keywords based on EER validate the effectiveness of proposed algorithm. Although keyword detection results are encouraging and competitive as compared to the results reported in [1], more simulations are needed with a larger vocabulary database to demonstrate further the effectiveness of proposed algorithm.

4.4 Speaker Classification via Supervised Hierarchical Clustering using ICA Mixture Model

In this section, speaker classification using supervised hierarchical clustering is provided. Bounded generalized Gaussian mixture model with ICA is adapted for statistical learning in the clustering framework. In the presented framework ICA mixture model is learned through training data and the posterior probability is used to split the training data into clusters. The class label of the training data is further selected to mark each cluster into a specific class. The cluster-class information from the training process is taken as reference for the classification of test data into different speaker classes. This framework is employed for the gender and 10 speakers classification and TIMIT and TSP speech corpora are selected to validate and test the classification framework. This

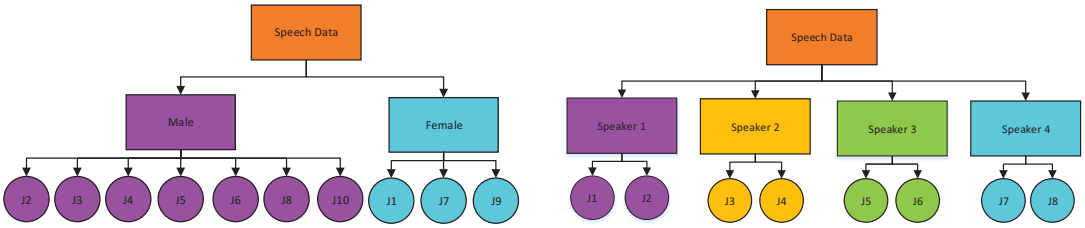
classification framework also validates the statistical learning of our recently proposed ICA mixture model. In order to examine the performance of the ICA mixture model, the classification results are compared with same framework using Gaussian mixture model. It is observed that: (i) presented clustering framework performs well for the speaker classification, (ii) ICA mixture model outperforms Gaussian mixture model in the statistical learning based on the classification accuracy for gender and multi-class scenarios.

4.4.1 Supervised Hierarchical Clustering via ICA Mixture Model

In this section, supervised hierarchical clustering framework based on ICA mixture model is presented, which is applied to the speaker classification. ICA mixture model is trained using training data and the posterior probability is employed to compute the specific cluster membership for each observation of the training data. The class label of training data is selected to decode the clusters into particular class. The posterior probability is computed for the testing data and cluster-class information from the training is employed to find the particular class for each observation of the testing data. Since the class label of the training data is used to decode the clusters into particular class and ICA mixture model is adapted for the statistical learning, therefore this framework is called the supervised hierarchical clustering framework based on ICA mixture model. Let us consider the training data represented as $\mathscr{X} = (\vec{X}_1, ..., \vec{X}_N)$, then complete data log-likelihood can be written as Eq. (4.7). By replacing each Z_{ij} by its expectation, that ith observation belongs to jth component of mixture model, posterior probability is defined as Eq. (5.43). The membership of $\vec{X}_i$ computed from the posterior probability can be selected to mark the clusters into a particular class. This information will further help for decoding the clusters into particular class for testing data using the membership function of the posterior probability for the observations of test data. If testing data is represented as $\mathscr{Y} = (\vec{Y}_1, ..., \vec{Y}_L)$, the posterior probability for $\vec{Y}_l$ can be computed using the trained mixture model and is represented as follows:

$$p(j|\vec{Y}_l) = \frac{p(\vec{Y}_l|\xi_j)p_j}{\sum_{j=1}^{K} p(\vec{Y}_l|\xi_j)p_j} \tag{4.43}$$

The supervised hierarchical framework for gender speaker classification is shown in Fig. (4.2a). The speech data contains the MFCC features for male and female speakers and the class label is also provided. The ICA mixture model is trained in unsupervised fashion and the posterior probability for each observation of the training data is computed. The posterior probability marks each observation to a specific cluster and the class information of the training data can be selected to mark each cluster to a specific class to which it belongs. For instance, if $\vec{X}_i$ belongs to the male class and it lies in the cluster 2, then cluster 2 is marked as male cluster. All the clusters can be marked as male or female from the training information and class label. In Fig. (4.2a), it is



(a) Gender Speaker (b) Multi-Speakers

Figure 4.2: Speaker Classification using Clustering

assumed that the ICA mixture model is learned with 10 mixture densities and we have the class label for each observation. From posterior probability it is inferred that female observations from the speech data belongs to cluster $J1$, $J7$ and $J9$, so these clusters can be further labeled as female class and rest of the clusters were inferred as male class in the same way. It is worth mentioning that training of the ICA mixture model is unsupervised because the speech data is adopted without any class label during the training. However, the clustering framework is supervised because class label is employed after the training to mark the clusters into specific class. In the 10 speakers classification, the same binary classification framework is extended for 10 classes (see Fig. (4.2)) and clusters obtained from the posterior probability are decoded into particular classes based on class label of the training data. In the classification using clustering, one important aspect is to accurately mark the number of classes representing data. In the classical approach, data is modeled by a fixed number of components of the mixture model which is equal to the number of classes. There are two problems associated with classical approach: (i) one single density component for each class does not necessarily fit the class data (ii) there is an overlap between the classes when using a single distribution to model each class [232]. In speaker recognition, while modeling several speakers in one class or even a single speaker in one class may have the above problems. This is because several speakers in a single class always have some distinct features and even same speaker will have dissimilar behavior while pronouncing the same words or utterances on different times. Due to the problems associated with classical model, we have adopted multi-cluster model which improve the learning of classification framework. There is another problem with the learning of female speakers and it is reported that speaker recognition performance of female speakers is almost worse as compare to the male speakers [256, 257]. It is observed that in multi-cluster modeling, the performance of female speakers is improved during learning for their particular class.

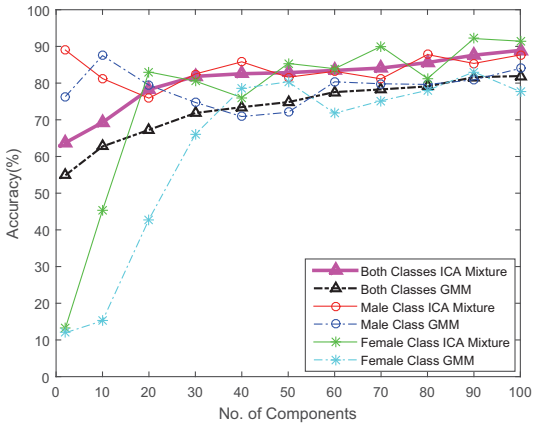
4.4.2 Experiments and Results

4.4.2.1 Design of Experiments

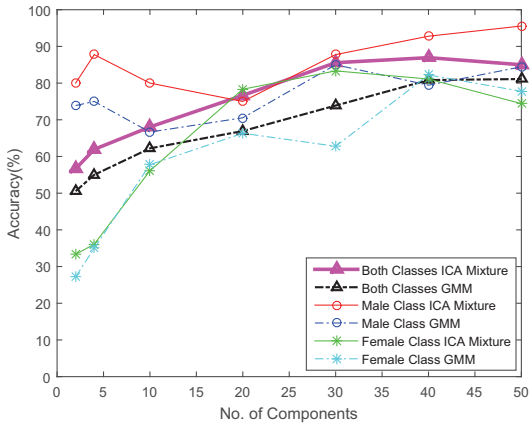
In this section, experimental framework for male/female and 10 speakers classification based on supervised hierarchical clustering is presented, which uses ICA mixture model for the statistical learning as described in section II. In the pre-processing stage, voice activity detection (VAD) is employed to distinguish between speech and non-speech parts of the speech sequences. By introducing the VAD in the pre-processing it is assured that the training of ICA mixture model is not inferred with the non-speech segments of the data set. The next stage is feature extraction and Mel frequency cepstral coefficients (MFCCs) are selected as features. MFCCs have demonstrated their effectiveness in speech recognition and speaker classification and we have computed 13 dimensional features same as standard hidden Markov model toolkit (HTK). The ICA mixture model is trained using training part of the speech databases and the posterior probability is employed to determine the membership of an observation to a particular cluster. The class label for the training data is adopted to decode the clusters into particular class. The posterior probability is computed for the testing data and clustering information from the training is selected to find the particular class for each observation of the testing data. This classification framework is called the supervised hierarchical clustering based on ICA mixture model and presented in a detail in section II. This framework is also implemented using Gaussian mixture model for comparison.

4.4.2.2 Experimental Framework and Results

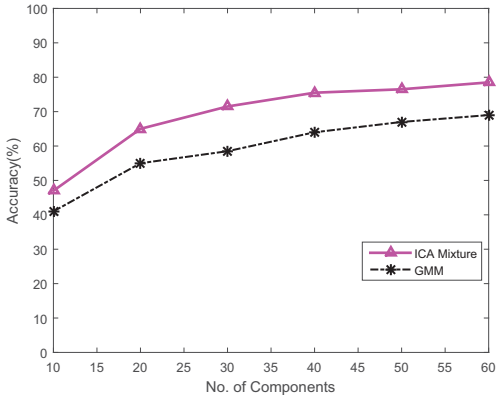
Speaker classification based on supervised hierarchical clustering is evaluated on TIMIT and TSP speech databases [91, 143]. The TIMIT speech corpus consists of 6300 speech utterances which contains 4620 speech utterances for training and 1680 speech utterances for testing. The TSP speech database consists of 1378 speech utterances spoken by 23 speakers (11 male, 12 female). For gender speaker classification, 6 speakers are selected for testing from the TSP and rest of the data is dedicated for training. For 10 speakers classification, 10 speakers (5 male, 5 female) having 60 speech utterances for each speaker are selected from the TSP with 40 speech utterances for training and 20 utterances for testing. The TIMIT speech corpus is employed for gender speaker classification whereas TSP database is selected for both classification scenarios. In the clustering framework for both scenarios, each speech utterance is segmented into frames of 25 ms with a window shifting of 10 ms, where each frame is represented by 13 MFCCs. The VAD is applied before feature extraction in order to have only speech frames in the training and testing data. The k-means is employed to initialize the parameters of ICA mixture model, with shape parameter set to 2 for each component of the mixture model. For the gender speaker classification, ICA mixture



(a) TIMIT (Gender)



(b) TSP (Gender)



(c) TSP (10 Speakers)

Figure 4.3: Classification Accuracy for Gender and 10 Speakers using ICA Mixture and GMM

model is trained using the training sets of both speech databases separately. From the posterior probability, speech utterances are divided into clusters by the membership of particular component of the mixture model. The class label for each utterance is provided for the training data which further leads to label the clusters into particular class. Once the clusters are labeled into the particular classes, the cluster-class information can be selected to decode the testing data into male/female speakers. The classification framework is evaluated using classification accuracy computed from the confusion matrices. For the TIMIT speech corpus, the classification accuracy is computed for different number of component of mixture model between 2-100 and plotted in Fig. (4.3a). In the classification accuracy curve for both classes, it is observed that by increasing the number of components of the mixture model, the classification rate is increased. However, after 30 components of the mixture model, the increase in classification accuracy is slow. The classification framework having ICA mixture model is compared with the same framework having GMM on the basis of

Table 4.4: 10 Speakers classification confusion matrix using TSP database.

(a) ICA Mixture, M=10

	MH	MI	MJ	MK	ML	FH	FI	FJ	FK	FL
MH	12	1	2	1	3	0	1	0	0	0
MI	2	9	1	4	1	1	0	1	0	1
MJ	1	3	11	1	2	0	1	0	0	1
MK	2	1	5	9	1	1	0	1	0	0
ML	1	1	2	1	10	1	1	1	2	0
FH	1	0	1	1	0	8	1	2	4	2
FI	0	1	0	2	1	5	7	1	1	2
FJ	0	0	1	1	0	0	1	12	2	3
FK	1	1	0	1	0	2	1	3	9	2
FL	1	1	0	1	0	1	2	5	2	7

(b) ICA Mixture, M=40

	MH	MI	MJ	MK	ML	FH	FI	FJ	FK	FL
MH	15	1	1	1	1	0	0	1	0	0
MI	0	13	2	2	1	1	0	0	0	1
MJ	1	1	17	1	0	0	0	0	0	0
MK	1	1	1	16	1	0	0	0	0	0
ML	0	1	0	1	18	0	0	0	0	0
FH	1	0	0	1	0	13	1	2	1	1
FI	0	1	0	0	0	1	15	1	1	1
FJ	0	0	1	0	0	1	1	14	1	2
FK	1	0	0	0	0	0	1	1	2	14	1
FL	0	0	0	0	1	1	0	1	1	16

(c) ICA Mixture, M=60

	MH	MI	MJ	MK	ML	FH	FI	FJ	FK	FL
MH	17	1	0	1	1	0	0	0	0	0
MI	1	16	1	0	1	1	0	0	0	0
MJ	0	1	18	0	0	1	0	0	0	0
MK	2	0	1	14	1	1	0	1	0	0
ML	0	1	2	1	13	1	1	0	0	1
FH	0	0	0	0	0	15	1	1	2	1
FI	0	0	0	0	0	1	17	1	0	1
FJ	0	1	0	0	0	1	0	16	1	1
FK	1	0	1	0	0	1	3	1	13	0
FL	0	0	0	0	0	1	1	0	0	18

classification rate. The overall classification rate for ICA mixture model in the setting of 100 mixture components is 88.92% whereas in same setting for GMM, the classification rate is 81.87%. It is also noted that for smaller number of mixture components, the recognition of female speakers is very poor which is improved for higher number of mixture components. It is also observed that multi-cluster model has improved the model learning for both classes as compared to the classic model. In the classic model, the female speakers have poor performance while fitting the data in one class. In comparison with GMM, ICA mixture model has performed well which validates the effectiveness of ICA mixture model for speaker classification and statistical learning. For the TSP speech database, the speech utterances from 17 speakers (8 male, 9 female) are adopted to train the ICA mixture model whereas 6 speakers (half male, half female) are employed for the testing with each speaker having 60 speech utterances. The classification accuracy for different number of components of ICA mixture model and GMM in gender speaker classification framework is computed and plotted in Fig. (4.3b). The highest value for overall classification accuracy is observed at 40 mixture components (86.94%) for ICA mixture model and at 50 mixture components (81.11%) for GMM. For the 10 class speaker classification TSP speech database is employed for tuning the speaker classification framework. In this scenario, 10 speakers are chosen and 40 speech utterances for each speaker are selected for training and 20 speech utterances for each speaker are adopted for testing. The classification results are computed for different number of mixture components and the resulting confusion matrices for classic and multi-cluster models are shown in Tables 4.4a, 4.4b and 4.4c. In order to have a comparison of ICA mixture model with GMM for 10 speakers classification, the same framework is implemented with GMM and overall classification rate is plotted for both models in Fig. (4.3c). The highest classification rate is observed at 60 mixture components for both scenarios of 10 speakers classification (78.50% for ICA mixture & 69% for GMM) which demonstrates the effectiveness of ICA mixture model in this setting.

4.4.3 Discussion about Speaker Classification

In this section, supervised hierarchical clustering framework is presented which is adopted for speaker classification. The first stage of the clustering is performed by the ICA mixture model

and in the second stage, clusters received from the posterior probability are further classified using the class label of the training data. The cluster-class label information from training process is used for the classification of testing data. The classification framework is validated on TIMIT and TSP speech corpora. This framework also validates the statistical learning of ICA mixture model proposed in [2]. In order to examine the performance of the ICA mixture model, the classification framework is also implemented with GMM and the classification accuracy in different modes is compared. The proposed framework having ICA mixture model is employed for gender and 10 speakers classification. It is concluded that supervised hierarchical clustering framework has performed considerably well for the speaker classification and ICA mixture model surpasses the GMM in the classification rate and model learning. It is also concluded that multi-cluster model has improved the problem of female speakers to fit the class data as compared to classic model.

4.5 Blind Source Separation

In this section, blind source separation (BSS) using ICA mixture model is provided. In BSS, mixing information of instantaneous linear mixtures is estimated which is further employed for recovering the source signals. The mixing information can be estimated using an ICA mixture model. In the proposed ICA mixture model setting we denote mixing matrix as basis function to avoid confusion from the mixture used in mixture model. For the viability and effectiveness of proposed framework, we have computed several objective measure which demonstrates the quality of speech signal after source separation. The details of the experiments and results are given in the following subsections.

4.5.1 Experiments and Results

4.5.1.1 Design of Experiments

In this subsection, experimental framework for BSS is described. It uses ICA mixture model for statistical learning as described in Section 4.2. In BSS, basis functions are estimated using ICA mixture model which is further applied to separate mixed signals. We have estimated basis functions 2×2, 3×3, 4×4 and 5×5 to compute 2, 3, 4 and 5 sources in separate experiments. In order to validate this BSS framework, TIMIT, TSP and NOIZEUS speech corpora are adopted during the experiments [91, 143, 258]. For BSS, only speech signal after linear mixing are observed. No prior information about basis functions is utilized during the source separation. BSS framework is evaluated using subjective and objective measures. Subjective analysis consists of speech signals before and after the source separation. Objective analysis consists of SDR, SIR, SAR and PESQ.

Objective measures SDR, SIR and SAR are measured in dB and PESQ score lies in the range -0.5 to 4.5. Further details on objective measures can be found in [251–253]. This framework is also implemented using ICA in order to compare and examine the validity of statistical learning of ICA mixture model in BSS. ICA used in this work is implemented using Infomax [259].

4.5.1.2 Experimental Results

Blind source separation based on ICA mixture model is validated using TIMIT, TSP and NOIZEUS speech corpora. We have conducted 4 experiments to compute 2, 3, 4 and 5 speech sources from this BSS framework. For the recovery of 2, 3, 4 and 5 speech sources, we have taken linear

Table 4.5: Objective measure for separation of 2 speech signals

Measure	TIMIT		TSP		NOIZEUS	
	ICA Mix	ICA	ICA Mix	ICA	ICA Mix	ICA
SDR (dB)	61.57	57.42	60.28	55.47	51.86	45.38
SIR (dB)	62.29	55.89	61.15	57.91	47.95	43.53
SAR (dB)	292.75	276.75	295.81	279.19	289.48	280.69
PESQ	2.40	1.80	2.30	1.90	2.35	2.15

Table 4.6: Objective measure for separation of 3 speech signals

Measure	TIMIT		TSP		NOIZEUS	
	ICA Mix	ICA	ICA Mix	ICA	ICA Mix	ICA
SDR (dB)	59.42	54.87	55.31	45.67	41.25	36.75
SIR (dB)	61.13	55.93	53.48	48.28	42.13	35.77
SAR (dB)	293.41	276.36	291.48	279.29	261.19	258.15
PESQ	2.35	1.65	2.40	1.80	2.20	1.90

Table 4.7: Objective measure for separation of 4 speech signals

Measure	TIMIT		TSP		NOIZEUS	
	ICA Mix	ICA	ICA Mix	ICA	ICA Mix	ICA
SDR (dB)	52.91	41.24	40.56	52.36	38.63	35.48
SIR (dB)	50.24	43.17	43.16	48.15	37.14	34.00
SAR (dB)	292.58	278.00	274.42	276.24	263.35	249.85
PESQ	2.10	2.00	1.90	2.15	2.20	1.85

mixture of 2, 3, 4 and 5 sources, respectively, from each database and performed blind source separation by employing BGGMM using ICA. Once the sources are recovered, objective analysis is performed on sources to examine quality of recovered speech signals and viability of ICA mixture

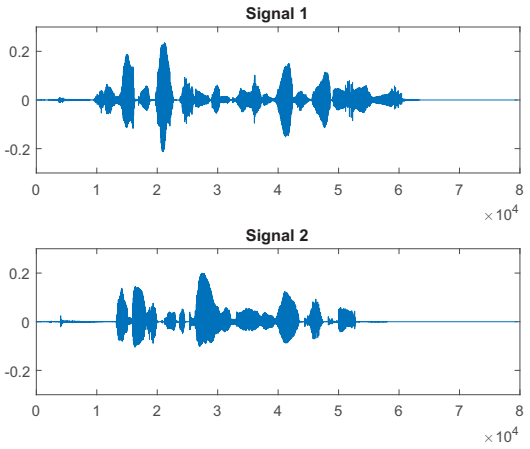
Table 4.8: Objective measure for separation of 5 speech signals

Measure	TIMIT		TSP		NOIZEUS	
	ICA Mix	ICA	ICA Mix	ICA	ICA Mix	ICA
SDR (dB)	51.87	46.44	52.92	40.55	49.75	38.04
SIR (dB)	52.65	47.16	49.21	39.49	50.71	40.63
SAR (dB)	291.29	277.32	285.03	276.55	271.19	260.12
PESQ	2.15	1.65	1.90	1.60	2.00	1.70

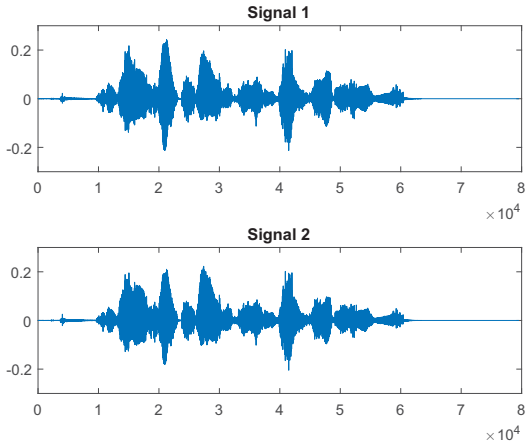
model in BSS. Objective measures include SDR, SIR, SAR and PESQ analysis. SDR is a measure of distortion in output signal and it is defined as ratio between energy of clean signal and distortion and it is measured in dB. SIR is the ratio of target signal prower to the interference signal. It measures the amount of undesired interference still present after BSS and it is measured in dB. SAR measures the quality after the source separation in terms of absence of artificial noise and measured in dB. PESQ is an objective assessment tool which correlates well with subjective listening scores [251–253]. The experiments are repeated 10 times with different linear speech mixtures of 2 and 3 sources from each database and average of objective measures is computed. In BSS for reconstruction of 4 and 5 sources, the experiments are repeated 10 times for TIMIT and TSP databases and 7 and 6 times (due to the limitation of database) for NOIZEUS speech corpus, respectively and average of the objective measures is computed. We have performed same analysis using ICA in order to have a comparison of proposed BSS framework. The objective measures after the recovery speech source signals are given in Table 4.5, 4.6, 4.7 & 4.8. From the objective measures, it is observed that ICA mixture model outperforms the ICA in a relative setting of BSS for 2, 3 and 5 sources for all databases and TIMIT and NOIZEUS speech corpora for recovery of 4 sources. However, in TSP database, ICA performs better for BSS in recovery of 4 sources. The speech signals before mixing, after mixing and after BSS are shown in Figs. (4.4 & 4.5).

4.5.2 Discussion About BSS

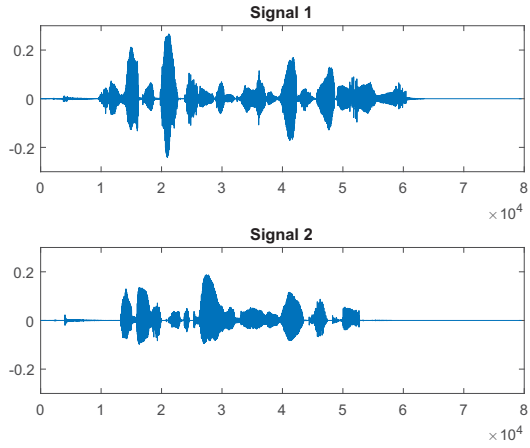
In this section, ICA mixture model is proposed as solution to BSS. For the validation of proposed framework, TIMIT, TSP and NOIZEUS speech corpora are selected. The BSS framework is evaluated using signal-to-distortion ratio (SDR), signal-to-interference ratio (SIR), signal-to-artifact ratio (SAR) and perceptual evaluation of speech quality (PESQ). From the above experiments on BSS using BGGMM using ICA, it is observed that ICA mixture model performs better as compared to ICA except for the recovery of 4 sources in TSP speech database. It is also observed that rate of this improvement becomes slower when we increase the number linear mixtures in source separation. From the objective measures and speech signals given in Figs. (4.4 & 4.5), BGGMM



(a) Unmixed Signals



(b) Before Separation



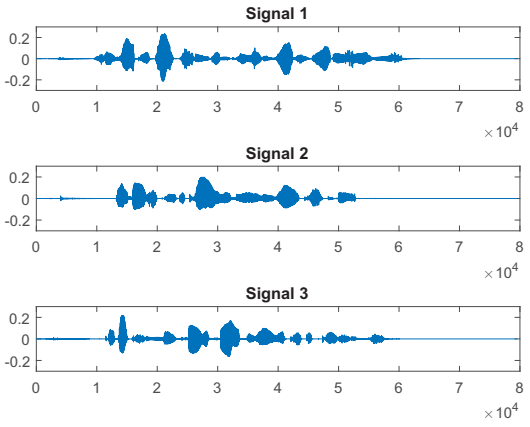
(c) After Separation

Figure 4.4: Blind Source Separation with 2 Signals

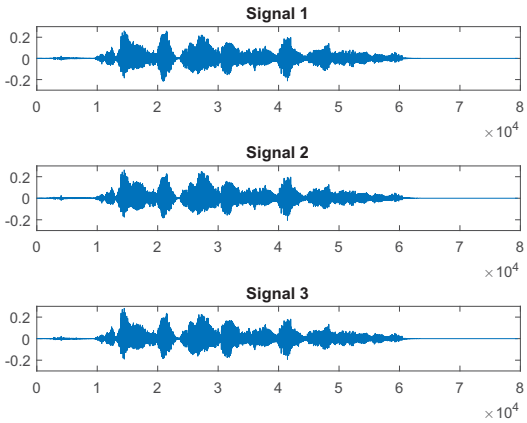
with ICA has demonstrated its success in BSS.

4.6 Blind Source Separation as preprocessing to Keyword Spotting

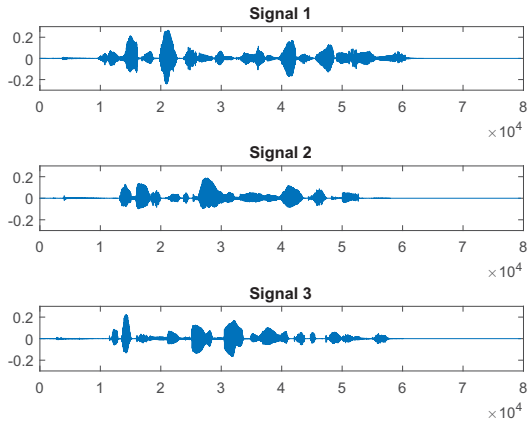
In this subsection, proposed framework for BSS as pre-processing to unsupervised keyword spotting using an ICA mixture is presented. In real time applications, detection rate of speech recognition and keyword spotting is badly affected by mixing of speech signals with noise or other speech signals. It is also possible to intentionally mix the speech signal with noise or some other speech utterances to reduce or some times completely eliminate the chances of getting spotted by keyword



(a) Unmixed Signals



(b) Before Separation



(c) After Separation

Figure 4.5: Blind Source Separation with 3 Signals

spotting systems. In many security application of keyword spotting, it becomes critically important to use BSS as pre-processing when we are interested to spot particular keywords and we do not want to lose any piece of information.

An unsupervised keyword spotting framework via segmental DTW on Gaussian posteriorgrams was presented in [1]. However, instead of using independently trained phonetic recognizer or GMM, an ICA mixture was proposed for training the model and generation of posteriorgram in [2]. The training process involves directly modeling the speech without any transcription information using the proposed ICA mixture model. The trained model was further used to decode the keyword examples and test utterances in posteriorgrams. Segmental DTW was used to compare the posteriorgrams between keyword examples and the test utterances. The distortion scores were

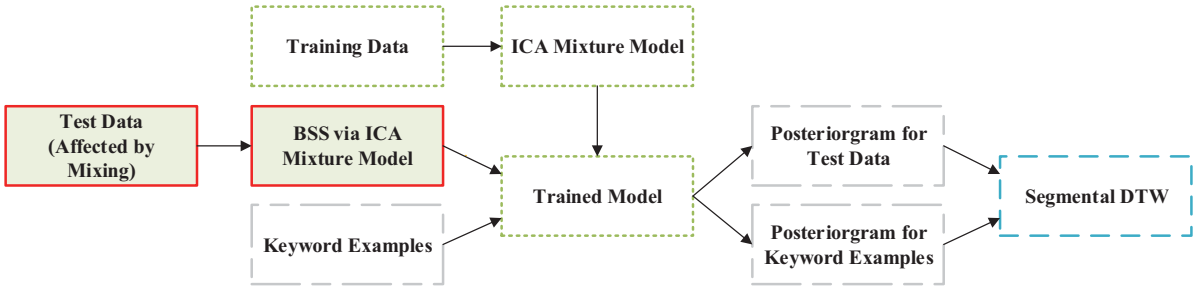


Figure 4.6: Blind Source Separation as Pre-processing to Unsupervised Keyword Spotting via an ICA Mixture Model

ranked for the most reliable warping path to achieve keyword detection [1, 255]. The detailed description on the keyword spotting is provided in [1, 2]. In the experimental setup presented in [2], parameters of the keyword spotting framework were chosen exactly the same as given in [1], in order to have fair comparison of the keyword detection results. ICA mixture model has demonstrated its viability and effectiveness in Keyword spotting framework based on detection rate presented in [2]. Experiments were performed on TIMIT speech corpus and a list of 10 keywords was selected to test the trained model for keyword spotting. In this framework same keyword spotting based on ICA mixture is adopted.

We have extended BSS framework presented in Subsection 4.5 and proposed as pre-processing for keyword spotting when the speech utterances with target keywords are mixed with noise or other speech utterances. The training phase of this proposed framework will remain same as presented in [2]. In order to examine the performance of keyword spotting framework, BSS is applied on test data to recover the speech signals. Once source separation is achieved through BSS via ICA mixture, the recovered signals can be applied to trained model for keyword detection. The proposed framework is shown in Fig. (4.6), which is inspired by [3]. Two types of problems occur in keyword spotting, when source mixing between speech utterances exist at initial stage during the testing. In the first case, if a speech utterance with a particular keyword is mixed with another speech utterance(s) and an overlap of a word exist in the second utterance(s) on the same place as the particular keyword in the first utterance. In this case, the keyword will be mixed with the word of second utterance and it will more likely not detected during the keyword spotting. In the second case, if a silent patch of speech exist in the second utterance at the same place as keyword in the first speech utterance, it will be detected in the first speech utterance during the keyword spotting. But it will also get detected in the second utterance which is a false alarm because keyword actually don't exist in the second speech utterance. These issues are addressed by proposing BSS as pre-processing to keyword spotting.

4.6.1 Experiments and Results

4.6.1.1 Design of Experiment

In this subsection, experimental framework and detection results for BSS based keyword spotting are presented. For keyword spotting, we have adopted the framework proposed in [2], and for the pre-processing stage, blind source separation framework presented in Subsection 4.5 is adopted. In both frameworks, ICA mixture is employed for statistical modeling and estimation of basis functions. In the training phase, speech data dedicated for training are used directly for statistical modeling without any transcription information. Once the model is trained, it can be used further to decode the keyword examples and test utterances into posteriorgrams. In this framework, it is assumed that test data are mixed with noise or other speech signals which requires the application of BSS before generation of posteriorgrams by employing the trained model. In order to perform pre-processing through BSS, we have created mixtures of 2, 3, 4 and 5 speech signals on test data. TIMIT speech corpus is employed during the modeling of keyword spotting framework and validation of the said framework is performed through the selected part of test data after being processed through BSS [143]. The speech signals processed through BSS are further applied to the trained model for generation of posteriorgrams. Segmental DTW is employed to compare the posteriorgrams for test utterances and keyword examples. Mel frequency cepstral coefficients (MFCCs) are used as features for in this framework.

4.6.1.2 Experimental Framework and Results

The BSS based keyword spotting framework is evaluated on TIMIT speech corpus. The TIMIT speech corpus is composed of 6300 speech utterances which contains 4620 speech utterances for training and 1680 speech utterances for testing. In this work, keyword spotting framework is modeled by all of the training data. For testing, speech utterances with target keywords and without target keywords were selected. The speech utterances with target keywords were mixed with the speech utterances without target keywords, for creating a mixture of 2, 3, 4, and 5 speech files. In these mixtures only one speech utterance has the target keyword while the rest of the speech utterances have no target keyword. Voice activity detection and feature extraction are applied directly before the modeling during the training. For testing, feature extraction is applied after the test data are being processed through blind source separation. For feature extraction, each speech utterance is segmented into frames of 25 ms with a window shifting of 10 ms, where each frame is represented by 13 MFCCs. In order to initialize the parameters of ICA mixture during the training, K-Means is applied for mean, standard deviation and mixing weights estimation whereas shape parameter is set to 2 for each component of mixture model. During the training for Keyword

Table 4.9: TIMIT 10 Keyword List used in [1, 2]

age(3:10)	warm(10:8)	year(11:20)	money(19:17)	artists(7:7)
problem(22:9)	children(18:15)	surface(3:7)	development(9:8)	organizations(7:7)

spotting, number of components of ICA mixture is set to be 50 as in [1, 2]. The smoothing factor, segmental DTW adjustment window size and score weighing factor are chosen to be 0.00001, 6 and 0.5, respectively as in [1, 2]. The keyword "Year" is uttered 177 times in the test part of dataset but in these experiments only 20 speech utterances with this keyword were selected, because rest of the keywords are uttered less than 20 times in the test data. For the testing, 10-keyword set presented in [1, 2] is adopted and given in Table 4.9.

For the evaluation of keyword detection, three different evaluation matrices reported in [1, 2, 255] are examined, which are defined as: (1) the average precision for the top 10 utterance hits termed as P@10, (2) the average precision for the top N utterance hits termed as P@N, where N is equal to the number of occurrences of the each keyword in the test data, (3) the average equal error rate (EER), where false acceptance rate is equal to false rejection rate. It is assumed that test data are affected by source mixing and it needs to be processed through BSS before applying to the trained model for generation of posteriorgrams. In order to validated the effectiveness of BSS as pre-processing, a new test data from the selected part of test data from TIMIT speech corpus is created. The purpose of this new dataset is to create the mixtures of 2×2, 3×3, 4×4 and 5×5 with speech utterances having target keywords and having no target keywords. In each mixture, one speech utterance has the target keyword while rest of them do not have target keyword. For example, in the case of keyword "age", all the 10 speech utterances with this keyword are taken and each utterance is mixed with another speech utterance with no target keyword for creating a mixture of 2×2. For mixture of 3×3, each speech utterance of target keyword is mixed with 2 more utterances having no target keyword. For the keyword "age", with mixtures of 2×2, we have 20 speech utterances in total (10 of them have target keyword and 10 have no target keyword), whereas with mixtures of 3×3, we have 30 speech utterances in total (10 of them have target keyword and 20 have no target keyword). All mixtures for the keywords given in Table 4.9 were created in the same fashion as discussed before. During the whole experiment, 100 speech utterances with no target keywords from the Table 4.9 and all the speech utterances with target keywords were selected and used according to the requirement for creating the mixture of speech data for each keyword. The next stage is to apply BSS and then adopt trained ICA mixture to generate posteriorgrams. BSS is performed by ICA mixture and same framework is adopted for

BSS as discussed in Subsection 4.5. Table 4.10 indicates the performance of keyword detection before and after BSS, for different number of keyword examples based on P@N, P@N and EER.

For P@10 evaluation, 4 keywords from Table 4.9 are considered because only they have occurred more than 10 times both in the training and test part of dataset. For P@N and EER evaluations, with one keyword example experiment, all the keywords from the Table 4.9 were used. For P@N and EER evaluations, with 5 keyword examples experiment, 8 keywords were used because they have occurred more than 5 times in the training set. For P@N and EER evaluations, with 10 keyword examples experiment, only 5 keywords were used because they have occurred more than 10 times in the training set. The average precision for each keyword is calculated first and then mean of average precisions of all keywords for P@10 or P@N were computed. The EER for each keyword was computed based on false acceptance rate (FAR) and false rejection rate (FRR). The EER mentioned in Table 4.10 is the average of EER for all keywords used for that particular case [1, 2]. Table 4.10 indicates considerable improvement in the evaluation matrices after being processed through BSS for 2×2 and 3×3 mixtures in comparison to the case when no BSS was applied. There is also improvement for 4×4 and 5×5 mixtures as compared to the case when no BSS was applied, but the trend of improvement is slow as compared to the 2×2 and 3×3 mixtures.

Table 4.10: Evaluation matrix with BSS and without BSS

Mixture	Without BSS				After BSS		
	Examples	P@10	P@N	EER	P@10	P@N	EER
2×2	1	11.43%	9.58%	81.19%	23.15%	22.45%	37.43%
	5	13.76%	12.25%	77.55%	46.86%	43.89%	25.38%
	10	15.27%	13.81%	76.19%	53.44%	52.11%	24.81%
3×3	1	8.97%	7.13%	85.47%	22.67%	22.15%	39.19%
	5	10.14%	9.58%	81.44%	44.63%	41.74%	26.88%
	10	10.76%	9.85%	80.11%	51.46%	49.15%	25.43%
4×4	1	7.54%	6.45%	89.13%	20.13%	21.37%	40.87%
	5	8.10%	6.93%	87.46%	40.92%	38.49%	29.15%
	10	8.37%	7.21%	86.79%	46.12%	44.32%	28.87%
5×5	1	6.13%	5.89%	91.37%	18.67%	18.15%	42.37%
	5	6.48%	6.27%	88.49%	38.19%	36.56%	31.13%
	10	7.22%	6.91%	88.07%	43.68%	42.06%	30.45%

The results for average precisions (P@10 and P@N) are very close to each other, because utterance of available keywords in the test data is very close to 10 in most of the cases. It is also important to note that only 4 keywords are present more than 10 times in both training and testing and hence P@10 was computed only for 4 keywords from the list given in Table 4.9. However for P@N, most of the keywords were used for computations, so it more effective for examining the

viability of this framework. It is also observed that trend of improvement is higher when going from one keyword example to 5 keyword examples, whereas it is slow from 5 to 10 keyword examples. If we compare the results presented in this chapter with the results presented in the frameworks when no source mixing is considered, there is a lot of room for improvement. However, comparison of keyword spotting with BSS and keyword spotting without BSS indicates the effectiveness of this framework in keyword spotting when speech signals are affected by mixing. It is also observed that the problem of false alarm due to the mixing of sources is more severe in computing the detection rate for keyword spotting, which actually reduces the overall performance of keyword spotting. In many security applications, it is necessarily important to find the particular keywords because they are further used to detect the particular speakers. If false alarm occur and even correct speaker is also detected, it will increase the number of possibilities to find the particular speaker. In the other case, when keyword is mixed with the words of other speech utterances and it is more likely not detected during the keyword spotting, it can increase the chances of completely losing a particular information. It is important when it is mixed intentionally to hide the particular information (keyword) which is critical to security. The experiments performed in this work only include the mixing of speech utterances. This framework needs to be extended for keyword spotting when speech utterances are mixed with noise. This experiment can be further extended with a larger vocabulary database by having more number of keyword examples.

4.6.2 Discussion About Unsupervised Keyword Spotting with BSS as Pre-processing

We proposed BSS as pre-processing to unsupervised keyword spotting by employing an ICA mixture model when speech utterances having target keywords are affected by mixing of noise or other keywords.We have used the same ICA mixture model for statistical modeling in the keyword spotting framework as recently proposed in our work. The experiments are performed by employing TIMIT speech corpus to train the ICA mixture for keyword spotting and then selecting the part of test data for creating a mixture of 2, 3, 4 and 5 speech signals to perform the blind source separation before the keyword spotting. The purpose of creating these mixtures of speech utterances with target keyword and with no target keyword is to validate the effectiveness of proposed framework. The keyword detection results are presented before and after the test data being processed through blind source separation. The keyword detection results based on average precision (P@10 & P@N), and EER validate the effectiveness of proposed framework when speech utterances with target keywords are affected by mixing. Our experiments have shown significant improvement in detection of keywords when mixed speech signals are processed through BSS via an ICA mixture.

Chapter 5

Multivariate Bounded Support Asymmetric Mixture Models and MML

In this chapter, we have proposed two algorithms using asymmetric distributions. First, bounded asymmetric Gaussian mixture model (BAGMM) is proposed. In the described model, parameter estimation is performed by maximization of log-likelihood via expectation maximization (EM) and Newton Raphson algorithm. This model is applied to several applications for data clustering. As a first step, to validate our model, we have chosen spambase dataset for clustering spam and non-spam emails. Another application selected for validation of our algorithm is object data clustering and we have used two popular datasets (Caltech 101 & Corel) in this task. Finally we have performed clustering on texture data and VisTex dataset is employed for this task. In order to evaluate the clustering, in all above mentioned applications, several performance metrics are employed and experimental results are further compared in similar settings with asymmetric Gaussian mixture model (AGMM). From the experiments and results in all applications, it is examined that BAGMM has outperformed AGMM in the clustering task.

Second, bounded support asymmetric generalized Gaussian mixture model (BAGGMM) is proposed for data modeling as an alternative to unbounded mixture models for the cases when the data lies in bounded support region. The parameters of the model are learned through maximum likelihood estimation and Expectation Maximization (EM) with Newtons Raphson method is adopted for optimization of parameters. Model selection in mixtures is also considered to be an integral part of clustering, thus we also have proposed model selection criterion for BAGGMM through minimum message length. In order to validate the performance of mixture model, it is applied to image spam detection, object clustering and visual scene categorization. For the experiments, Spam Hunter, ETHZ, GHIM and 15-Scene image datasets are adopted and several clustering scenarios are developed to see the effectiveness of proposed model. The clustering framework

is also compared with AGGMM in a similar setting with all the experiments. In the next phase, whole clustering framework is extended to examine the performance of proposed model selection criterion and compared with different techniques to find the optimal mixture component, which further improve the clustering process. From the experiments, it is observed that proposed BAGGMM and model selection criterion have demonstrated its success in several learning applications.

5.1 Multivariate Bounded Asymmetric Gaussian Mixture Model

In the case of GMMs, the components distribution is symmetric in nature. However, generally while using real data this is not the case. The data might not be symmetrical, which means GMM could not provide a good fit to the data. So, using an asymmetric distribution will be a better choice for our model. Hence in our model, we use an asymmetric Gaussian distribution which will provide a better fit to the data [32, 55–57]. Asymmetric Gaussian distribution has two standard deviation parameters on the left and right side of distribution, which make it possible to model asymmetric data [32]. Motivated by observations in [62], we propose the idea of bounded asymmetric Gaussian mixture model (BAGMM) for data modeling which also has the ability to model asymmetric nature of data. In the proposed model, parameter estimation is performed by maximum likelihood with Newton Raphson via expectation maximization algorithm (EM). In order to evaluate the effectiveness of our model, BAGMM is applied to several data clustering applications. As a first step, it is applied to categorize spam and non-spam emails and spambase dataset is employed for this task. The performance of clustering task is examined by 9 different metrics which provide insightful knowledge about the effectiveness of BAGMM in clustering the spambase dataset. The results of this task are further compared with AGMM in a similar framework. In second application, BAGMM is applied to object categorization and two popular image datasets renowned for object categorization (Caltech 101 & Corel) are employed for this task. The clustering performance is observed by difference metrics and with a comparison with AGMM in a similar framework. In the third application for data clustering, BAGMM is applied to texture image dataset (VisTex) and performance of our proposed algorithm is examined via performance measures and a comparison with AGMM. In Fig. (5.1), graphical abstract is presented which also provide more clear understanding of the contributions of this research work.

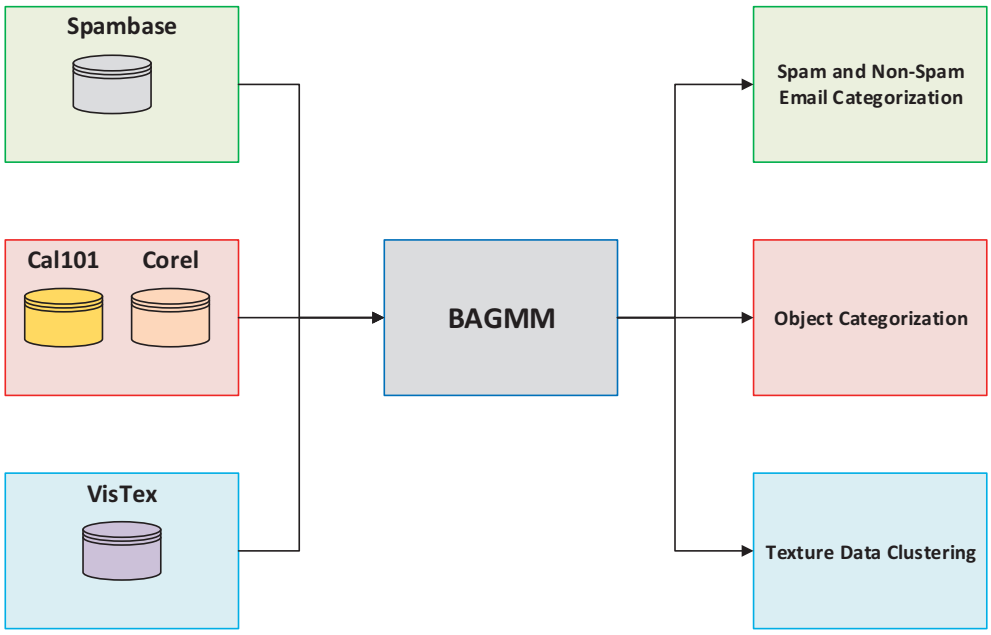


Figure 5.1: Graphical abstract

5.2 Proposed Model

We propose BAGMM as an extension to AGMM for an improved data modeling. In this section, proposed bounded asymmetric Gaussian mixture model is presented which uses maximum log-likelihood for the estimation of its parameters. Before presenting the proposed model, asymmetric Gaussian mixture model.

5.2.1 Mixture of Asymmetric Gaussian Distributions

Asymmetric Gaussian mixture model was proposed to handle the asymmetric properties present in different kind of data [32, 56, 260]. For a univariate data, if one data sample is represented by X, then asymmetric Gaussian distribution is represented as follows:

$$f(X|\mu,\sigma_l,\sigma_r) = \frac{2}{\sqrt{2\pi}(\sigma_l+\sigma_r)} \times \begin{cases} \exp\left[-\frac{(\mathrm{X}-\mu)^2}{2\sigma_l^2}\right] & \text{if} \quad \mathrm{X} < \mu \\ \exp\left[-\frac{(\mathrm{X}-\mu)^2}{2\sigma_r^2}\right] & \text{if} \quad \mathrm{X} \geq \mu \end{cases} \tag{5.1}$$

where parameters of distribution μ, σ_l & σ_r are mean, left standard deviation and right standard deviation, respectively. The parameters of AGMM are estimated using ML estimate and complete parameter estimation is explained in [32, 56, 260]. In Fig. (5.2), graphical representation of AGMM is displayed, where X_i is a data point with $i = 1, ..., N$, μ, σ_l & σ_r, parameters of distribution and p and Z_i are mixing weight and posterior probability in a mixture model and they are explained in detail in Section 5.2.2.

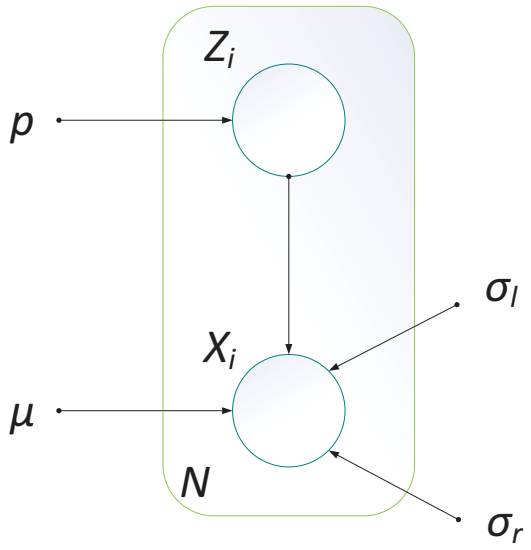


Figure 5.2: Graphical representation of an asymmetric Gaussian mixture model

5.2.2 Mixture of Bounded Asymmetric Gaussian Distribution for Multidimensional Data

Consider that a D-dimensional random variable $\vec{X} = (X_1, ..., X_D)$, follows a K components mixture distribution if its probability function can be written in the following form:

$$p(\vec{X}|\Theta) = \sum_{j=1}^{K} p(\vec{X}|\xi_j) p_j \tag{5.2}$$

provided $p_j \geq 0$, $\sum_{j=1}^{K} p_j = 1$, $\Theta = (\xi_1, \xi_2, \xi_3, \xi_4)$ with $\xi_1 = (\vec{\mu}_1, ..., \vec{\mu}_K)$, $\xi_2 = (\vec{\sigma}_{l_1}, ..., \vec{\sigma}_{l_K})$, $\xi_3 = (\vec{\sigma}_{r_1}, ..., \vec{\sigma}_{r_K})$ and $\xi_4 = (p_1, ..., p_K)$. The term $p(\vec{X}|\xi_j)$ is BAGD for the vector $\vec{X}$ and defined as:

$$p(\vec{X}|\xi_j) = \frac{f(\vec{X}|\xi_j)\mathrm{H}(\vec{X}|\Omega_j)}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \tag{5.3}$$

$$\text{where } \mathrm{H}(\vec{X}|\Omega_j) = \begin{cases} 1 & \text{if } \vec{X} \in \partial_j \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

$$f(\vec{X}|\xi_j) = \prod_{d=1}^{D} \frac{2}{\sqrt{2\pi}(\sigma_{l_{jd}} + \sigma_{rjd})} \times \begin{cases} \exp\left[-\frac{(\mathrm{X}_d - \mu_{jd})^2}{2\sigma_{l_{jd}}^2}\right] & \text{if } \mathrm{X}_d < \mu_{jd} \\ \exp\left[-\frac{(\mathrm{X}_d - \mu_{jd})^2}{2\sigma_{r_{jd}}^2}\right] & \text{if } \mathrm{X}_d \geq \mu_{jd} \end{cases} \tag{5.5}$$

where $\vec{\mu}_j = (\mu_{j1}, ..., \mu_{jD})$, $\vec{\sigma}_{l_j} = (\sigma_{l_{j1}}, ..., \sigma_{l_{jD}})$, and $\vec{\sigma}_{r_j} = (\sigma_{r_{j1}}, ..., \sigma_{r_{jD}})$ are the mean, left standard deviation and right standard deviation of the D-dimensional BAGD, respectively. The term $\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}$ in Eq. (5.3) is the normalization constant that indicates the share of $f(\vec{X}|\xi_j)$ which belongs to the support region ∂. The AGD $f(\vec{X}|\xi_j)$ can also be defined as:

$$f(\vec{X}|\xi_j) = \begin{cases} \mathrm{g}_1(\vec{X}|\xi_j) & \text{if } \mathrm{X}_d < \mu_{jd} \\ \mathrm{g}_2(\vec{X}|\xi_j) & \text{if } \mathrm{X}_d \geq \mu_{jd} \end{cases} \tag{5.6}$$

where

$$\mathrm{g}_1(\vec{X}|\xi_j) = \prod_{d=1}^{D} \frac{2}{\sqrt{2\pi}(\sigma_{l_{jd}} + \sigma_{rjd})} \exp\left[-\frac{(\mathrm{X}_d - \mu_{jd})^2}{2\sigma_{l_{jd}}^2}\right] \tag{5.7}$$

$$\mathrm{g}_2(\vec{X}|\xi_j) = \prod_{d=1}^{D} \frac{2}{\sqrt{2\pi}(\sigma_{l_{jd}} + \sigma_{rjd})} \exp\left[-\frac{(\mathrm{X}_d - \mu_{jd})^2}{2\sigma_{r_{jd}}^2}\right] \tag{5.8}$$

Consider the case where the input is set of vectors represented as $\mathscr{X} = (\vec{X}_1, ..., \vec{X}_N)$. With a mixture of K BAGDs, the distribution of $\mathscr{X}$ can be modeled by a mixture of K BAGDs:

$$p(\mathscr{X}|\Theta) = \prod_{i=1}^{N} \sum_{j=1}^{K} p(\vec{X}_i|\xi_j)p_j \tag{5.9}$$

provided $p_j \geq 0$ and $\sum_{j=1}^{K} p_j = 1$. In Eq. (5.9), Θ represents the parameters of mixture model having K classes as $\Theta = (\xi_1, \xi_2, \xi_3, \xi_4)$, where $\xi_1 = (\vec{\mu}_1, ..., \vec{\mu}_K)$, $\xi_2 = (\vec{\sigma}_{l_1}, ..., \vec{\sigma}_{l_K})$, $\xi_3 = (\vec{\sigma}_{r_1}, ..., \vec{\sigma}_{r_K})$ and $\xi_4 = (p_1, ..., p_K)$. Stochastic indicator vectors $\vec{Z}_i = (Z_{i1}, ..., Z_{iK})$, one for each observation are introduced. The role is to encode the membership of each observation for a relative component of the mixture model. In other words, Z_{ij}, the unobserved variable in each indicator vector, equals 1

if $\vec{X}_i$ belongs to class j and 0, otherwise. The complete data likelihood is given below.

$$p(\mathscr{X},\mathscr{Z}|\Theta)=\prod_{i=1}^{N}\prod_{j=1}^{K}\Big(p(\vec{X}_i|\xi_j)p_j\Big)^{Z_{ij}} \tag{5.10}$$

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

$$Z_{ij}=p(j|\vec{X}_i)=\frac{p(\vec{X}_i|\xi_j)p_j}{\sum_{j=1}^{K}p(\vec{X}_i|\xi_j)p_j} \tag{5.11}$$

and $\mathscr{Z}=\{\vec{Z}_1,...,\vec{Z}_N\}$.

5.2.3 Parameters Learning

The parameters are estimated from the maximization of positive log-likelihood function. The log-likelihood function can be written as:

$$\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)=\sum_{i=1}^{N}\sum_{j=1}^{K}Z_{ij}\log\Big(p(\vec{X}_i|\xi_j)p_j\Big) \tag{5.12}$$

$$\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)=\sum_{i=1}^{N}\sum_{j=1}^{K}Z_{ij}\left\{\log p_j+\log f(\vec{X}_i|\xi_j)+\log \mathrm{H}(\vec{X}_i|\Omega_j)-\log\int_{\partial_j}f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right\} \tag{5.13}$$

The complete-data log-likelihood can be maximized with respect to the model parameters. This can be done by taking the gradient of the log-likelihood with respect to p_j, μ_j, σ_{l_j} and σ_{r_j}. The estimation of mixing parameter is followed from Section 2.2.3.1. The estimation for μ_j, σ_{l_j} and σ_{r_j} in a bounded support asymmetric Gaussian mixture model is explained below.

5.2.3.1 Mean Parameter Estimation

The new value of Mean μ_{jd}, can be estimated by maximizing the log-likelihood function given in Eq. (5.13) with respect to $\vec{\mu}_j$. The derivative of log-likelihood and estimation of maximum likelihood are given in Appendix D.1 & D.2. The estimated value of mean is given as follows:

$$\frac{\partial\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial\mu_{jd}}=0 \tag{5.14}$$

$$\hat{\mu}_{jd} = \frac{\sum_{i=1}^{N} Z_{ij} \left\{ \mathrm{X}_{id} - \frac{\int_{\partial_j} f(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})d\mathrm{x}}{\int_{\partial_j} f(\mathrm{u}|\xi_j)d\mathrm{u}} \right\}}{\sum_{i=1}^{N} Z_{ij}} \tag{5.15}$$

Note that, in Eq. (5.15), the term $\int_{\partial_j} f(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})dx$ is the expectation of function $(\mathrm{u}-\mu_{jd})$ under the probability distribution $f(X_d|\xi_j)$. Then, this expectation can be approximated as:

$$\int_{\partial_j} f(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})dx \approx \frac{1}{M}\sum_{m=1}^{M}(s_{m_{jd}}-\mu_{jd})\mathrm{H}(s_{m_{jd}}|\Omega_j) \tag{5.16}$$

where $s_{m_{jd}} \sim f(\mathrm{u}|\xi_j)$ is a set of random variables drawn from the asymmetric Gaussian distribution for the particular component j of the mixture model. The set of data with random variables have M vectors with D dimensions. M is a large integer chosen to generate the set of random variables. Similarly, the term $\int_{\partial_j} f(\mathrm{u}|\xi_j)dx$ in Eq. (5.15) can be approximated as:

$$\int_{\partial_j} f(\mathrm{u}|\xi_j)dx \approx \frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(s_{m_{jd}}|\Omega_j) \tag{5.17}$$

$$\hat{\mu}_{jd} = \frac{\sum_{i=1}^{N} Z_{ij} \left\{ \mathrm{X}_{id} - \frac{\sum_{m=1}^{M}(s_{m_{jd}}-\mu_{jd})\mathrm{H}(s_{m_{jd}}|\Omega_j)}{\sum_{m=1}^{M}\mathrm{H}(s_{m_{jd}}|\Omega_j)} \right\}}{\sum_{i=1}^{N} Z_{ij}} \tag{5.18}$$

5.2.3.2 Left Standard Deviation Estimation

The new value of left standard deviation $\sigma_{l_{jd}}$, can be estimated by maximizing the log-likelihood function given in Eq. (5.13) with respect to $\vec{\sigma}_{l_j}$.

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma_{l_{jd}}} = 0 \tag{5.19}$$

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma_{l_{jd}}} = \sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma_{l_{jd}}^3}\right) - \sum_{i=1,\mathrm{u}<\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma_{l_{jd}}^3}\left\{\frac{\int_{\partial_j}\frac{2}{\sqrt{2\pi}(\sigma_{l_{jd}}+\sigma_{r_{jd}})}\left(\exp\left[-\frac{(\mathrm{u}-\mu_{jd})^2}{2\sigma_{l_{jd}}^2}\right]\right)(\mathrm{u}-\mu_{jd})^2 dx}{\int_{\partial_j}\mathrm{g}_1(\mathrm{u}|\xi_j)dx}\right\} \tag{5.20}$$

$$\sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma_{l_{jd}}^3}\right)-\sum_{i=1,\mathrm{u}<\mu_{jd}}^{N}\frac{Z_{ij}}{\sigma_{l_{jd}}^3}\left\{\frac{\int_{\partial_j}\mathrm{g}_1(\mathrm{u}|\xi_j)dx(\mathrm{u}-\mu_{jd})^2dx}{\int_{\partial_j}\mathrm{g}_1(\mathrm{u}|\xi_j)dx}\right\}=0 \tag{5.21}$$

The term $\int_{\partial_j}\mathrm{g}_1(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^2dx$ can be approximated as below:

$$\int_{\partial_j}\mathrm{g}_1(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^2dx\approx\frac{1}{M}\sum_{m=1}^{M}(\mathrm{l}_{m_{jd}}-\mu_{jd})^2\mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j) \tag{5.22}$$

where $\mathrm{l}_{m_{jd}}\sim\mathrm{g}_1(X_d|\xi_j)$ is a set of random variables drawn from the asymmetric Gaussian distribution with $\mathrm{u}<\mu_{jd}$ for the particular component j of the mixture model. Similarly, the term $\int_{\partial_j}f(\mathrm{u}|\xi_j)dx$ in Eq. (5.15) can be approximated as:

$$\int_{\partial_j}\mathrm{g}_1(\mathrm{u}|\xi_j)dx\approx\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j) \tag{5.23}$$

$$\sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma_{l_{jd}}^3}\right)-\sum_{i=1}^{N}\frac{Z_{ij}}{\sigma_{l_{jd}}^3}\left\{\frac{\frac{1}{M}\sum_{m=1}^{M}(\mathrm{l}_{m_{jd}}-\mu_{jd})^2\mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j)}{\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j)}\right\}=0 \tag{5.24}$$

It is noticed that Eq. (5.24) is non-linear and Newton-Raphson method is used for the estimation of $\hat{\sigma}_{l_{jd}}$, which requires the computation of second derivative in a similar manner as we have computed in Eqs. (5.20 & 5.24). The complete procedure for first and second order derivatives is provided in Appendix D.3 & D.4, respectively along with approximation procedure for second order derivative, similar to Eq. (5.24).

$$\hat{\sigma}_{l_{jd}}\simeq\sigma_{l_{jd}}-\left[\left(\frac{\partial^2\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial{\sigma^2}_{l_{jd}}}\right)^{-1}\left(\frac{\partial\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial\sigma_{l_{jd}}}\right)\right] \tag{5.25}$$

5.2.3.3 Right Standard Deviation Estimation

Right standard deviation $\sigma_{r_{jd}}$, can be estimated by maximizing the log-likelihood function given in Eq. (5.13) with respect to $\vec{\sigma}_{r_j}$.

$$\frac{\partial\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial\sigma_{l_{jd}}}=0 \tag{5.26}$$

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma_{r_{jd}}} = \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma^3_{r_{jd}}}\right) - \sum_{i=1,\mathrm{u}\geq\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^3_{r_{jd}}}\left\{\frac{\int_{\partial_j}\frac{2}{\sqrt{2\pi}(\sigma_{l_{jd}}+\sigma_{r_{jd}})}\left(\exp\left[-\frac{(\mathrm{u}-\mu_{jd})^2}{2\sigma^2_{r_{jd}}}\right]\right)(\mathrm{u}-\mu_{jd})^2 dx}{\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)dx}\right\} \tag{5.27}$$

$$\sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma^3_{r_{jd}}}\right) - \sum_{i=1,\mathrm{u}\geq\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^3_{r_{jd}}}\left\{\frac{\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)dx(\mathrm{u}-\mu_{jd})^2 dx}{\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)dx}\right\} = 0 \tag{5.28}$$

The term $\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^2 dx$ can be approximated as below:

$$\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^2 dx \approx \frac{1}{M}\sum_{m=1}^{M}(\mathrm{r}_{m_{jd}}-\mu_{jd})^2\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j) \tag{5.29}$$

where $\mathrm{r}_{m_{jd}} \sim \mathrm{g}_2(X_d|\xi_j)$ is a set of random variables drawn from the asymmetric Gaussian distribution with $\mathrm{u}\geq\mu_{jd}$ for the particular component j of the mixture model. Similarly, the term $\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)dx$ in Eq. (5.15) can be approximated as:

$$\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)dx \approx \frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j) \tag{5.30}$$

$$\sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma^3_{r_{jd}}}\right) - \sum_{i=1}^{N}\frac{Z_{ij}}{\sigma^3_{r_{jd}}}\left\{\frac{\frac{1}{M}\sum_{m=1}^{M}(\mathrm{r}_{m_{jd}}-\mu_{jd})^2\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j)}{\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j)}\right\} = 0 \tag{5.31}$$

It is noticed that Eq. (5.31) is non-linear, therefore Newton-Raphson method is used for the estimation of $\hat{\sigma}_{r_{jd}}$, which requires the computation of second derivative in a similar manner as computed in Eqs. (5.27 & 5.31). The complete procedure for first and second order derivatives is provided in Appendix D.5 & D.6, respectively along with approximation procedure for second order derivative, similar to Eq. (5.31).

$$\hat{\sigma}_{r_{jd}} \simeq \sigma_{r_{jd}} - \left[\left(\frac{\partial^2\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial{\sigma^2}_{r_{jd}}}\right)^{-1}\left(\frac{\partial\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial\sigma_{r_{jd}}}\right)\right] \tag{5.32}$$

The complete learning of BAGMM is given in Algorithm 5, where t_{min} is minimum threshold used to monitor the convergence criteria in each iteration. In the initialization phase, K-Means is applied

Algorithm 5 Model Learning for BAGMM

Input:Dataset $\mathscr{X} = \{\vec{X}_1, \ldots, \vec{X}_N\}$, t_{min}.
Output: Θ, $\mathscr{Z}$.
{Initialization}:
  K-Means Algorithm (Computation of $\vec{\mu}_1, \ldots, \vec{\mu}_K$ & cluster assignment)
    for all $1 \leq j \leq K$ do
      Computation of p_j
      Computation of $\{(\vec{\sigma}_{l_j}$ & $\vec{\sigma}_{r_j})$=$\vec{\sigma}_j\}$
    end for
{Expectation Maximization}:
while relative change in log-likelihood $\geq t_{min}$ do
  $\{$[**E Step**]$\}$:
    for all $1 \leq j \leq K$ do
      Compute $p(j|\vec{X}_i)$ for $i = 1, \ldots, N$. using Eq. (5.11).
    end for
  $\{$[**M step**]$\}$:
    for all $1 \leq j \leq K$ do
      Estimation of mixing parameter p_j using Eq. (2.12).
      Estimation of mean $\vec{\mu}_j$ using Eq. (5.18).
      Estimation of left standard deviation $\vec{\sigma}_{l_j}$ using Eq. (5.25).
      Estimation of right standard deviation $\vec{\sigma}_{r_j}$ using Eq. (5.32).
    end for
end while

for computation of mean and data assignment in each cluster. This information is further used for computation of standard deviation and mixing weight during initialization phase.

5.3 Textual Spam Detection

Email has become the prominent choice of communication, particularly for professional purposes [261]. Among the legitimate emails conveying meaningful and important information, there is an immense amount of spam ones which not only contain disturbing commercial contents but also deliver scamming schemes such as phishing [262]. Indeed, the ubiquitous usage of emails has made it the fitting platform for cyberattacks, which bring about annoyance and unnecessary time or possibly money loss. Furthermore, unsolicited spams have also been the leading cause for the productivity and financial cost of various companies due to hiring cybersecurity specialists and expanding email servers [263]. Therefore, it is crucial that spam instances be efficiently and accurately detected and removed to avoid wasting additional efforts. Recent works applying

Table 5.1: Performance of spambase data clustering based on different metrics

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BAGMM	85.69	84.23	87.15	86.76	12.85	85.47	71.40	85.48	85.67
AGMM	77.05	73.75	80.36	78.97	19.64	76.27	54.23	76.31	76.98

Gaussian mixture models on spam detection have shown their efficiency and modeling capabilities [264, 265]. Thus, we propose continuation of this research via asymmetric Gaussian mixture model. We have applied our proposed BAGMM for clustering the spam and non-spam emails and it is further extended with AGMM to have a comparison in order to evaluate the effectiveness of BAGMM in clustering.

The spambase dataset [266] is chosen for our experiment, in which each feature vector represents the occurrences 'histograms of words' in emails. There are 3626 emails evenly divided as spams and non-spams. The confusions matrix given in Fig. (5.3) and results in Table (5.1) show that proposed algorithm outperforms the AGMM in clustering the spam and non-spam emails. The evaluation of this data clustering framework is done by choosing all above performance metrics and results of all metrics are better for BAGMM as compared to AGMM. For spam detection, low value of FPR is very important and in the results for BAGMM, FPR is improved as compared to AGMM.

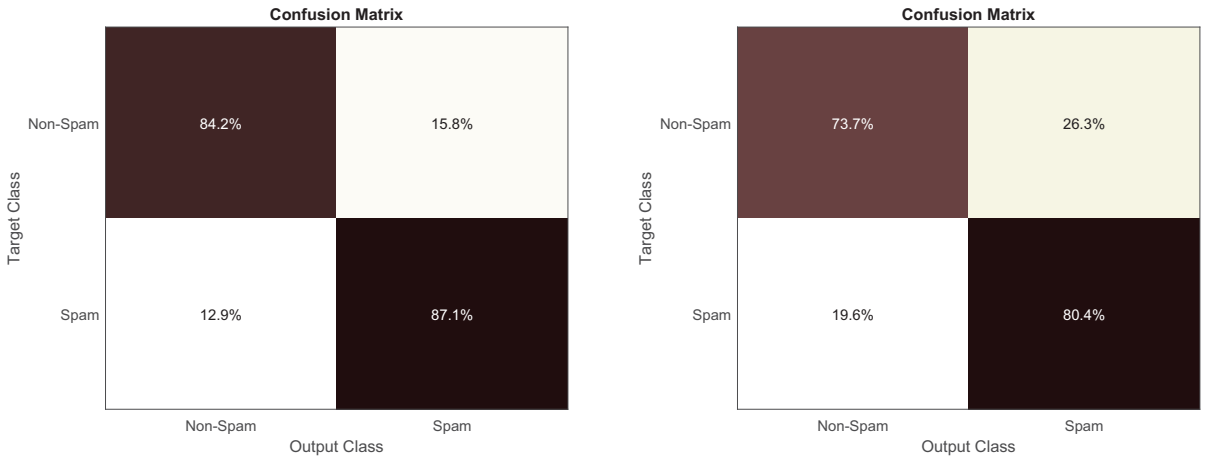


Figure 5.3: Confusion matrix of spambase dataset with BAGMM and AGMM, respectively

Figure 5.4: Sample images of each class of Caltech 101 dataset

Table 5.2: Performance of object data clustering (Caltech 101) based on different metrics

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BAGMM	81.41	77.04	95.67	72.08	4.33	72.67	69.43	74.52	85.85
AGMM	73.35	64.11	93.91	61.65	6.09	60.49	56.98	62.87	77.59

5.4 Object Categorization via Bounded Asymmetric Gaussian Mixture Model

Object clustering, one of the most fundamental topic in computer vision, has received increasing attention as the rapid development of machine learning techniques and latest machines having good computational capabilities [267]. The challenging aspects of the aforementioned task are due to the status variation of the objects in natural environments such as different postures, angles, distances, etc. Furthermore, objects captured in real world conditions usually contain other items in the background which may cause the mis-classification with the noises. Recent clustering analyses using mixture models have shown good results on numerous categorization problems namely scenes [268], sport activities [269], medial related images [270], and 3D objects [232]. Thus, the prospective progress has motivated the authors to apply the proposed model on this challenging task with two widely used datasets: Caltech 101 [271] and Corel [272, 273].

An accurate representation of the images is essential for performing efficient inference process. Excellent outcomes have been achieved by utilizing frameworks based on Bag of Visual Words (BOVW). The main idea is extracting local features for each image using SIFT(Scale Invariant Feature transform) [274]. Then, the collection of all the 128-D descriptors are clustered with K-means in order to build the visual words vocabulary, in which the dimension of the feature vectors is the number of centroids.

5.4.1 Experiments and Results

5.4.1.1 Experimental Framework and Results: Caltech 101 Dataset

In this subsection, we used the Caltech 101 dataset for object clustering. This dataset is popular [271, 275–277] which has demonstrated its effectiveness for object categorization using different

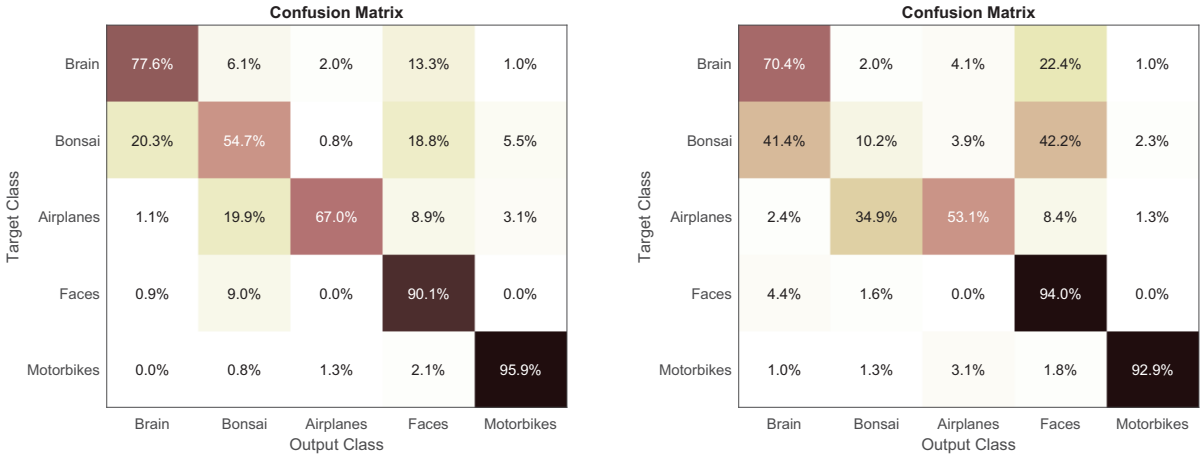


Figure 5.5: Confusion matrix of Caltech 101 dataset with BAGMM and AGMM, respectively

algorithms [278, 279], techniques [280] and feature extraction methods [281–284] and hence, it is well suited for object clustering in our current research. It contains 101 categories of different objects. It consists of 3D pose variations along with multiple objects in a single image. The images inside this dataset are of moderately of good quality, the categories are well annotated, selected and has pose variation controlled. For the experimentation, we have used 5 classes namely "Brain", "Bonsai", "Airplane", "Faces" and "Motorbikes" where these classes contains 98, 128, 800, 435 and 798 images, respectively. Some examples of images from these classes are given in Fig. (5.4). After several experiments, we examined that optimal vocabulary size is 50 and hence, BOVW gives a matrix having a size of 2259×50, where columns represent the frequency of visual words and row is equal to the number of images. Afterward, this matrix is given as an input to the proposed mixture model. In order to ensure the performance of our proposed algorithm, we have used several performance metrics as described in Section 5.3. For comparison, we have implemented the same framework with AGMM. In this data clustering task, the distribution of classes is not balanced which make it difficult to differentiate between different classes and it is depicted from the confusion matrix provided for AGMM as shown in Fig (5.5). By applying BAGMM, same clustering task is improved a lot and it is worth to note that our proposed algorithm outperformed the AGMM as presented in Table (5.2).

5.4.1.2 Experimental Framework and Results: Corel Dataset

In this subsection, we discuss the experiment design. We employed the Corel dataset, which consists of 10,000 images from 100 categories. We have used SIFT and BOVW methods in order to achieve a good representation of the images in feature space. In order to conduct the experiments,

Figure 5.6: Sample images of each class of Corel dataset

Table 5.3: Performance of object data clustering (Corel dataset) based on different metrics

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BAGMM	93.40	93.40	98.35	94.32	1.65	93.45	92.12	93.86	95.84
AGMM	83.80	83.80	95.95	85.37	4.05	84.14	80.41	84.58	89.67

we have used 5 classes where each class contains 100 images. The classes chosen in this experiment are "Playing Cards", "Paintings", "Easter Eggs", "Beads" and "Cups". Some examples of images from these classes are given in Fig. (5.6). After feature extraction, BOVW is a matrix of dimension 500×50, where columns represents the frequency of visual words and row is equal to the number of images. The introduced model is applied to perform the clustering task. In order to validate the performance of our model, we have used several metrics as described in Section 5.3. In order to have a comparison of our model with AGMM, we also have performed clustering using AGMM. Based on the results given in Table (5.3) and confusion matrix in Fig. (5.7), it is observed that our proposed algorithm performed better than AGMM. By applying BAGMM, we have received very high clustering accuracy in this object categorization task and FPR is reduced from 4.05% to 1.65%.

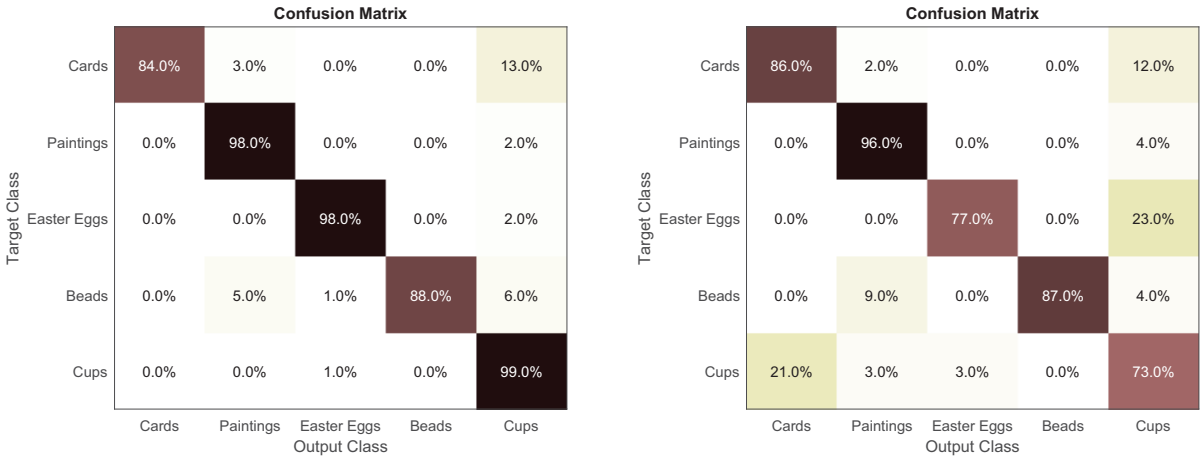


Figure 5.7: Confusion matrix of Corel dataset with BAGMM and AGMM, respectively

5.5 Texture Image Clustering

Texture is a fundamental element of human visual impression towards the world [285]. Indeed, understanding different textures is very beneficial for further complicated object classification, segmentation analyses, which includes various objects and surface types [286]. In order to counter issues namely noise, complexity, slow convergence, and over-fitting, feature extraction is required. Various types of feature extraction methods exist [287]. But, the co-occurrence matrix is a popular feature extraction technique when it comes to texture data [288–290]. Thus, co-occurrence matrix is used to extract the texture characteristics [291]. The co-occurrences are calculated with respect to their neighbors: (1;0), (1;$\frac{\pi}{4}$), (1;$\frac{\pi}{2}$), and (3;$\frac{\pi}{4}$). Then, the co-occurrence matrix of each neighborhood is constructed by considering four features: Homogeneity, Contrast, Correlation, and Energy. Thus, each image is represented as a 16-D feature vector.

Figure 5.8: Sample images of each class of VisTex dataset

5.5.1 Experiments and Results

5.5.1.1 Experimental Framework and Results for VisTex Texture Dataset

This section is dedicated for experiments and results on texture data clustering. We employed the MIT Vision Texture (VisTex) dataset [292]. It is a collection of texture images that are representative of real-world conditions. We treated the original images as parent images and further created offspring images from it. In our experiment, we are using co-occurrence matrix for feature extraction. For the experimentation, we divided each 512×512 parent image into 64×64 off-springs images, where each parent image is converted to 64 off-springs images from VisTex dataset. By using co-occurrence matrix for feature extraction, we converted each offspring image into feature vector of 1×16. We have used images from 4 different categories namely “fabric”, “food”, “paintings” and “tiles”, where these classes contains 192, 320, 448 and 128 sub-images. Some examples of images from VisTex dataset is given in Fig. (5.8). The data matrix after feature extraction is provided to BAGMM for data clustering. In order to validate our proposed algorithm,

Table 5.4: Performance of texture data clustering based on different metrics

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BAGMM	81.34	84.93	92.80	85.42	07.20	85.17	77.97	85.17	88.78
AGMM	73.90	79.62	89.90	79.59	10.10	79.60	69.51	79.60	84.60

we have used several performance metrics as described in Section 5.3. In order to have a comparison, we have implemented the same clustering framework with AGMM. From the results provided in Table (5.4), it is observed that our proposed algorithm outperformed the AGMM. It is necessary to mention that the classes in this application are not balanced which make the clustering task very difficult and it is obvious from the confusion matrix for AGMM in Fig. (5.9). By applying, BAGMM, the clustering accuracy is improved tremendously and FPR is reduced from 10.10% to 7.20%.

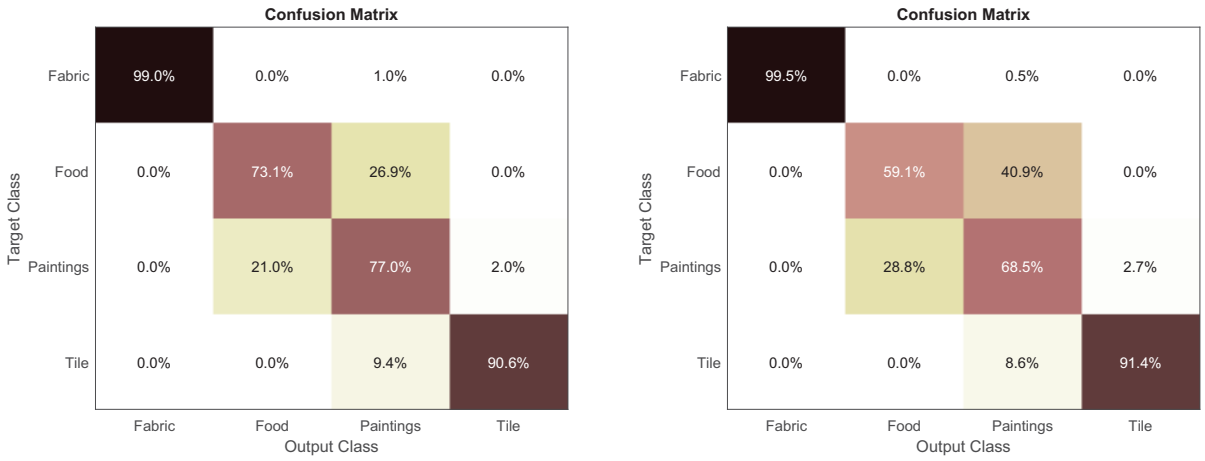


Figure 5.9: Confusion matrix of Vistex dataset with BAGMM and AGMM, respectively

5.6 Discussion about BAGMM

We have proposed BAGMM which uses maximum likelihood for parameter estimation and Newton Raphson via expectation maximization approach. The basic reason to propose bounded support mixture models is that most of the data lies in a bounded range. Due to the bounded nature of most of the data in different real applications, it makes more sense to propose bounded distributions for modeling the data. To validate the effectiveness of proposed algorithm in data modeling, we have chosen spam and non-spam email clustering, object categorization and texture image clustering applications. For spam and non-spam email clustering, spambase dataset is employed. For object categorization, Caltech 101 and Corel datasets are chosen with 5 classes from each dataset. For

texture data clustering, VisTex image texture dataset is used and 4 classes are chosen in our experiments. We have used several performance metrics to examine the effectiveness of our algorithm in data clustering. We also have used AGMM for data clustering in all proposed experiments in order to have a comparison with our approach. From the set of experiments on all datasets and in the light of results achieved based on performance metrics, it is concluded that BAGMM has performed better in data modeling and data clustering as compared to AGMM. Due to great success of BAGMM in image and spambase datasets, for our future work, we propose the application of BAGMM in speech and video datasets to explore its modeling capabilities on different kinds of data.

5.7 Bounded Asymmetric Generalized Gaussian Mixture Model with MML for Model Selection

To solve problem of symmetry, the asymmetric Gaussian distribution (AGD) is considered with two parameters dictating the left and right region of the distribution [264, 293]. All the approaches mentioned above are based on unbounded support distributions, hence called unbounded support mixtures. However, in many real applications, data are compactly supported [61–64] and bounded support mixture models have been applied to many applications in speech and image processing [18, 60, 294].

Due to the success of asymmetric distributions in many learning applications [32, 56, 58, 59], it is proposed to extend the idea of bounded support mixtures with asymmetric distributions and in this chapter, bounded support asymmetric generalized Gaussian mixture model (BAGGMM) is presented and its parameters are estimated with EM and Newton Raphson method. The proposed algorithm is applied to various clustering applications to examine viability and effectiveness in data modeling. In this chapter, different image clustering tasks are selected for conducting the experiments using BAGGMM. The first experiment is performed for spam detection using Spam Hunter dataset. In the second experiment, object recognition is performed by using two datasets (ETHZ and GHIM) and multiple clustering scenarios are performed. With ETHZ dataset, experiment is performed with 5 clusters for object recognition on a very small dataset which makes it unique to validate the performance of the model due to it size and categories of data. With GHIM, two clustering experiments are conducted for object recognition by choosing 5 clusters in each experiment. In the third experiment, visual scene categorization is chosen and clustering framework is implemented and tested using 15-Scene and GHIM datasets and multiple clustering scenarios are created. For the validation of clustering performance in visual scene categorization, 4 experiments are created with 15-Scene (2 with 4 clusters and 2 with 5 clusters) and 2 experiments with GHIM

dataset (5 clusters in each). The performance of proposed model in all applications is compared with AGGMM.

Model selection is another crucial part, in which the optimal number of components that best fits the dataset is estimated through the inference process. We propose Minimum message length (MML) based on information theory for our model as it has shown great performances in previous research works [113, 114, 116]. The proposed model selection criterion for BAGGMM using MML is validate through several experiments and all the datasets used in clustering experiments are selected to test the model selection criterion. The experiments for model selection are conducted for multiple clustering scenarios and 10 different experimental scenarios created with 2, 3, 4 and 5 clusters. The results of each experiment are also compared with 7 different model selection criteria to examine its performance in finding the optimal number of clusters.

5.8 Proposed Model

In order to overcome the problems associated with unbounded support range, a new distribution (bounded Gaussian distribution) was presented in [62]. In this section, same idea is extended and bounded support Asymmetric Generalized Gaussian mixture model is presented.

5.8.1 Mixture of Asymmetric Generalized Gaussian Distributions

Asymmetric Gaussian mixture model was proposed to handle the asymmetric properties present in different kinds of data [56, 58, 295]. For a univariate data, if one data sample is represented by X, then asymmetric Gaussian distribution is represented as follows:

$$f(X|\mu,\sigma_l,\sigma_r,\lambda) = \frac{\lambda\left[\frac{\Gamma(3/\lambda)}{\Gamma(1/\lambda)}\right]^{1/2}}{(\sigma_l+\sigma_r)\Gamma(1/\lambda)} \times \begin{cases} \exp\left[-A(\lambda)\left(\frac{\mu-X}{\sigma_l}\right)^{\lambda}\right] & \text{if} \quad X<\mu \\ \exp\left[-A(\lambda)\left(\frac{X-\mu}{\sigma_r}\right)^{\lambda}\right] & \text{if} \quad X\geq\mu \end{cases} \tag{5.33}$$

where parameters of distribution μ, σ_l, σ_r & λ are mean, left standard deviation, right standard deviation and shape parameters, respectively. The parameters of AGMM are estimated using ML estimate and complete parameter estimation is explained in [56, 58, 295]. In Fig. (5.10), graphical representation of AGMM is shown, where X_i is a data point with $i = 1,...,N$, μ, σ_l, σ_r and λ, parameters of distribution and p and Z_i are mixing weight and posterior probability in a mixture model and they are explained in details in Section 5.8.2.

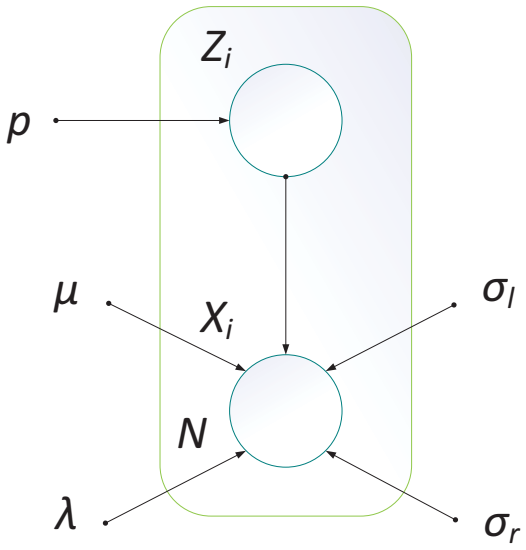


Figure 5.10: Graphical representation of Asymmetric Generalized Gaussian mixture model

5.8.2 Mixture of Bounded Asymmetric Gaussian Distribution for Multidimensional Data

Consider that a D-dimensional random variable $\vec{X} = (X_1, ..., X_D)$, follows a K components mixture distribution if its probability function can be written in the following form:

$$p(\vec{X}|\Theta) = \sum_{j=1}^{K} p(\vec{X}|\xi_j) p_j \tag{5.34}$$

provided $p_j \geq 0$, $\sum_{j=1}^{K} p_j = 1$, $\Theta = (\xi_1, \xi_2, \xi_3, \xi_4, \xi_5)$ with $\xi_1 = (\vec{\mu}_1, ..., \vec{\mu}_K)$, $\xi_2 = (\vec{\sigma}_{l_1}, ..., \vec{\sigma}_{l_K})$, $\xi_3 = (\vec{\sigma}_{r_1}, ..., \vec{\sigma}_{r_K})$, $\xi_4 = (\vec{\lambda}_1, ..., \vec{\lambda}_K)$ and $\xi_5 = (p_1, ..., p_K)$. The term $p(\vec{X}|\xi_j)$ is BAGGD for the vector $\vec{X}$ and defined as:

$$p(\vec{X}|\xi_j) = \frac{f(\vec{X}|\xi_j)\mathrm{H}(\vec{X}|j)}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \tag{5.35}$$

$$\text{where } \mathrm{H}(\vec{X}|j) = \begin{cases} 1 & \text{if } \vec{X} \in \partial_j \\ 0 & \text{otherwise} \end{cases} \tag{5.36}$$

$$f(\vec{X}|\xi_j)=\prod_{k=1}^{d}\begin{cases}\frac{\lambda_{jd}\left[\frac{\Gamma\left(3/\lambda_{jd}\right)}{\Gamma\left(1/\lambda_{jd}\right)}\right]^{1/2}}{\left(\sigma_{l_{jd}}+\sigma_{r_{jd}}\right)\Gamma\left(1/\lambda_{jd}\right)}\exp\left[-A\left(\lambda_{jd}\right)\left(\frac{\mu_{jd}-X_d}{\sigma_{l_{jd}}}\right)^{\lambda_{jd}}\right] & \text{if} \quad X_k<\mu_{jd}\\ \\ \frac{\lambda_{jd}\left[\frac{\Gamma\left(3/\lambda_{jd}\right)}{\Gamma\left(1/\lambda_{jd}\right)}\right]^{1/2}}{\left(\sigma_{l_{jd}}+\sigma_{r_{jd}}\right)\Gamma\left(1/\lambda_{jd}\right)}\exp\left[-A\left(\lambda_{jd}\right)\left(\frac{X_d-\mu_{jd}}{\sigma_{r_{jd}}}\right)^{\lambda_{jd}}\right] & \text{if} \quad X_k\geq\mu_{jd}\end{cases} \tag{5.37}$$

where $\vec{\mu}_j=(\mu_{j1},...,\mu_{jD})$, $\vec{\sigma}_{l_j}=(\sigma_{l_{j1}},...,\sigma_{l_{jD}})$, $\vec{\sigma}_{r_j}=(\sigma_{r_{j1}},...,\sigma_{r_{jD}})$, $\vec{\lambda}_j=(\lambda_{j1},...,\lambda_{jD})$ are the mean, left standard deviation, right standard deviation and shape parameters of the D-dimensional BAGGD, respectively. The term $\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}$ in Eq. (5.35) is the normalization constant that indicates the share of $f(\vec{X}|\xi_j)$ which belongs to the support region ∂. The AGGD $f(\vec{X}|\xi_j)$ can also be defined as:

$$f(\vec{X}|\xi_j)=\begin{cases}\mathrm{g}_1(\vec{X}|\xi_j) & \text{if} \quad \mathrm{X}_d<\mu_{jd}\\ \\ \mathrm{g}_2(\vec{X}|\xi_j) & \text{if} \quad \mathrm{X}_d\geq\mu_{jd}\end{cases} \tag{5.38}$$

where

$$\mathrm{g}_1(\vec{X}|\xi_j)=\prod_{d=1}^{D}\frac{\lambda_{jd}\left[\frac{\Gamma\left(3/\lambda_{jd}\right)}{\Gamma\left(1/\lambda_{jd}\right)}\right]^{1/2}}{\left(\sigma_{l_{jd}}+\sigma_{r_{jd}}\right)\Gamma\left(1/\lambda_{jd}\right)}\exp\left[-A\left(\lambda_{jd}\right)\left(\frac{\mu_{jd}-X_k}{\sigma_{l_{jd}}}\right)^{\lambda_{jd}}\right] \tag{5.39}$$

$$\mathrm{g}_2(\vec{X}|\xi_j)=\prod_{d=1}^{D}\frac{\lambda_{jd}\left[\frac{\Gamma\left(3/\lambda_{jd}\right)}{\Gamma\left(1/\lambda_{jd}\right)}\right]^{1/2}}{\left(\sigma_{l_{jd}}+\sigma_{r_{jd}}\right)\Gamma\left(1/\lambda_{jd}\right)}\exp\left[-A\left(\lambda_{jd}\right)\left(\frac{X_k-\mu_{jd}}{\sigma_{l_{jd}}}\right)^{\lambda_{jd}}\right] \tag{5.40}$$

Consider the case where the input is set of vectors represented as $\mathscr{X}=(\vec{X}_1,...,\vec{X}_N)$. With a mixture of K BAGDs, the distribution of $\mathscr{X}$ is given by:

$$p(\mathscr{X}|\Theta)=\prod_{i=1}^{N}\sum_{j=1}^{K}p(\vec{X}_i|\xi_j)p_j \tag{5.41}$$

Stochastic indicator vectors $\vec{Z}_i=(Z_{i1},...,Z_{iK})$, one for each observation are introduced. The role is to encode the membership of each observation for a relative component of the mixture model. In other words, Z_{ij}, the unobserved variable in each indicator vector, equals 1 if $\vec{X}_i$ belongs to class j and 0, otherwise. The complete data likelihood is given by

$$p(\mathscr{X},\mathscr{Z}|\Theta)=\prod_{i=1}^{N}\prod_{j=1}^{K}\left(p(\vec{X}_i|\xi_j)p_j\right)^{Z_{ij}} \tag{5.42}$$

where $\hat{Z}_{ij}$ is the posterior probability and its expectation can be written as:

$$\hat{Z}_{ij} = p(j|\vec{X}_i) = \frac{p(\vec{X}_i|\xi_j)p_j}{\sum_{j=1}^{K} p(\vec{X}_i|\xi_j)p_j} \tag{5.43}$$

and $\mathscr{Z} = \{\vec{Z}_1, ..., \vec{Z}_N\}$.

5.8.3 Parameters Learning

The parameters are estimated from the maximization of log-likelihood function which can be written as:

$$\mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{K} \hat{Z}_{ij} \log\left(p(\vec{X}_i|\xi_j)p_j\right) \tag{5.44}$$

$$\mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{K} Z_{ij}\left\{\log p_j + \log f(\vec{X}_i|\xi_j) + \log \mathrm{H}(\vec{X}_i|j) - \log\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right\} \tag{5.45}$$

The complete-data log-likelihood can be maximized with respect to the model parameters. This can be done by taking the gradient of the log-likelihood with respect to p_j, μ_j, σ_{l_j}, σ_{r_j} and λ_j. The estimation of mixing parameter is discussed in Section 2.2.3.1. Estimation of rest of the parameters for bounded support asymmetric generalized Gaussian mixture model is explained below.

5.8.3.1 Mean Parameter Estimation

Updated value of Mean μ_{jd}, can be estimated by maximizing the log-likelihood function given in Eq. (5.45) with respect to $\vec{\mu}_j$. Taking the first derivative of log-likelihood with respect to μ_j does not give a closed form solution which can be observed from Appendix E.1. The parameters are estimated using Newton Raphson method which requires the computation of first and second order derivatives of log-likelihood with respect to μ_j which are given in Appendix E.1 & E.2. Note that, in Appendix E.1, the term $\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)(\mu_{jd} - \mathrm{u})^{\lambda_{jd}-1}d\mathrm{u}$ is the expectation of function $(\mu_{jd} - \mathrm{u})^{\lambda_{jd}-1}$ under the probability distribution $\mathrm{g}_1(\mathrm{u}|\xi_j)$. Then, this expectation can be approximated as:

$$\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)(\mu_{jd} - \mathrm{u})^{\lambda_{jd}-1}d\mathrm{u} \approx \frac{1}{M}\sum_{m=1}^{M}(\mu_{jd} - \mathrm{l}_{m_{jd}})^{\lambda_{jd}-1}\mathrm{H}(\mathrm{l}_{m_{jd}}|j) \tag{5.46}$$

where $\mathrm{l}_{m_{jd}} \sim \mathrm{g}_1(\mathrm{u}|\xi_j)$ is a set of random variables drawn from the asymmetric generalized Gaussian distribution with $\mathrm{u} < \mu_{jd}$ for the particular component j of the mixture model. The set of data with random variables have M vectors with D dimensions. M is a large integer chosen to generate the

set of random variables. Similarly, the term $\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)d\mathrm{u}$ in Appendix E.1 can be approximated as:

$$\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)d\mathrm{u} \approx \frac{1}{M}\sum_{m=1}^{M} \mathrm{H}(\mathrm{I}_{m_{jd}}|j) \tag{5.47}$$

In a similar manner, the terms $\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^{\lambda_{jd}-1}d\mathrm{u}$ and $\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)d\mathrm{u}$ are approximated as follows:

$$\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^{\lambda_{jd}-1}d\mathrm{u} \approx \frac{1}{M}\sum_{m=1}^{M} (\mathrm{r}_{m_{jd}}-\mu_{jd})^{\lambda_{jd}-1}\mathrm{H}(\mathrm{r}_{m_{jd}}|j) \tag{5.48}$$

$$\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)d\mathrm{u} \approx \frac{1}{M}\sum_{m=1}^{M} \mathrm{H}(\mathrm{r}_{m_{jd}}|j) \tag{5.49}$$

where $\mathrm{r}_{m_{jd}} \sim \mathrm{g}_2(X_d|\xi_j)$ is a set of random variables drawn from the asymmetric generalized Gaussian distribution with $\mathrm{u} \geq \mu_{jd}$ for the particular component j of the mixture model. Further approximation in a similar manner can applied on Appendix E.1 & E.2 and it is presented in Eqs. (5.50 & 5.51) and it is further used in the Newton Raphson method for parameter estimation (Eq. (5.52)), which is applied within EM algorithm for parameter estimation.

$$\begin{aligned}\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \mu_{jd}} = A(\lambda_{jd})\lambda_{jd}\Bigg[&\sum_{i=1,X_{id}\geq\mu_{jd}}^{N} Z_{ij}\frac{\left(X_{id}-\mu_{jd}\right)^{\lambda_{jd}-1}}{\sigma_{r_{jd}}^{\lambda_{jd}}} \\ &-\sum_{i=1,X_{id}<\mu_{jd}}^{N} Z_{ij}\frac{\left(\mu_{jd}-X_{id}\right)^{\lambda_{jd}-1}}{\sigma_{l_{jd}}^{\lambda_{jd}}} \\ &-\sum_{i=1,X_{id}\geq\mu_{jd}}^{N} Z_{ij}\left\{\frac{\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|j)\frac{(\mathrm{r}_{m_{jd}}-\mu_{jd})^{\lambda_{jd}-1}}{\sigma_{r_{jd}}^{\lambda_{jd}}}}{\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|j)}\right\} \\ &+\sum_{i=1,X_{id}<\mu_{jd}}^{N} Z_{ij}\left\{\frac{\sum_{m=1}^{M}\mathrm{H}(\mathrm{l}_{m_{jd}}|j)\frac{(\mu_{jd}-\mathrm{l}_{m_{jd}})^{\lambda_{jd}-1}}{\sigma_{l_{jd}}^{\lambda_{jd}}}}{\sum_{m=1}^{M}\mathrm{H}(\mathrm{l}_{m_{jd}}|j)}\right\}\Bigg]\end{aligned} \tag{5.50}$$

$$\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \mu_{jd}^2} = A(\lambda_{jd})\lambda_{jd}(\lambda_{jd}-1)\left[-\sum_{i=1,X_{ik}<\mu_{jd}}^{N} \hat{Z}_{ij}\frac{\left(\mu_{jd}-X_{ik}\right)^{\lambda_{jd}-2}}{\sigma_{l_{jd}}^{\lambda_{jd}}}\right. \tag{5.51}$$

$$
\begin{aligned}
&- \sum_{i=1, X_{ik} \geq \mu_{jd}}^{N} \hat{Z}_{ij} \frac{\left(X_{ik} - \mu_{jd}\right)^{\lambda_{jd}-2}}{\sigma_{r_{jd}}^{\lambda_{jd}}} \Bigg] \\
&- A(\lambda_{jd}) \frac{\lambda_{jd}}{\sigma_{jd}^{\lambda_{jd}}} \Bigg[\sum_{i=1, X_{id} < \mu_{jd}}^{N} Z_{ij} \Bigg\{ \frac{-A(\lambda_{jd}) \sum_{m=1}^{M} \mathrm{H}(\mathrm{I}_{m_{jd}}|j) \frac{(\mu_{jd} - \mathrm{I}_{m_{jd}})^{2(\lambda_{jd}-1)}}{\sigma_{l_{jd}}^{\lambda_{jd}}}}{\sum_{m=1}^{M} \mathrm{H}(\mathrm{I}_{m_{jd}}|j)} \\
&+ \frac{\sum_{m=1}^{M} \mathrm{H}(\mathrm{I}_{m_{jd}}|j)(\lambda_{jd}-1)(\mu_{jd} - \mathrm{I}_{m_{jd}})^{\lambda_{jd}-2}}{\sum_{m=1}^{M} \mathrm{H}(\mathrm{I}_{m_{jd}}|j)} \Bigg\} \\
&+ A(\lambda_{jd}) \frac{\lambda_{jd}}{\sigma_{jd}^{\lambda_{jd}}} \left(\frac{\sum_{m=1}^{M} \mathrm{H}(\mathrm{I}_{m_{jd}}|j)(\mu_{jd} - \mathrm{I}_{m_{jd}})^{\lambda_{jd}-1}}{\sum_{m=1}^{M} \mathrm{H}(\mathrm{I}_{m_{jd}}|j)} \right)^{2} \Bigg] \\
&+ A(\lambda_{jd}) \frac{\lambda_{jd}}{\sigma_{jd}^{\lambda_{jd}}} \Bigg[\sum_{i=1, X_{id} \geq \mu_{jd}}^{N} \hat{Z}_{ij} \Bigg\{ \frac{\sum_{m=1}^{M} \mathrm{H}(\mathrm{r}_{m_{jd}}|j) A(\lambda_{jd}) \frac{(\mathrm{r}_{m_{jd}} - \mu_{jd})^{2(\lambda_{jd}-1)}}{\sigma_{r_{jd}}^{\lambda_{jd}}}}{\sum_{m=1}^{M} \mathrm{H}(\mathrm{r}_{m_{jd}}|j)} \\
&+ \frac{\sum_{m=1}^{M} \mathrm{H}(\mathrm{r}_{m_{jd}}|j)(\lambda_{jd}-1)(\mathrm{r}_{m_{jd}} - \mu_{jd})^{\lambda_{jd}-2} d\mathrm{u}}{\sum_{m=1}^{M} \mathrm{H}(\mathrm{r}_{m_{jd}}|j)} \Bigg\} \\
&+ A(\lambda_{jd}) \frac{\lambda_{jd}}{\sigma_{jd}^{\lambda_{jd}}} \left(\frac{\sum_{m=1}^{M} \mathrm{H}(\mathrm{r}_{m_{jd}}|j)(\mathrm{r}_{m_{jd}} - \mu_{jd})^{\lambda_{jd}-1} d\mathrm{u}}{\sum_{m=1}^{M} \mathrm{H}(\mathrm{r}_{m_{jd}}|j)} \right)^{2} \Bigg]
\end{aligned}
$$

$$
\hat{\mu}_{jd} \simeq \mu_{jd} - \left[\left(\frac{\partial^2 \mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta)}{\partial {\mu^2}_{jd}} \right)^{-1} \left(\frac{\partial \mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta)}{\partial \mu_{jd}} \right) \right] \tag{5.52}
$$

5.8.3.2 Left Standard Deviation Estimation

The new value of left standard deviation $\sigma_{l_{jd}}$, can be estimated by maximizing the log-likelihood function given in Eq. (5.45) with respect to $\vec{\sigma}_{l_j}$ and it is observed that from the derivative given in Appendix E.3, that a closed form solution for this parameter does not exist and Newtons Raphson method is used. We have computed the first order and second order derivatives with respect to left standard deviation, which is presented in Appendix E.3 & E.4. The integral terms in the derivatives (Appendix E.3 & E.4) are approximated in similar fashion as described in Section 5.8.3.1 and updated equation for derivatives with respect to left standard deviation are presented in Eq. (5.53 & 5.54) which are further used in Netwon Rphson method as in Eq. (5.55).

$$
\frac{\partial \mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta)}{\partial \sigma_{l_{jd}}} = \sum_{i=1, X_{id} < \mu_{jd}}^{N} \hat{Z}_{ij} \frac{A(\lambda_{jd}) \lambda_{jd}}{\sigma_{l_{jd}}} \left(\frac{\mu_{jd} - X_{id}}{\sigma_{l_{jd}}} \right)^{\lambda_{jd}} \tag{5.53}
$$

$$\sum_{i=1,X_{id}<\mu_{jd}}^{N} \hat{Z}_{ij} \left\{ \frac{\sum_{m=1}^{M} \mathrm{H}(\mathrm{I}_{m_{jd}}|j) \frac{A(\lambda_{jd})\lambda_{jd}(\mu_{jd}-\mathrm{l}_{m_{jd}})^{\lambda_{jd}}}{\sigma_{l_{jd}}^{\lambda_{jd}+1}}}{\sum_{m=1}^{M} \mathrm{H}(\mathrm{I}_{m_{jd}}|j)} \right\}$$

$$\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial {\sigma^2}_{l_{jd}}} = -\sum_{i=1,X_{id}<\mu_{jd}}^{N} \hat{Z}_{ij} \frac{A(\lambda_{jd})\lambda_{jd}(\lambda_{jd}+1)}{\sigma_{l_{jd}}^2} \left(\frac{\mu_{jd}-\mathrm{X}_{id}}{\sigma_{l_{jd}}}\right)^{\lambda_{jd}} \tag{5.54}$$
$$-\sum_{i=1,X_{id}<\mu_{jd}}^{N} \hat{Z}_{ij} A(\lambda_{jd}) \frac{\lambda_{jd}}{\sigma_{l_{jd}}^{\lambda_{jd}+1}} \left(1+\frac{1}{(\sigma_{l_{jd}}+\sigma_{r_{jd}})}\right) \frac{\sum_{m=1}^{M}(\mu_{jd}-X_{jd})^{\lambda_{jd}}\mathrm{H}(\mathrm{I}_{m_{jd}}|j)}{\sum_{m=1}^{M}\mathrm{H}(\mathrm{I}_{m_{jd}}|j)}$$
$$-\sum_{i=1,X_{id}<\mu_{jd}}^{N} \hat{Z}_{ij} \left(A(\lambda_{jd}) \frac{\lambda_{jd}}{\sigma_{l_{jd}}^{\lambda_{jd}+1}}\right)^2 \frac{\sum_{m=1}^{M}(\mu_{jd}-X_{jd})^{2\lambda_{jd}}\mathrm{H}(\mathrm{I}_{m_{jd}}|j)}{\sum_{m=1}^{M}\mathrm{H}(\mathrm{I}_{m_{jd}}|j)}$$
$$+\sum_{i=1,X_{id}<\mu_{jd}}^{N} \hat{Z}_{ij} A(\lambda_{jd}) \frac{\lambda_{jd}(\lambda_{jd}+1)}{\sigma_{l_{jd}}^{\lambda_{jd}+2}} \frac{\sum_{m=1}^{M}(\mu_{jd}-X_{jd})^{\lambda_{jd}}\mathrm{H}(\mathrm{I}_{m_{jd}}|j)}{\sum_{m=1}^{M}\mathrm{H}(\mathrm{I}_{m_{jd}}|j)}$$
$$+\sum_{i=1,X_{id}<\mu_{jd}}^{N} \hat{Z}_{ij} \left(A(\lambda_{jd}) \frac{\lambda_{jd}(\lambda_{jd}+1)}{\sigma_{l_{jd}}^{\lambda_{jd}+2}}\right)^2 \frac{\left(\sum_{m=1}^{M}(\mu_{jd}-X_{jd})^{\lambda_{jd}}\mathrm{H}(\mathrm{I}_{m_{jd}}|j)\right)^2}{\left(\sum_{m=1}^{M}\mathrm{H}(\mathrm{I}_{m_{jd}}|j)\right)^2}$$

$$\hat{\sigma}_{l_{jd}} \simeq \sigma_{l_{jd}} - \left[\left(\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial {\sigma^2}_{l_{jd}}}\right)^{-1} \left(\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma_{l_{jd}}}\right)\right] \tag{5.55}$$

5.8.3.3 Right Standard Deviation Estimation

Right standard deviation $\sigma_{r_{jd}}$, can be estimated by maximizing the log-likelihood function given in Eq. (5.45) with respect to $\vec{\sigma}_{r_j}$. The first order derivative presented in Appendix E.5 could not provide a closed form solution and second order derivative with regard to right standard deviation (Appendix E.6) is computed. Both derivatives are adopted in Newton Raphson method to estimate the parameters which is applied within EM algorithm. The integral terms in the derivatives (Appendix E.5 & E.6) are approximated in a similar fashion as described in the estimation of mean and left standard deviation and complete derivatives are represented as Eqs. (5.56 & 5.58).

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma_{r_{jd}}} = \sum_{i=1,X_{id}\geq\mu_{jd}}^{N} \hat{Z}_{ij} \frac{A(\lambda_{jd})\lambda_{jd}}{\sigma_{r_{jd}}} \left(\frac{X_{id}-\mu_{jd}}{\sigma_{r_{jd}}}\right)^{\lambda_{jd}} \tag{5.56}$$

$$\sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N}\hat{Z}_{ij}\left\{\frac{\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|j)\frac{A(\lambda_{jd})\lambda_{jd}(\mathrm{r}_{m_{jd}}-\mu_{jd})^{\lambda_{jd}}}{\sigma_{r_{jd}}^{\lambda_{jd}+1}}}{\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|j)}\right\} \tag{5.57}$$

$$\begin{aligned}\frac{\partial^2\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial\sigma^2{}_{r_{jd}}} &= -\sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N}\hat{Z}_{ij}\frac{A(\lambda_{jd})\lambda_{jd}(\lambda_{jd}+1)}{\sigma_{r_{jd}}^2}\left(\frac{\mathrm{X}_{id}-\mu_{jd}}{\sigma_{r_{jd}}}\right)^{\lambda_{jd}} \\ &- \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N}\hat{Z}_{ij}A(\lambda_{jd})\frac{\lambda_{jd}}{\sigma_{r_{jd}}^{\lambda_{jd}+1}}\left(1+\frac{1}{(\sigma_{l_{jd}}+\sigma_{r_{jd}})}\right)\frac{\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|j)(\mu_{jd}-\mathrm{r}_{m_{jd}})^{\lambda_{jd}}}{\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|j)} \\ &- \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N}\hat{Z}_{ij}\left(A(\lambda_{jd})\frac{\lambda_{jd}}{\sigma_{r_{jd}}^{\lambda_{jd}+1}}\right)^2\frac{\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|j)(\mu_{jd}-\mathrm{r}_{m_{jd}})^{2\lambda_{jd}}}{\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|j)} \\ &+ \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N}\hat{Z}_{ij}A(\lambda_{jd})\frac{\lambda_{jd}(\lambda_{jd}+1)}{\sigma_{r_{jd}}^{\lambda_{jd}+2}}\frac{\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|j)(\mu_{jd}-\mathrm{r}_{m_{jd}})^{\lambda_{jd}}}{\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|j)} \\ &+ \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N}\hat{Z}_{ij}\left(A(\lambda_{jd})\frac{\lambda_{jd}(\lambda_{jd}+1)}{\sigma_{r_{jd}}^{\lambda_{jd}+2}}\right)^2\frac{\left(\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|j)(\mu_{jd}-\mathrm{r}_{m_{jd}})^{\lambda_{jd}}\right)^2}{\left(\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|j)\right)^2}\end{aligned} \tag{5.58}$$

$$\sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N}Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma_{r_{jd}}^3}\right)-\sum_{i=1,\mathrm{u}\geq\mu_{jd}}^{N}\frac{Z_{ij}}{\sigma_{r_{jd}}^3}\left\{\frac{\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)d\mathrm{u}(\mathrm{u}-\mu_{jd})^2d\mathrm{u}}{\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)d\mathrm{u}}\right\}=0 \tag{5.59}$$

$$\hat{\sigma}_{r_{jd}}\simeq\sigma_{r_{jd}}-\left[\left(\frac{\partial^2\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial\sigma^2{}_{r_{jd}}}\right)^{-1}\left(\frac{\partial\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial\sigma_{r_{jd}}}\right)\right] \tag{5.60}$$

5.8.3.4 Shape Parameter Estimation

The shape parameter is also estimated by taking the first and second order derivatives of log-likelihood and Newton Raphson method is applied in the EM algorithm and it was needed because first order derivative in the maximum likelihood estimate does not provide a closed form solution as it can be observed from the first order derivative presented in Appendix E.7.

$$\frac{\partial\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial\lambda_{jd}}=0 \tag{5.61}$$

For parameter estimation, first order derivative of the log-likelihood $\frac{\partial L(\Theta,Z,\mathscr{X})}{\partial \lambda_{jd}}$ is computed as follows:

$$\frac{\partial \mathscr{L}(\Theta,Z,\mathscr{X})}{\partial \lambda_{jd}} = \sum_{i=1}^{N}\sum_{j=1}^{K} Z_{ij}\left\{\frac{\partial}{\partial \lambda_{jd}}\log f(\vec{X}_i|\xi_j) - \frac{\partial}{\partial \lambda_{jd}}\log\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right\} \tag{5.62}$$

The computation of first term in the log-likelihood is $\frac{\partial}{\partial \lambda_{jd}}\log f(X_{id}|\xi_j)$ is adopted from [58] and we denote it here as $h(X_{id}|\xi_j)$. The computation of second term in Eq. (5.62) is as:

$$\frac{\partial}{\partial \lambda_{jd}}\log\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u} = \frac{\frac{\partial}{\partial \lambda_{jd}}\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}} = \frac{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)h(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}} \tag{5.63}$$

The complete procedure for taking the first order derivative with regard to shape parameter is given in Appendix E.7. The term $\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)h(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}$ can be approximated similar to Section 5.8.3.1.

$$\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)h(\vec{\mathrm{u}}|\xi_j)d\mathrm{u} \approx \frac{1}{M}\sum_{m=1}^{M} h(s_{j_{md}}|\xi_j)\mathrm{H}(s_{j_{md}}|j) \tag{5.64}$$

The complete expression for first order derivative for the shape parameter is expressed as follows:

$$\frac{\partial \mathscr{L}(\Theta,Z,\mathscr{X})}{\partial \lambda_{jd}} = \sum_{i=1}^{N}\sum_{j=1}^{K} Z_{ij}\left\{h(X_{id}|\xi_j) - \frac{\sum_{m=1}^{M} h(s_{j_{md}}|\xi_j)\mathrm{H}(s_{j_{md}}|j)}{\sum_{m=1}^{M}\mathrm{H}(s_{j_{md}}|j)}\right\} \tag{5.65}$$

The computation of second order derivative for shape parameter $\frac{\partial^2 \mathscr{L}(\Theta,Z,\mathscr{X})}{\partial {\lambda^2}_{jd}}$ is presented as:

$$\frac{\partial^2 \mathscr{L}(\Theta,Z,\mathscr{X})}{\partial {\lambda^2}_{jd}} = \sum_{i=1}^{N}\sum_{j=1}^{K} Z_{ij}\left\{\frac{\partial}{\partial \lambda_{jd}}\left(\frac{\partial}{\partial \lambda_{jd}}\log f(\vec{X}_i|\xi_j)\right) - \frac{\partial}{\partial \lambda_{jd}}\left(\frac{\partial}{\partial \lambda_j}\log\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right)\right\} \tag{5.66}$$

The computation of $\frac{\partial}{\partial \lambda_{jd}}\left(\frac{\partial}{\partial \lambda_{jd}}\log f(\vec{X}_i|\xi_j)\right)$ is influenced by [18, 47, 58] and denoted as $h^{'}(\vec{X}_i|\xi_j)$ and it is provided in Appendix E.8. The computation of second term in Eq. (5.66) is as:

$$\frac{\partial}{\partial \lambda_{jd}}\left(\frac{\partial}{\partial \lambda_j}\log\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right) = \left\{\frac{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)\{h^2(\vec{\mathrm{u}}|\xi_j)+h^{'}(\vec{\mathrm{u}}|\xi_j)\}d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}} - \frac{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)h(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right)^2}{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right)^2}\right\} \tag{5.67}$$

The complete procedure for taking the second order derivative with regard to shape parameter is provided in Appendix E.8. The term $\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)\{h^2(\vec{\mathrm{u}}|\xi_j)+h^{'}(\vec{\mathrm{u}}|\xi_j)\}d\mathrm{u}$ can be approximated as:

$$\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)\{h^2(\vec{\mathrm{u}}|\xi_j)+h^{'}(\vec{\mathrm{u}}|\xi_j)\}d\mathrm{u} \approx \frac{1}{M}\sum_{m=1}^{M}\left\{h^2(s_{j_{md}}|\xi_j)+h^{'}(s_{j_{md}}|\xi_j)\right\}\mathrm{H}(s_{j_{md}}|j) \tag{5.68}$$

The complete expression for second order derivative after the approximations is as follows:

$$\frac{\partial^2 \mathscr{L}(\Theta, Z, \mathscr{X})}{\partial {\lambda^2}_{jd}} = \sum_{i=1}^{N}\sum_{j=1}^{K} Z_{ij}\left\{h^{'}(\vec{X}_i|\xi_j) - \frac{\sum_{m=1}^{M}\left\{h^2(s_{j_{md}}|\xi_j)+h^{'}(s_{j_{md}}|\xi_j)\right\}\mathrm{H}(s_{j_{md}}|j)}{\sum_{m=1}^{M}\mathrm{H}(s_{j_{md}}|j)} + \frac{\left(\sum_{m=1}^{M}h(s_{j_{md}}|\xi_j)\mathrm{H}(s_{j_{md}}|j)\right)^2}{\left(\sum_{m=1}^{M}\mathrm{H}(s_{j_{md}}|j)\right)^2}\right\} \tag{5.69}$$

$$\hat{\lambda}_{jd} \simeq \lambda_{jd} - \left[\left(\frac{\partial^2 \mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta)}{\partial {\lambda^2}_{jd}}\right)^{-1}\left(\frac{\partial \mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta)}{\partial \lambda_{jd}}\right)\right] \tag{5.70}$$

The complete learning of BAGMM is given in Algorithm 6, where t_{min} is minimum threshold used to monitor the convergence criteria in each iteration. In the initialization phase, K-Means is applied for computation of mean and data assignment in each cluster. This information is further used for computation of standard deviation and mixing weight during initialization phase. The initial value of shape parameter is set to 2.

5.9 Experiments and Results for Data Clustering

In order to test the performance of our model we apply it on distinct image clustering tasks with varying properties. Different clustering scenarios are created to examine the effectiveness of proposed model. We compare our model with Asymmetric Generalized Gaussian Mixture Model (AGGMM) model in our experiments. This helps to know how our model improves over AGGMM. In the first experiment we demonstrate our model for spam image clustering followed by other applications related to object and scene categorization.

5.9.1 Spam Detection in Image Datasets

Differentiating spam images from important ones is an essential task as it is capable of inducing harmful security attacks. However, we have to take care that none of the important messages are being falsely identified as spam because it might cause loss of necessary information. This means that we have to achieve a very low False Positive Rate (FPR) to prove the effectiveness of our model. For this experiment we use the image spam hunter dataset [1]. The dataset consists of 928 spam images obtained from real spam images collected over 6 months. The normal images in the dataset were randomly picked online. A few scanned documents were also added to make the application more challenging. The total of the normal images amounted to 810. Samples of this

[1] https://users.cs.northwestern.edu/ yga751/ML/ISH.htm

Algorithm 6 Model Learning for BAGMM

Input:Dataset $\mathscr{X} = \{\vec{X}_1, \ldots, \vec{X}_N\}$, t_{min}.
Output: Θ, $\mathscr{Z}$.
{Initialization}:
    K-Means Algorithm (Computation of $\vec{\mu}_1, \ldots, \vec{\mu}_K$ & cluster assignment)
        for all $1 \leq j \leq K$ do
            Computation of p_j
            Computation of $\{(\vec{\sigma}_{l_j}$ & $\vec{\sigma}_{r_j})$=$\vec{\sigma}_j\}$
            Set the $\{(\vec{\lambda}_j = 2\}$
        end for
{Expectation Maximization}:
while relative change in log-likelihood $\geq t_{min}$ do
    $\{$[**E Step**]$\}$:
        for all $1 \leq j \leq K$ do
            Compute $p(j|\vec{X}_i)$ for $i = 1, \ldots, N$. using Eq. (5.43).
        end for
    $\{$[**M step**]$\}$:
        for all $1 \leq j \leq K$ do
            Estimation of mixing parameter p_j using Eq. (2.12).
            Estimation of mean $\vec{\mu}_j$ using Eq. (5.52).
            Estimation of left standard deviation $\vec{\sigma}_{l_j}$ using Eq. (5.55).
            Estimation of right standard deviation $\vec{\sigma}_{r_j}$ using Eq. (5.60).
            Estimation of shape parameter $\vec{\lambda}_j$ using Eq. (5.70).
        end for
end while

dataset from both classes are presented in Fig. (5.11). For feature extraction from images, we used Scale Invariant Feature Transform (SIFT) [296]. The SIFT approach gives feature vectors of 128 dimensions corresponding to the keypoints identified in each image. We then use bag of visual words algorithm to form a histogram of the features, which are used as input to our model. The results obtained by our model are shown in Table 5.5 and confusion matrix of this experiment is presented in Fig. (5.12). It clearly shows that our model has a higher accuracy and precision when compared to the AGGMM. Also, the most important criteria for our evaluation in this test which is FPR is also pretty much lower than AGGMM.

Table 5.5: Performance of Spam Detection from Spam Hunter dataset based on different metrics

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BAGGMM	96.37	92.96	99.35	99.20	00.64	95.98	92.85	96.03	96.10
AGGMM	94.88	94.69	95.04	94.34	04.95	94.51	89.71	94.51	94.87

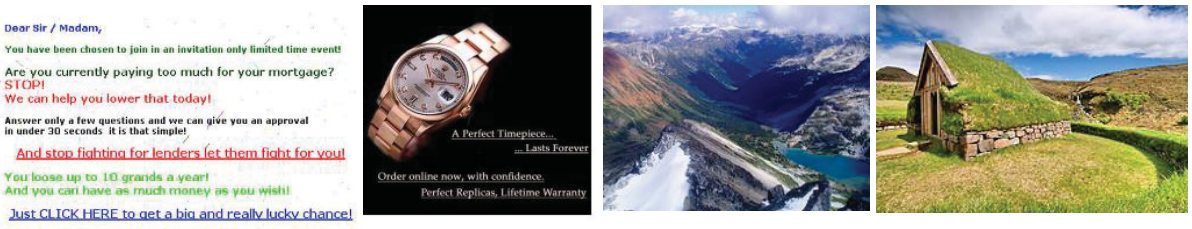


Figure 5.11: Samples of Spam Hunter Dataset (First two images from left are Spam and last two Ham)

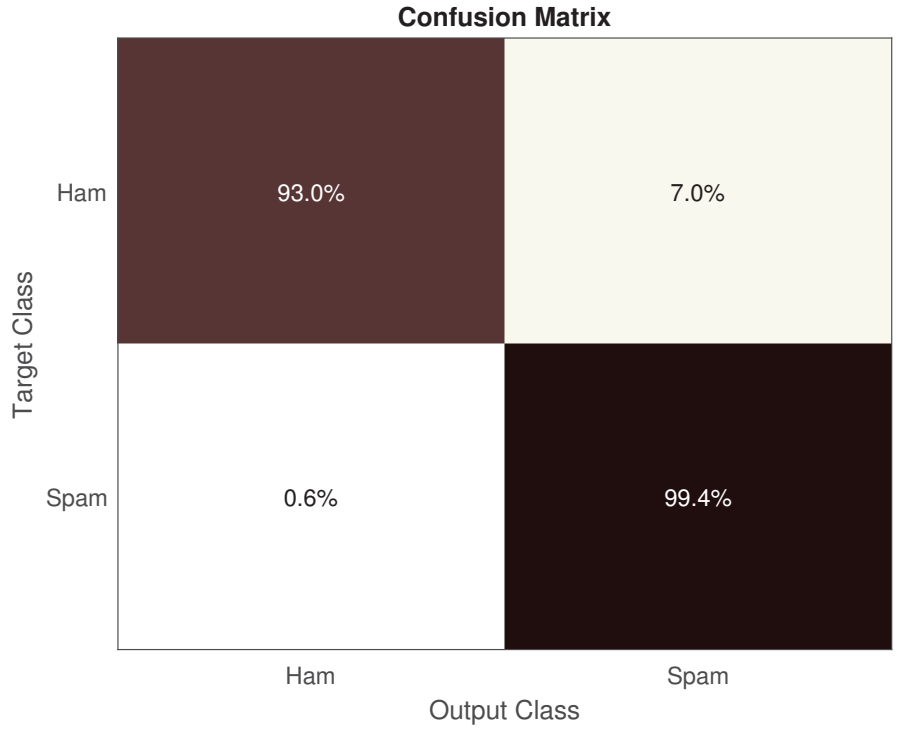


Figure 5.12: Confusion Matrix for Spam Detection with Spam Hunter dataset using BAGGMM

5.9.2 Object Clustering using ETHZ Dataset

Object clustering is one of the prime tasks in computer vision as it helps in applications such as image retrieval. In our experiment we use two challenging datasets. The first one is from the ETHZ dataset [2] which has 40 images of apple logos, 28 images of bottles, 87 images of giraffes, 48 images of mugs and 32 images of swans. This experiment helps us to evaluate our model when minimal data. It is to be noted that the number of images of giraffes in the dataset is twice as much as each of the other image categories. The sample images are presented in Fig. (5.13). The results are given in Table 5.6 which shows that our model was able to provide better performance than

[2]http://www.vision.ee.ethz.ch/en/datasets/

the AGGMM model. A high clustering accuracy, and very low FPR demonstrate the effectiveness of our model for object clustering. The confusion matrix to show the clustering results is given in Fig. (5.14).

Figure 5.13: Samples of ETHZ Dataset

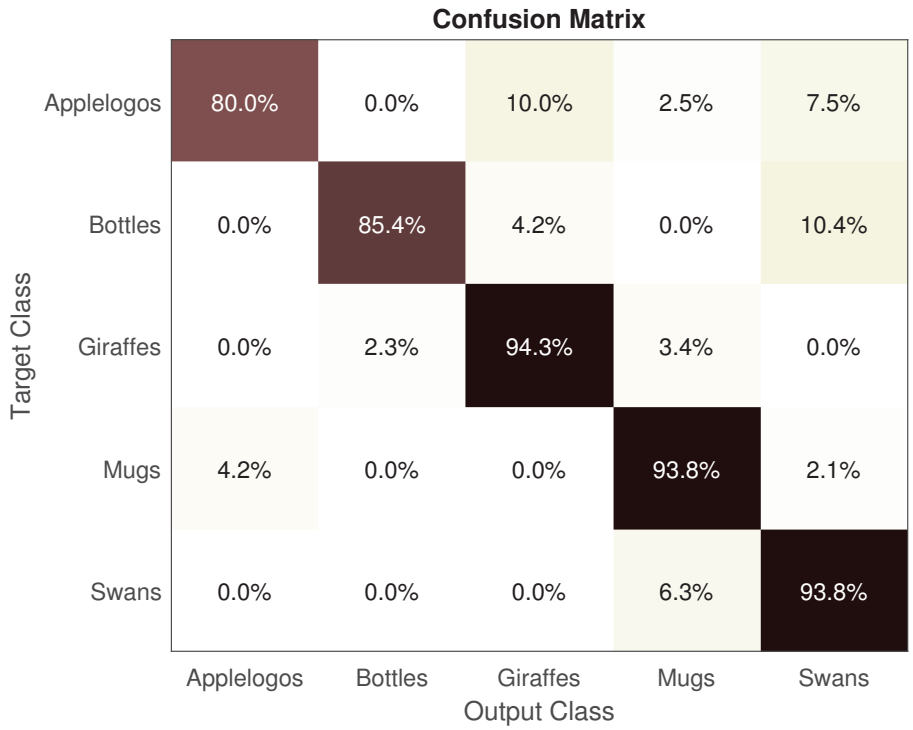


Figure 5.14: Confusion Matrix for ETHZ dataset with BAGGMM

Table 5.6: Performance of Object Categorization from ETHZ dataset based on different metrics

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BAGGMM	90.19	89.43	97.52	89.56	02.48	89.14	86.91	89.49	93.38
AGGMM	88.23	87.41	96.99	87.56	03.00	87.27	84.42	87.49	92.07

5.9.3 Object Clustering with GHIM Dataset

In the second experiment for object clustering, GHIM dataset [273] is chosen. It is composed of images with objects and natural scenes and for the experiment setup, it is divided into these two parts to investigate the performance of proposed model in object and scenes separately. The dataset is composed of 20 categories of images with 10 categories for natural scenes and 10 categories for objects. In the experiments for object clustering with GHM dataset, two clustering scenarios are created with 5 clusters in each experiment. The GHIM (objects) dataset is divided in to two parts to make complete use of all available data for the validation of proposed algorithm in object recognition. In both experiments, 400 images form each class are taken and first subset consisted of images of car, flower, plane, butterfly and bike and the second subset consisted of boat, ship, chicken, insects and horses and sample images are presented in Figs. (5.15 & 5.17). In all the experiments we used the same feature extraction as in the first application. Several performance measures are adopted for the validation of proposed application and results of both experiments are presented in Tables 5.7 and 5.8 which demonstrate the performance of BAGGMM. From both experiments, it is observed that proposed model has effectively improved the clustering performance as compared to AGGMM. Our experiments have shown increase in clustering accuracy and very low FPR. The best performances of both experiments are also presented as confusion matrix in Figs. (5.16 & 5.18) which present the clustering performance for each category.

Figure 5.15: Samples of GHIM (Objects) Dataset (Subset-1)

Table 5.7: Performance of Object Categorization for Ghim dataset (Objects) with 5 categories (subset-1)

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BAGGMM	86.32	86.32	96.58	86.37	03.42	86.31	82.92	86.35	91.30
AGGMM	85.15	85.15	96.28	85.29	03.71	85.16	81.50	85.22	90.55

5.9.4 Visual Scene Categorization with GHIM Dataset

Identifying the scene in a given image is an important information required in many automated decision making tasks involving computer vision. Hence, testing our model against datasets related

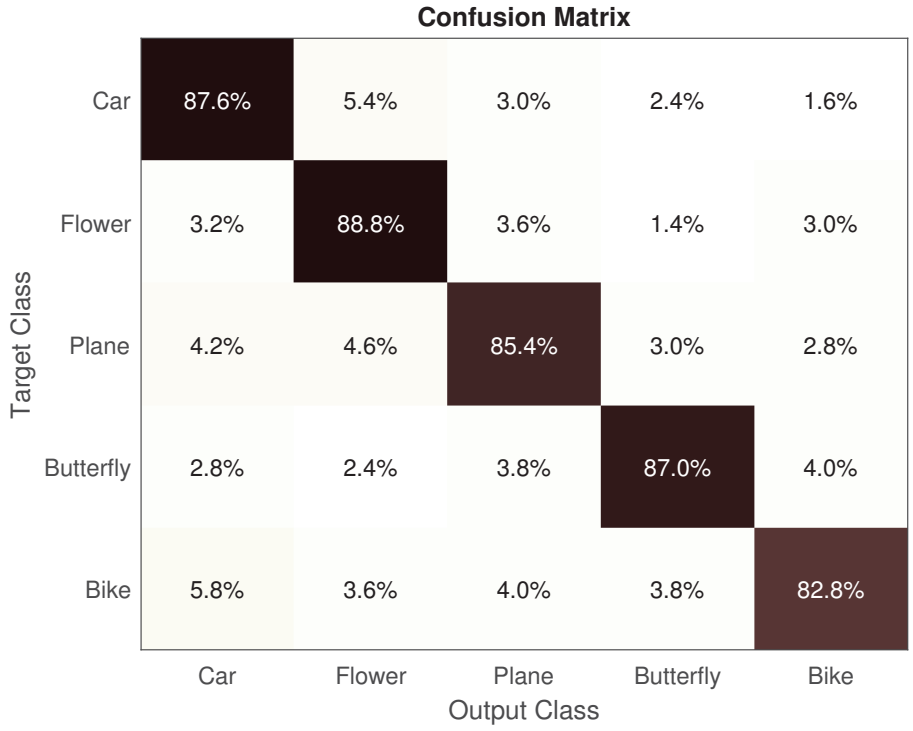


Figure 5.16: Confusion Matrix for Ghim dataset (Objects) using BAGGMM for 5 categories (subset-1)

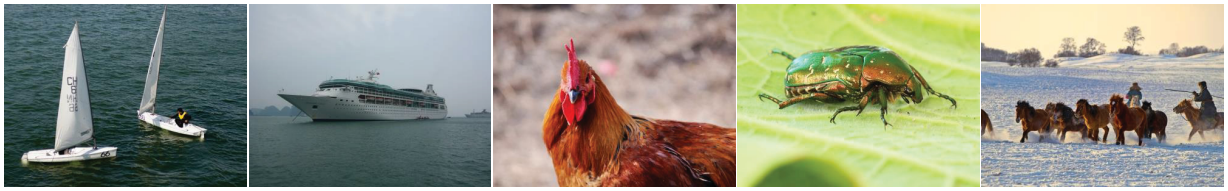


Figure 5.17: Samples of GHIM (Objects) Dataset (Subset-2)

Table 5.8: Performance of Object Categorization for Ghim dataset (Objects) with 5 categories (subset-2)

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BAGGMM	86.16	86.16	96.54	86.47	03.46	86.21	82.83	86.31	91.20
AGGMM	84.64	84.64	96.16	84.88	03.84	84.67	80.90	84.76	90.21

to natural scenes is an interesting experiment. In our first experiments for scene categorization, GHIM (Scene) dataset is selected which has 10 scene categories. For experiments with GHIM dataset for scene categorization, similar to the previous experiment, two clustering scenarios with 5 clusters in each experiment are selected. Two subsets from GHIM dataset are used where one subset contains images of fireworks, buildings, great wall of china, grass and mountains and the other contains images of trees, grass, Chinese buildings and sunset. In each experiment, 2000

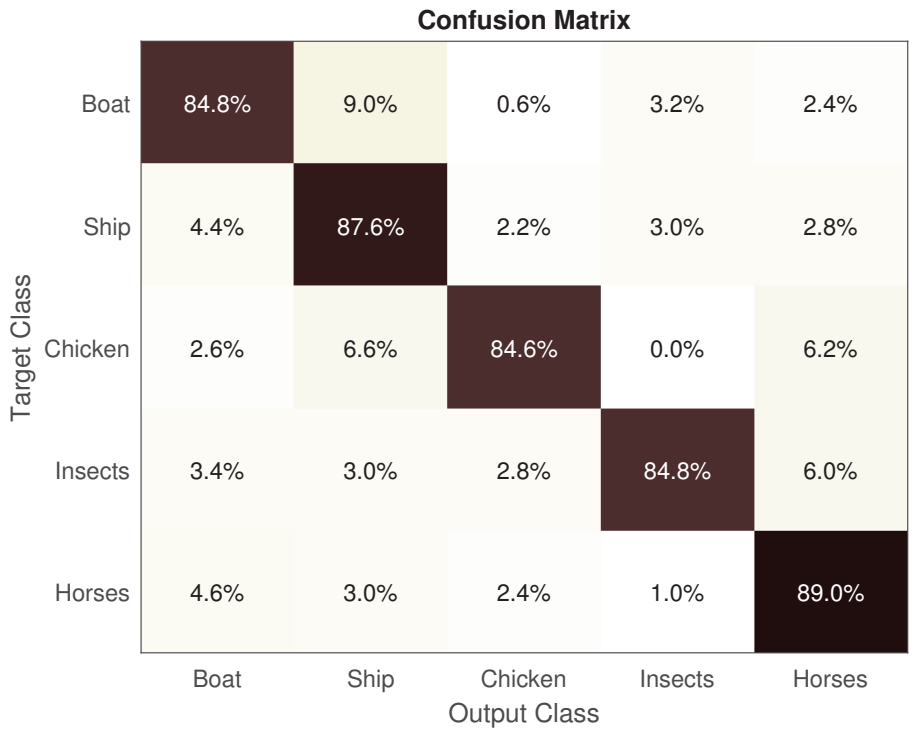


Figure 5.18: Confusion Matrix for Ghim dataset (Objects) using BAGGMM for 5 categories (subset-2)

images from 5 different categories with 400 images in each group are selected. Sample images for both experiments are shown in Figs. (5.20 & 5.22). The proposed algorithm is applied in scene categorization and results are compared with AGGMM. The results of these experiments are presented in Tables 5.9 & 5.10. Clustering performance is observed through several performance measures and it is concluded that proposed model has shown its success in clustering the visual scenes in both experiments. The confusion matrices for both experiments are shown in Figs. (5.20 & 5.22) which demonstrate the clustering performance for each cluster.

Figure 5.19: Samples of GHIM (Scenes) Dataset (Subset-1)

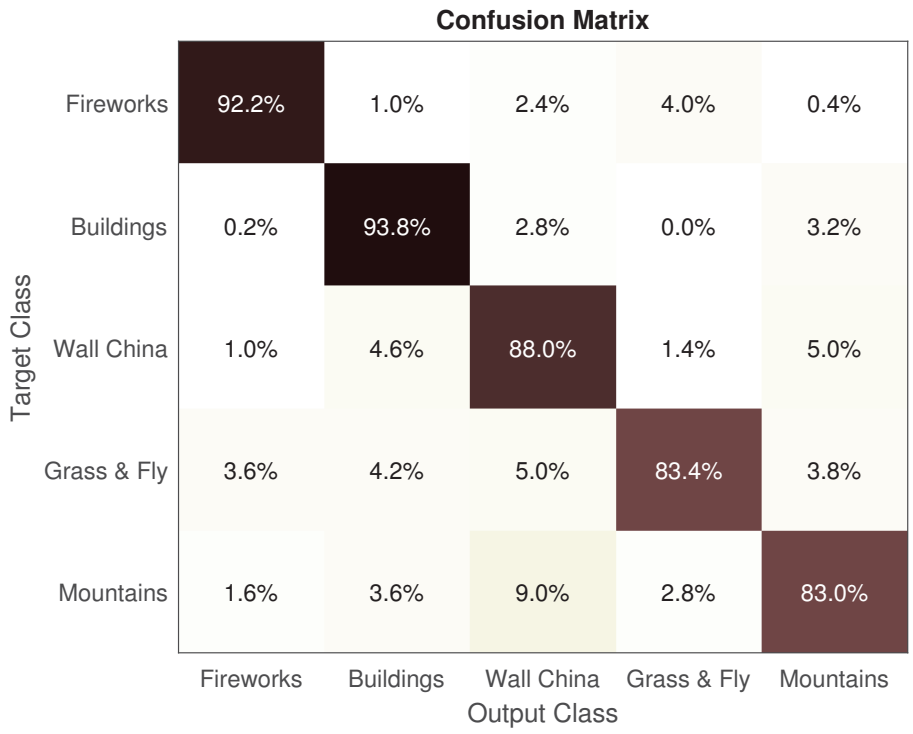


Figure 5.20: Confusion Matrix for Ghim dataset (Scene) using BAGGMM for 5 categories (subset-1)

Table 5.9: Performance of Scene Categorization for Ghim dataset (Scene) with 5 categories (subset-1)

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BAGGMM	88.08	88.08	97.02	88.23	02.98	88.06	85.15	88.15	92.44
AGGMM	85.76	85.76	96.44	86.07	03.56	85.81	82.33	85.91	90.94

Figure 5.21: Samples of GHIM (Scenes) Dataset (Subset-2)

Table 5.10: Performance of Scene Categorization for Ghim dataset (Scene) with 5 categories (subset-2)

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BAGGMM	87.24	87.24	96.81	87.28	03.19	87.24	84.06	87.26	91.90
AGGMM	85.08	85.08	96.27	85.23	03.73	85.11	81.41	85.15	90.50

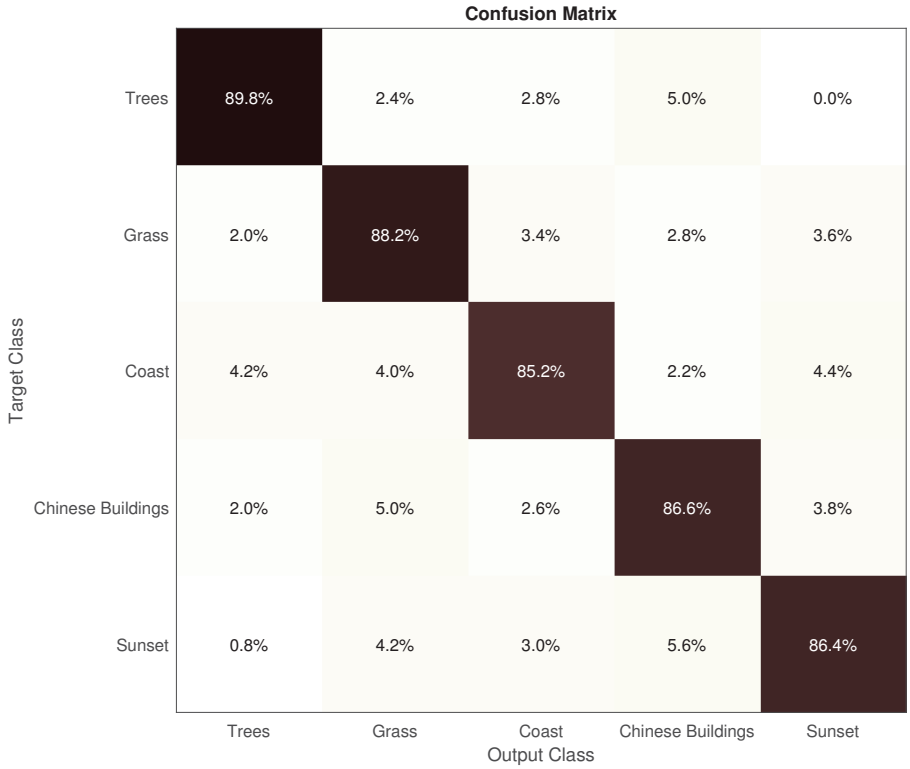


Figure 5.22: Confusion Matrix for Ghim dataset (Scene) using BAGGMM for 5 categories (subset-2)

5.9.5 Visual Scene Categorization with 15-Scene Dataset

In our next experiments for visual scene categorization using the proposed model, 15-Scenes dataset is chosen which contributed by [282, 297, 298]. Similar feature extraction pipeline as in the previous experiments was used. The model is evaluated on 4 different subsets of the 15-scenes dataset. The first two subsets consisted of visual scene images from 4 categories and the next two comprises of 5 different categories. The subset-1 consisted of images corresponding to bedroom, suburb, kitchen and office. The second subset consisted of living room, highway, inside city and street. The third subset has images corresponding to industry, coast, forest, building and store. Finally, the fourth subset has images from mountain, open country, coast, building and inside city. The samples images for each experiment are shown in Figs. (5.23,5.25,5.27 & 5.29). The experimental results for all the experiments with 15-Scene dataset are shown in Tables 5.11,5.12,5.13 & 5.14 which clearly indicates the superiority of our model when compared to AGGMM. The confusion matrices for these experiments are shown in Figs. (5.24,5.26,5.28 & 5.30) which demonstrate the performance of each cluster in the experiment.

Figure 5.23: Samples of 15-Scene Dataset (Subset-1)

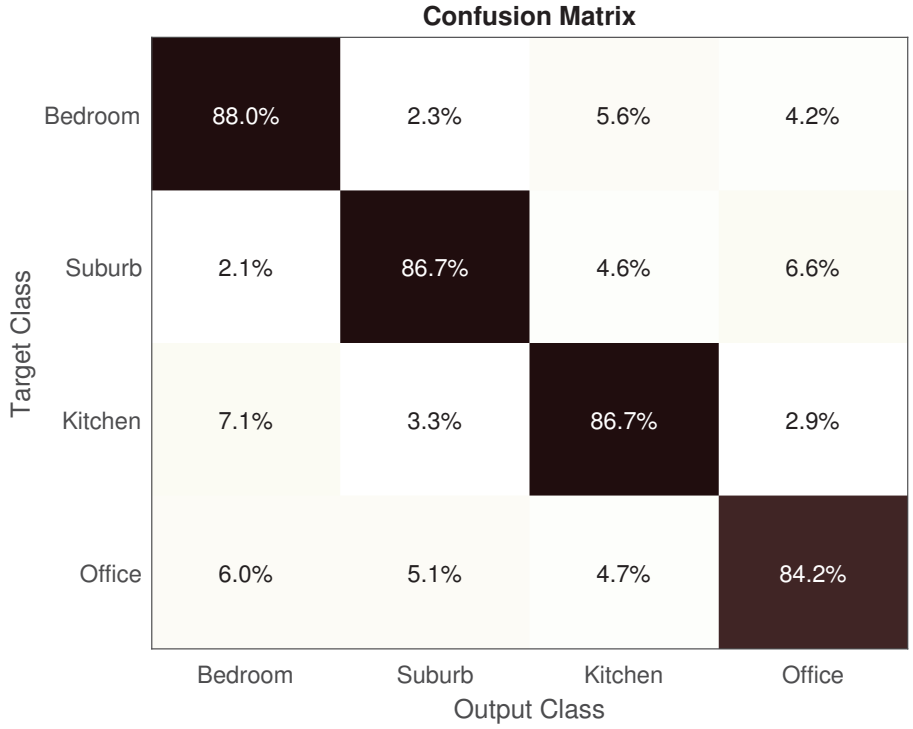


Figure 5.24: Confusion Matrix for 15-Scene dataset using BAGGMM for 4 categories (subset-1)

Table 5.11: Performance of Scene Categorization for 15 Scene dataset with 4 categories (subset-1)

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BAGGMM	86.39	86.38	95.47	86.32	04.52	86.33	81.82	86.35	90.81
AGGMM	83.78	83.72	94.60	83.75	05.40	83.73	78.33	83.74	88.99

Table 5.12: Performance of Scene Categorization for 15 Scene dataset with 4 categories (subset-2)

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BAGGMM	87.03	87.01	95.64	87.46	04.35	87.18	82.88	87.23	91.22
AGGMM	83.89	83.76	94.58	85.20	05.41	84.13	79.01	84.48	89.01

Figure 5.25: Samples of 15-Scene Dataset (Subset-2)

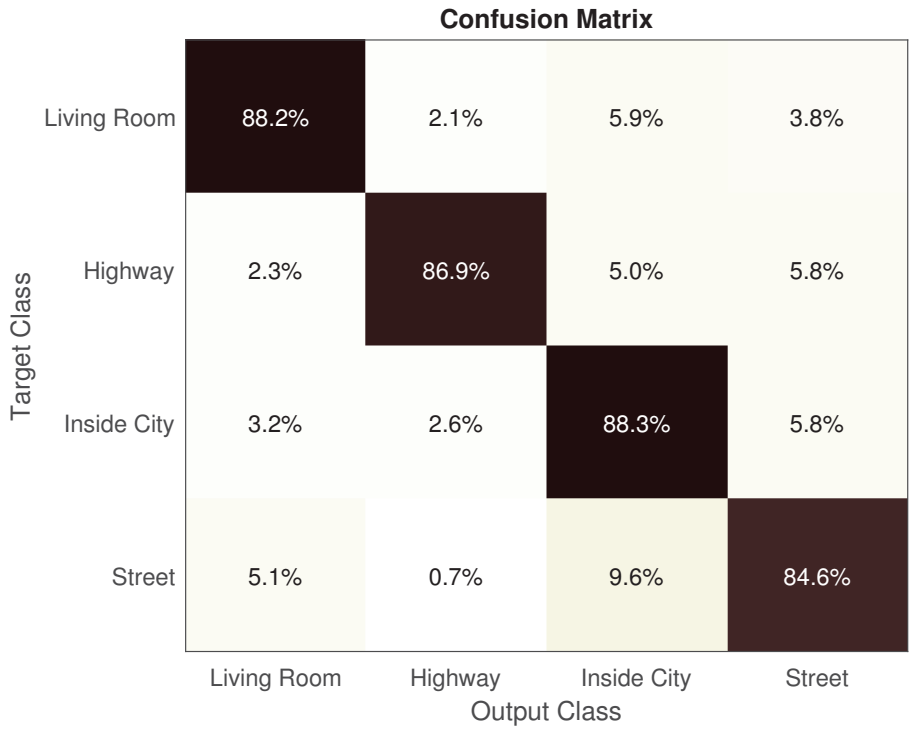


Figure 5.26: Confusion Matrix for 15-Scene dataset using BAGGMM for 4 categories (subset-2)

Figure 5.27: Samples of 15-Scene Dataset (Subset-3)

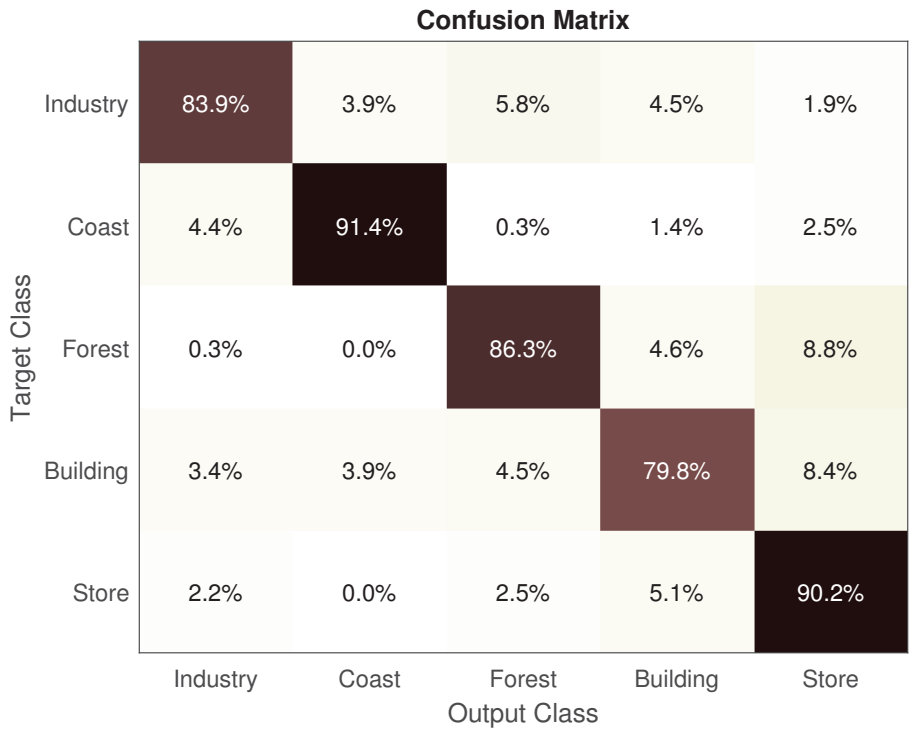


Figure 5.28: Confusion Matrix for 15-Scene dataset using BAGGMM for 5 categories (subset-3)

Table 5.13: Performance of Scene Categorization for 15 Scene dataset with 5 categories (subset-3)

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BAGGMM	86.28	86.30	96.57	86.34	3.42	86.22	82.87	86.32	91.29
AGGMM	84.61	84.36	96.16	84.54	03.83	84.41	80.60	84.45	90.06

Figure 5.29: Samples of 15-Scene Dataset (Subset-4)

5.10 Model Selection with Minimum Message Length (MML) Criterion

For estimation of number of mixture components, different model selection criteria have been proposed [30]. In this chapter, a model selection criterion based on MML is proposed for BAGGMM

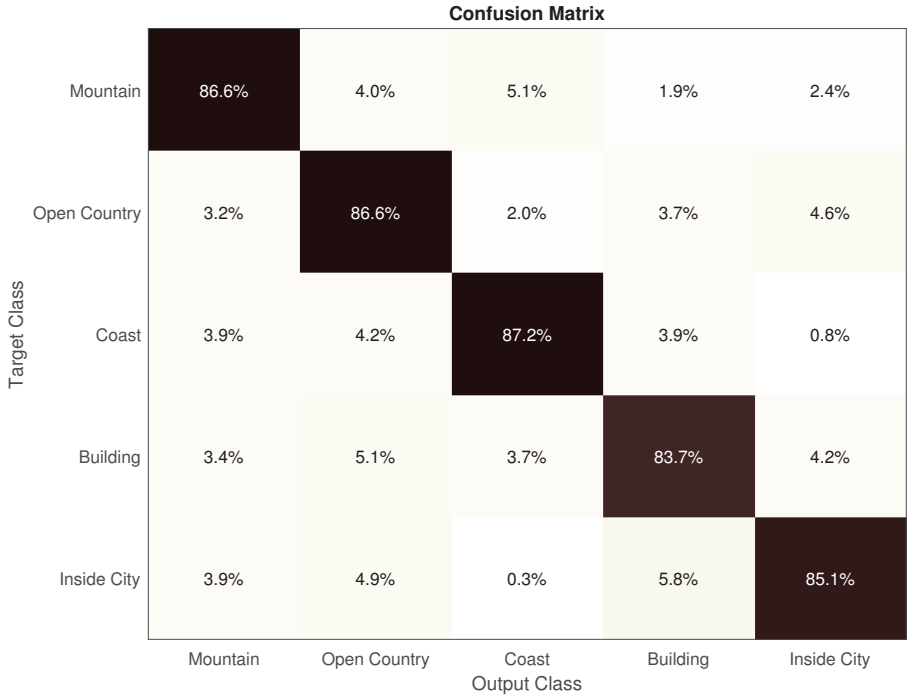


Figure 5.30: Confusion Matrix for 15-Scene dataset using BAGGMM for 5 categories (subset-4)

Table 5.14: Performance of Scene Categorization for 15 Scene dataset with 5 categories (subset-4)

Models	Performance Metrics (%)								
	Accuracy	Sensitivity	Specificity	Precision	FPR	F1-Score	MCC	G-Mean 1	G-Mean 2
BAGGMM	85.896	85.842	96.464	85.901	3.536	85.868	82.336	85.871	90.998
AGGMM	83.407	83.413	95.848	83.341	4.1516	83.355	79.218	83.377	89.415

and optimal number of mixture components can be obtained by following equation [113, 114]:

$$MessLen(K) \simeq -\log(p(\Theta_K)) - \mathscr{L}(\Theta_K, Z, \mathscr{X}) + \frac{1}{2}\log|F(\Theta_K)| + \frac{N_p}{2}(1+\log(k_{N_p})) \qquad (5.71)$$

where N_p is number of free parameters, Θ_K is set of parameters when mixture contains K components, $p(\Theta_K)$ is prior probability and $|F(\Theta_K)|$ is determinant of the Fisher information matrix minus the log-likelihood of mixture model. k_{N_p} is optimal quantization lattice constant $\mathbb{R}^{N_p}$ [115] and its written as $k_1 = 1/12 \simeq 0.83$ for $N_p = 1$. As N_p grows, k_{N_p} will become an asymptotic value as $1/2\pi e \simeq 0.05855$ and it is noted that k_{N_p} does not vary a lot and it can be approximated by $1/12$ [116]. The estimation of the number of classes is carried out by finding the minimum with respect to Θ of the message length [32, 47, 58, 116]. The derivation of $p(\Theta_K)$ and $|F(\Theta_K)|$ is given as follows:

5.10.1 Derivation of the prior $p(\Theta)$

In the model selection, a prior $p(\Theta)$ is defined to expresses the lack of knowledge about the parameters of mixture model. It is assumed that different mixture components have independent parameters, since having information about the parameters in one class does not provide any information about the parameters of another class. Thus, it is assumed that parameters of a mixture model are mutually independent, which cede the following prior distribution over the parameters π, μ, σ_l, σ_r and λ:

$$p(\Theta) = p(\pi)p(\mu)p(\sigma_{l_{jd}})p(\sigma_{r_{jd}})p(\lambda_{jd}) \tag{5.72}$$

where $\pi = (p_1, ..., p_K)$. Each of these densities in the prior distribution are defined separately. Beginning with $p(\pi)$, we know that vector π is defined on the simplex as $\{(p_1, ..., p_K) : \sum_{j=1}^{K} p_j = 1\}$. In this case, a natural choice as a prior for vector π is Dirichlet distribution, which is defined as:

$$p(\pi) = \frac{\Gamma(\sum_{j=1}^{K} \eta_j)}{\sum_{j=1}^{K} \Gamma(\eta_j)} \sum_{j=1}^{K} {p_j}^{\eta_j - 1} \tag{5.73}$$

where $(\eta_1, ..., \eta_K)$ is the parameters vector of Dirichlet distribution. By choosing, $\eta_1 = 1, ..., \eta_K = 1$, we get a uniform prior over the space $p_1 + ... + p_K = 1$, which is represented as:

$$p(\pi) = (K-1)! \tag{5.74}$$

For the parameter σ_l and σ_r, we have:

$$p(\sigma_l) = \prod_{j=1}^{K} p(\vec{\sigma}_{l_j}) \tag{5.75}$$

$$p(\sigma_r) = \prod_{j=1}^{K} p(\vec{\sigma}_{r_j}) \tag{5.76}$$

where different components of vector $\vec{\sigma}_{l_j}$ and $\vec{\sigma}_{r_j}$ are assumed to be independent. The principle of ignorance is adopted due to the absence of other knowledge about $\sigma_{l_{jd}}$ and $\sigma_{r_{jd}}$ with $d = 1, ..., D$, by taking a uniform prior. If $\vec{\mu} = (\mu_1, ..., \mu_D)$, $\vec{\sigma}_l = (\sigma_{l_1}, ..., \sigma_{l_D})$ and $\vec{\sigma}_r = (\sigma_{r_1}, ..., \sigma_{r_D})$ are mean, left standard deviation and right standard deviation vectors of whole dataset, then for each σ_{jjd} and σ_{rjd}, following uniform prior will be used:

$$p(\sigma_{l_{jd}}) = \frac{1}{\sigma_{l_j}} \tag{5.77}$$

$$p(\sigma_{r_{jd}}) = \frac{1}{\sigma_{r_j}} \tag{5.78}$$

where $0 \leq \sigma_{l_{jd}} \leq \sigma_{l_d}$, and $0 \leq \sigma_{r_{jd}} \leq \sigma_{r_d}$, $d = 1, ..., D$. It follows that

$$p(\vec{\sigma}_{l_j}) = \prod_{d=1}^{D} \frac{1}{\sigma_{l_d}} \tag{5.79}$$

$$p(\vec{\sigma}_{r_j}) = \prod_{d=1}^{D} \frac{1}{\sigma_{r_d}} \tag{5.80}$$

From Eqs. (5.79 & 5.80), we obtain:

$$p(\sigma_l) = \prod_{j}^{K} \prod_{d=1}^{D} \frac{1}{\sigma_{l_d}} = \prod_{d=1}^{D} \frac{1}{{\sigma_{l_d}}^K} \tag{5.81}$$

$$p(\sigma_r) = \prod_{j}^{K} \prod_{d=1}^{D} \frac{1}{\sigma_{r_d}} = \prod_{d=1}^{D} \frac{1}{{\sigma_{r_d}}^K} \tag{5.82}$$

For each μ_{jd}, uniform prior is chosen, similarly as standard deviation. Each μ_{jd} is chosen to be uniform in the region $(\mu_d - \sigma_{l_d} \leq \mu_{jd} \leq \mu_d + \sigma_{r_d})$, then prior for μ_j is given by the following equations:

$$p(\mu_{jd}) = \frac{1}{\sigma_{l_d} + \sigma_{r_d}} \tag{5.83}$$

$$p(\vec{\mu}_j) = \prod_{d=1}^{D} \frac{1}{\sigma_{l_d} + \sigma_{r_d}} \tag{5.84}$$

$$p(\mu) = \prod_{j=1}^{K} \prod_{d=1}^{D} \frac{1}{\sigma_{l_d} + \sigma_{r_d}} = \prod_{d=1}^{D} \frac{1}{(\sigma_{l_d} + \sigma_{r_d})^K} \tag{5.85}$$

For the prior of shape parameter, a uniform distribution $\mathscr{U}[0, h]$, where h is chosen to be sufficiently large and prior is as follows:

$$p(\lambda_{jd}) = \frac{1}{h} \tag{5.86}$$

$$p(\vec{\lambda}_j) = \prod_{d=1}^{D} \frac{1}{h} = \frac{1}{h^D} \tag{5.87}$$

$$p(\lambda) = \prod_{j}^{K} \prod_{d=1}^{D} \frac{1}{h} = \frac{1}{h^{D.K}} \tag{5.88}$$

Finally, by replacing the priors of parameters in Eq. (5.72) by Eqs. (5.74, 5.79 & 5.85), we get:

$$p(\Theta) = \frac{(K-1)!}{h^{KD}} \prod_{d=1}^{D} \frac{1}{\sigma_{l_d}^K \sigma_{r_d}^K (\sigma_{l_d} + \sigma_{r_d})^K} \tag{5.89}$$

5.10.2 Derivation of the Fisher information matrix $|F(\Theta)|$

Fisher information matrix is expected value of the Hessian matrix. It is difficult to reproduce the expected Fisher Information matrix because it leads to a complicated analytical form of MML. Therefore, Hessian matrix can be approximated by complete Fisher information matrix as follows:

$$|F(\Theta)| = |F(\pi)| \prod_{j=1}^{K} |F(\vec{\mu}_j)||F(\vec{\lambda}_j)||F(\vec{\sigma}_{lj})||F(\vec{\sigma}_{rj})| \tag{5.90}$$

$$|F(\pi)| = \frac{N^{K-1}}{\sum_{j=1}^{K} p_j} \tag{5.91}$$

$$F(\vec{\mu}_j)_{k_1,k_2} = \frac{\partial^2 \mathscr{L}(\Theta, Z, \mathscr{X}_j)}{\partial \mu_{jd_1} \partial \mu_{jd_2}} \tag{5.92}$$

$$F(\vec{\sigma}_{l_j})_{k_1,k_2} = \frac{\partial^2 \mathscr{L}(\Theta, Z, \mathscr{X}_j)}{\partial \sigma_{l_{jd_1}} \partial \sigma_{l_{jd_2}}} \tag{5.93}$$

$$F(\vec{\sigma}_{r_j})_{k_1,k_2} = \frac{\partial^2 \mathscr{L}(\Theta, Z, \mathscr{X}_j)}{\partial \sigma_{r_{jd_1}} \partial \sigma_{r_{jd_2}}} \tag{5.94}$$

$$F(\vec{\lambda}_j)_{k_1,k_2} = \frac{\partial^2 \mathscr{L}(\Theta, Z, \mathscr{X}_j)}{\partial \lambda_{jd_1} \partial \lambda_{jd_2}} \tag{5.95}$$

where $(d_1, d_2) \in (1, ..., D)$. By the using the second order derivatives computed in Section 5.8 for all the parameter of distribution and using the following equation, determinant of Fisher Information for each component can be computed.

$$\frac{\partial^2 \mathscr{L}(\Theta, Z, \mathscr{X}_j)}{\partial \xi_{jd_1} \partial \xi_{jd_2}} = 0 \tag{5.96}$$

Model selection algorithm also serves as a complete clustering solution because it provides the optimal number of mixture components which helps to estimate the optimal parameters learned through EM. The complete learning algorithm is given in Algorithm 7.

Algorithm 7 Complete Model Learning with BAGGMM and Model Selection using MML

Input:Dataset $\mathscr{X} = \{\vec{X}_1, \ldots, \vec{X}_N\}$, t_{min} and K_{max}.
Output: K^* and Θ_{K^*}.
Step 1: for $M = 1 : K_{max}$ **do**{
{Initialization}:
  K-Means Algorithm (Computation of $\vec{\mu}_1, \ldots, \vec{\mu}_K$ & cluster assignment)
    for all $1 \leq j \leq K$ do
      Computation of p_j
      Computation of $\{(\vec{\sigma}_{l_j}$ & $\vec{\sigma}_{r_j})$=$\vec{\sigma}_j\}$
      Set the $\{(\vec{\lambda}_j = 2\}$
    end for
{Expectation Maximization}:
while relative change in log-likelihood $\geq t_{min}$ do
  $\{$[**E Step**]$\}$:
    for all $1 \leq j \leq K$ do
      Compute $p(j|\vec{X}_i)$ for $i = 1, \ldots, N$. using Eq. (5.43).
    end for
  $\{$[**M step**]$\}$:
    for all $1 \leq j \leq K$ do
      Estimation of mixing parameter p_j using Eq. (2.12).
      Estimation of mean $\vec{\mu}_j$ using Eq. (5.52).
      Estimation of left standard deviation $\vec{\sigma}_{l_j}$ using Eq. (5.55).
      Estimation of right standard deviation $\vec{\sigma}_{r_j}$ using Eq. (5.60).
      Estimation of shape parameter $\vec{\lambda}_j$ using Eq. (5.70).
    end for
end while
Calculate the associated message length using Eq. (5.71).
}**end for**
Step 2: Select the Model K^* with smallest message length

5.11 Experiments on model selection and results

In order the evaluate the performance of proposed model selection criterion using MML we consider different clustering applications. In the experiments mentioned in Section 5.9, several applications and datasets are mentioned and model selection is applied in all scenarios. Model selection improves the performance of clustering by proving the information about optimal number of clusters in the data. In the experiments for model selection, several clustering scenarios using MML are created to see the effectiveness of proposed approach. Details about model selection criteria used in our experiments is given in Section 5.11.1. The experiments on model selection with spam detection, object recognition and visual scene categorization is provided in Section 5.11.2-5.11.5.

5.11.1 Comparison with other model selection criteria

In clustering with BAGGMM, proposed model selection with MML approach is compared with different deterministic model selection criteria given in literature. The comparison models for finding the optimal number of clusters include MDL [74], AIC [119], Bayesian inference criterion (BIC) [73], Consistent AIC (CAIC) [120], Mixture MDL (MMDL) [121], MML_{like} [30], LEC [16, 39]. Any deterministic model selection criteria can be expressed to form a general notation as follows:

$$C(\hat{\Theta}(K),K) = -\mathscr{L}(\Theta_K,Z,\mathscr{X}) + f(K) \tag{5.97}$$

where $\mathscr{L}(\Theta_K,Z,\mathscr{X})$ is complete log-likelihood of data and $f(K)$ is called an increasing function which penalize higher values of K and number of optimal mixture components can be computed as follows:

$$\hat{K} = arg\ min\{C(\hat{\Theta}(K),K), K = K_{min},...,K_{max}\} \tag{5.98}$$

Although model selection criteria has this common point, but conceptually they are different and they described by the following equations:

$$MDL(K) = -\mathscr{L}(\Theta_K,Z,\mathscr{X}) + \frac{N_p}{2}\log(N) \tag{5.99}$$

where N_p is number of free parameters estimated for mixture and computed as $K(2D+1)$ in our case.

$$AIC(K) = -\mathscr{L}(\Theta_K,Z,\mathscr{X}) + \frac{N_p}{2} \tag{5.100}$$

$$BIC(K) = -2\mathscr{L}(\Theta_K,Z,\mathscr{X}) + N_p\log(N) \tag{5.101}$$

$$CAIC(K) = -2\mathscr{L}(\Theta_K,Z,\mathscr{X}) + N_p(1+\log(N)) \tag{5.102}$$

$$MMDL(K) = -\mathscr{L}(\Theta_K,Z,\mathscr{X}) + \frac{1}{2}N_p\log(N) + \frac{c}{2}\sum_{j=1}^{K}\log(p_j) \tag{5.103}$$

where c is the number of free parameters for each mixture component and computed as $(2D+1)$ in our case.

$$MML_{Like}(K) = -\mathscr{L}(\Theta_K,Z,\mathscr{X}) + \frac{K}{2}\log\left(\frac{N}{12}\right) + \frac{c}{2}\sum_{j=1}^{K}\log\left(N\frac{p_j}{12}\right) + \frac{N_p}{2} \tag{5.104}$$

For computation of number of mixture components through LEC, prior probability and determinant of Fisher information matrix computed for MML is used in the following equation.

$$LEC(K) = \mathscr{L}(\Theta_K, Z, \mathscr{X}) - \log(P(\Theta_K)) - \frac{1}{2} N_p \log(2\pi) + \frac{1}{2} \log(|F(\Theta_K)|) \tag{5.105}$$

5.11.2 Model Selection on Spam Hunter Dataset

In the Spam Hunter dataset, there are two categories of images and it is selected as first step to examine the performance of MML criterion for model selection. MML is applied to find the optimal number of clusters in the dataset and it is compared with different model slection criteria mentioned in Section 5.11.1. From the results of experiment, it is observed that MML, MDL, MMDL, MML$_{Like}$ and LEC have successfully determined the number of clusters in the dataset and AIC, BIC and CAIC were unable in the test for model selection. The results on Spam Hunter dataset are provided in Table 5.15 and plotted in Fig. (5.31) which demonstrate the performance of MML with regard to other criteria.

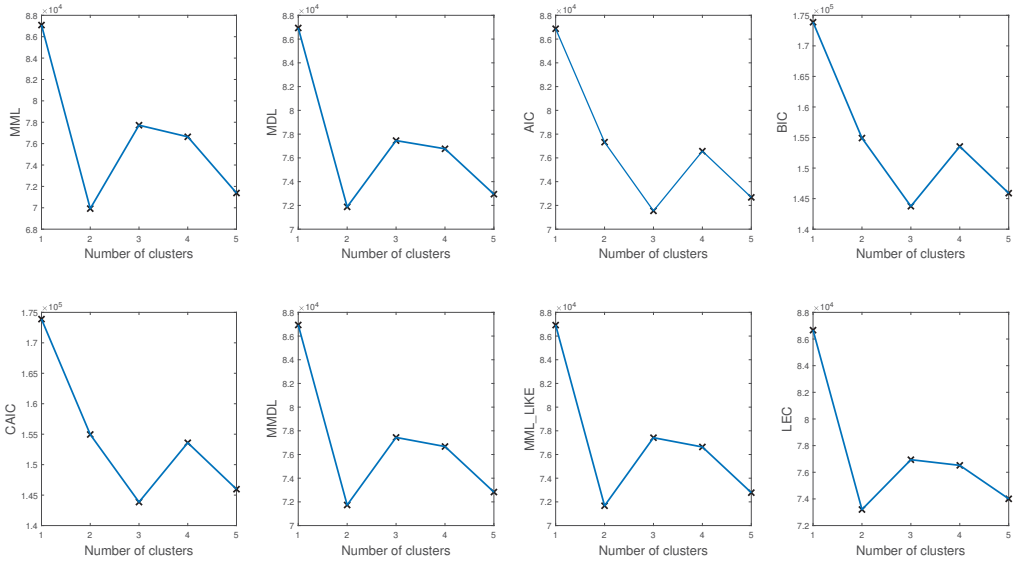


Figure 5.31: Model Selection Criteria for Spam Hunter Dataset with 2 clusters

5.11.3 Model Selection on Object Recognition with ETHZ Dataset

The experiments of model selection are further extended with ETHZ dataset where 5 different categories of object images are present and it is selected to test our algorithm for model selection. From the results of experiment for finding the optimal number of mixture components, it is

observed that MML and LEC have successfully computed the number of mixture components in ETHZ dataset and rest of model selection criteria have failed in this test. The results for this test are provided in Table 5.15 and plotted in Fig. (5.11.3).

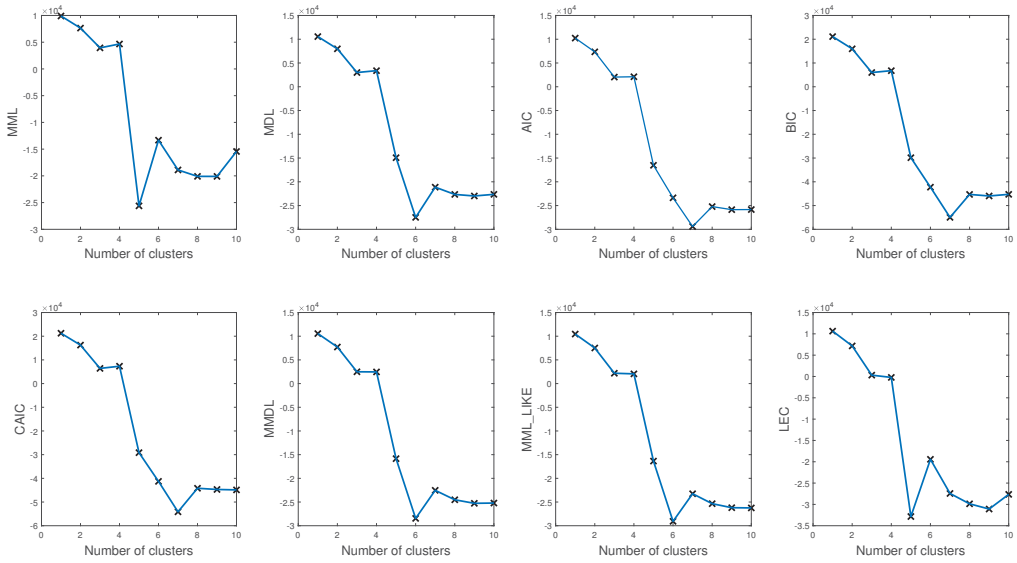


Figure 5.32: Model Selection Criteria for ETHZ Dataset with 5 clusters

5.11.4 Model Selection on Object Recognition with GHIM Dataset

In order to examine the effectiveness of model selection criteria, it is applied to GHIM dataset with object classes and several clustering scenarios are created with 2, 3, 4 and 5 clusters with 800, 1200, 1600 and 2000 images. MML is applied to the clustering framework for model selection and it is observed that for two cluster experiment, MML, MDL, MMDL, MML_{Like} and LEC have successfully determined the number of clusters in the dataset. For the experiments with data from 3, 4 and 5 clusters, MML and LEC have shown their success in finding the optimal number of components and rest of the model selection algorithm were unable to correctly determine the number of clusters from data. The experimental results are provided in Table 5.15 and plotted in Fig. (5.33-5.36), which provide a complete analysis of model selection on objection recognition with GHIM dataset.

5.11.5 Model Selection on Visual Scenes Categorization with 15-Scenes Dataset

The proposed model selection criterion (MML) is also validated through visual scene categorization. The experiments for fining the optimal number of mixture components are conducted on

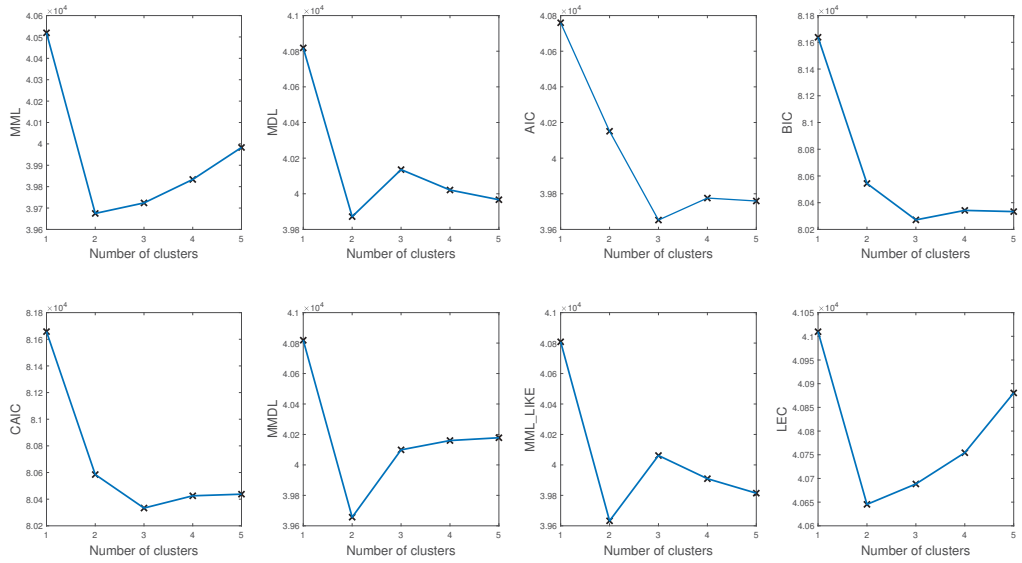


Figure 5.33: Model Selection Criteria for GHIM (Objects) Dataset with 2 clusters

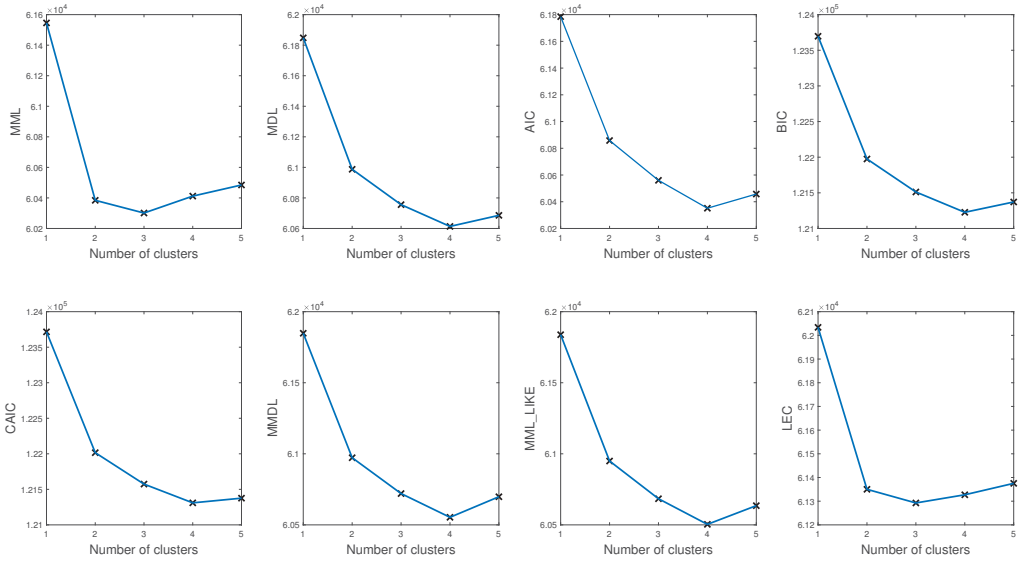


Figure 5.34: Model Selection Criteria for GHIM (Objects) Dataset with 3 clusters

15-Scene dataset by selecting 2, 3, 4 and 5 clusters with 784, 1144, 1500 and 1808 images of visual scenes, respectively. For the experiments with 2 and 3 clusters, it is observed that MML, MDL, MMDL, MML$_{Like}$ and LEC have demonstrated their success in model selection. In the experiment with 4 clusters, MML and LEC were successful in finding the correct number of mixture components. With data from 5 clusters, MML, MDL, MMDL, MML$_{Like}$ and LEC have shown

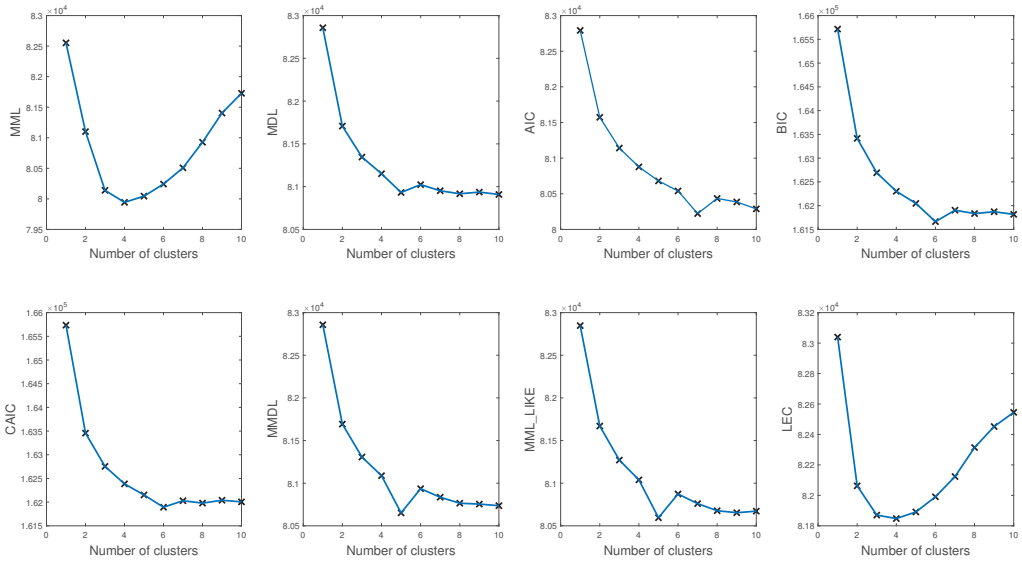


Figure 5.35: Model Selection Criteria for GHIM (Objects) Dataset with 4 clusters

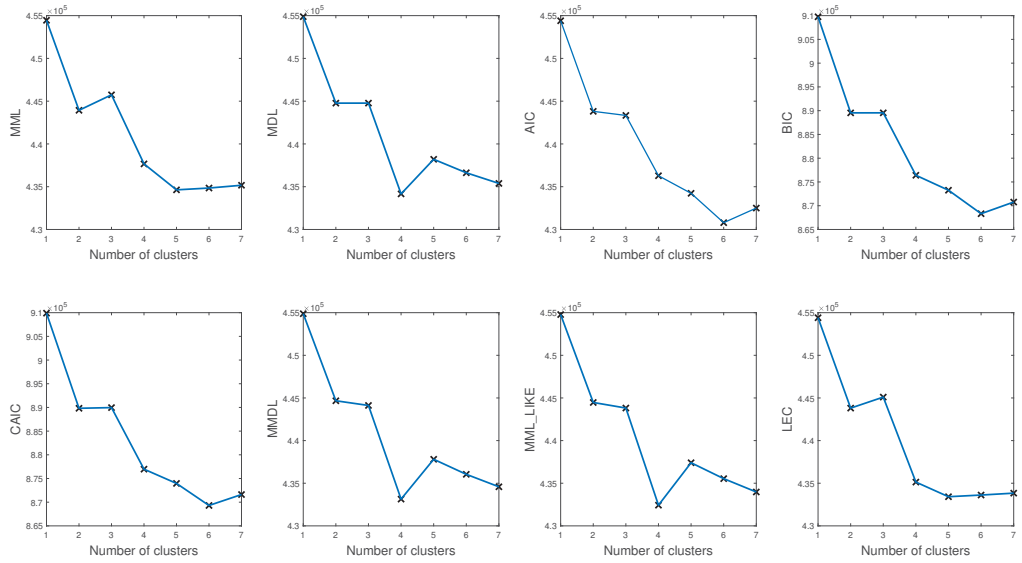


Figure 5.36: Model Selection Criteria for GHIM (Objects) Dataset with 5 clusters

their success in model selection and rest of the criteria were unable in the test. The results of the experiments of model selection with visual scenes are provided in Table 5.15 and plotted in Figs. (5.37-5.40) which clearly demonstrate the effectiveness of proposed model selection criteria for visual scene categorization.

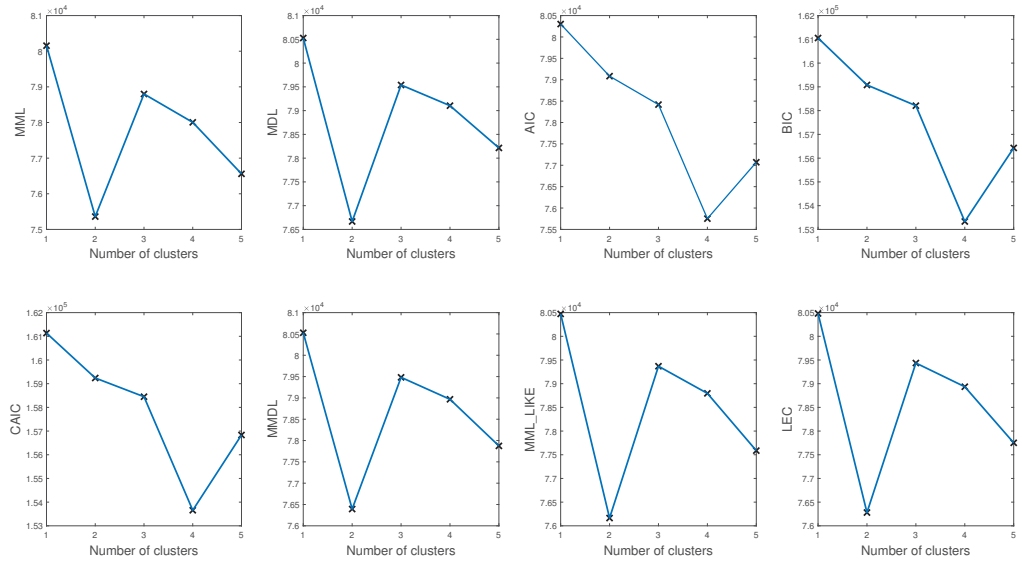


Figure 5.37: Model Selection Criteria for 15 Scene Dataset with 2 clusters

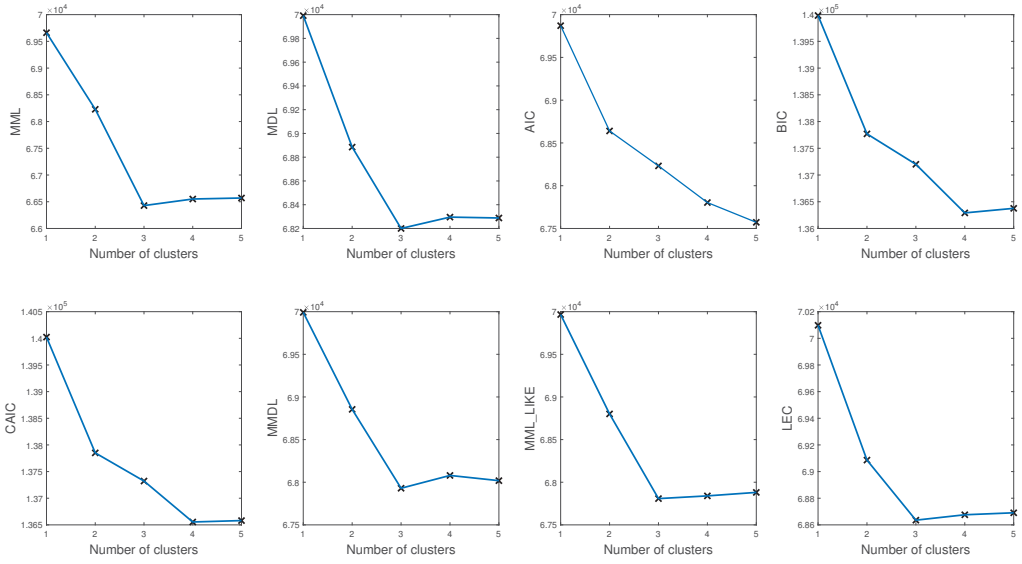


Figure 5.38: Model Selection Criteria for 15 Scene Dataset with 3 clusters

5.12 Discussion about BAGGMM and MML

In this chapter, a mixture of bounded support asymmetric generalized Gaussian distribution is proposed. Bounded support mixture is introduced by considering the fact that data in many real applications exist in a bounded support range whereas distributions to model the data are defined

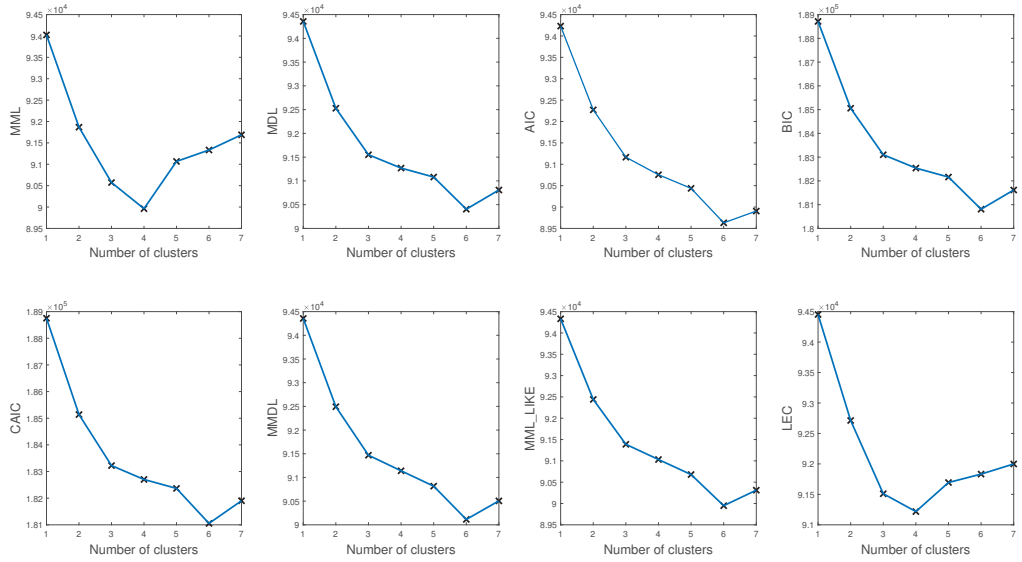


Figure 5.39: Model Selection Criteria for 15 Scene Dataset with 4 clusters

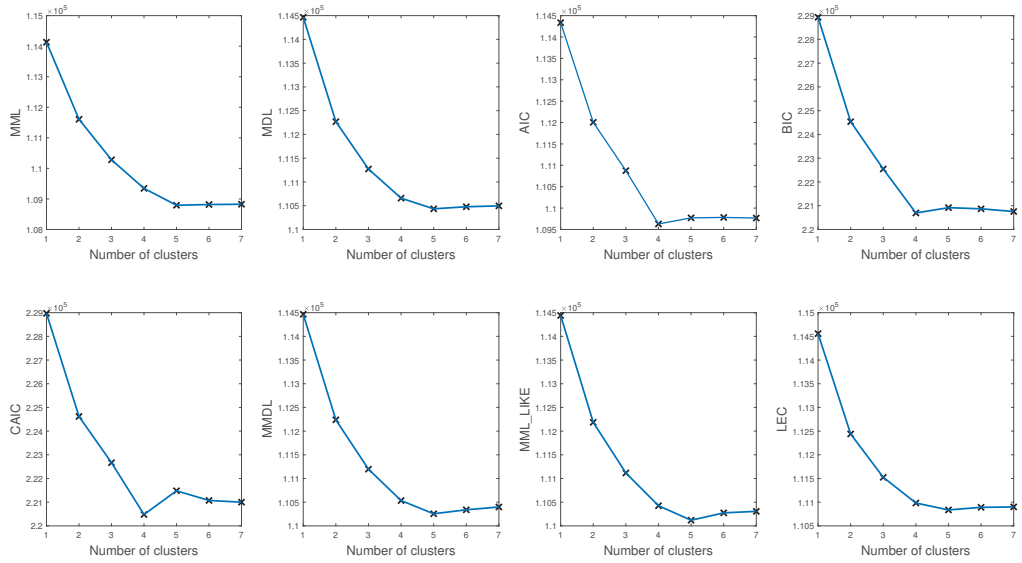


Figure 5.40: Model Selection Criteria for 15 Scene Dataset with 5 clusters

for unbounded support. By modeling the data with bounded support distribution can improve the learning process effectively. In the proposed model, parameter estimation is performed by maximum likelihood approach with EM algorithm. Some of mixture parameters does not get a closed form solution in the maximum likelihood and Newtons Raphson method is applied with EM algorithm for parameter estimation. To validate the performance of BAGGMM in clustering, several

Table 5.15: Number of Clusters Determined by Different Criteria using BAGGMM for Image Datasets used in clustering applications

Data set	D	N	$\hat{K^*}$	Model Selection Criteria							
				MML	MDL	AIC	BIC	CAIC	MMDL	MML_Like	LEC
Spam Hunter	50	1738	2	**2**	**2**	3	3	3	**2**	**2**	**2**
ETHZ	50	255	5	**5**	6	7	7	7	6	6	**5**
GHIM (Objects) 2	50	800	2	**2**	**2**	3	3	3	**2**	**2**	**2**
GHIM (Objects) 3	50	1200	3	**3**	4	4	4	4	4	4	**3**
GHIM (Objects) 4	50	1600	4	**4**	5	7	6	6	5	5	**4**
GHIM (Objects) 5	50	2000	5	**5**	4	6	6	6	4	4	**5**
15 Scene 2	50	784	2	**2**	**2**	4	4	4	**2**	**2**	**2**
15 Scene 3	50	1144	3	**3**	**3**	5	4	4	**3**	**3**	**3**
15 Scene 4	50	1500	4	**4**	6	6	6	6	6	6	**4**
15 Scene 5	50	1808	5	**5**	**5**	4	4	4	**5**	**5**	**5**

applications with images data are proposed. Initially, Spam Hunter dataset is used for spam detection and the proposed model effectively categorize the images into Spam and Ham. In this application, we have received a very high accuracy (96.37%) and very low false positive rate (00.64%). The results are compared with AGGMM in a similar experimental settings and proposed model has demonstrated its effectiveness in spam detection. BAGGMM is applied in object recognition and ETHZ and GHIM datasets are selected for in this application. GHIM dataset is composed of objects and visual scenes and we have used objects parts of dataset in this application. In object clustering one experiment on ETHZ with 5 clusters and two separate experiments with GHIM dataset with 5 clusters each, are performed. In all three experiments for object recognition, BAGGMM has performed extremely well as compared to AGGMM in a similar experimental setting and it is demonstrated through different performance measures. Our next experiments for image clustering are performed on visual scene categorization and we have selected GHIM dataset (part of visual scenes) and 15-Scene dataset. With GHIM dataset, two separate clustering scenarios are created with 5 clusters in each test and it is observed that BAGGMM has shown better performance in this task as compared to AGGMM. In our experiments with 15-Scene dataset, we have created 4 clustering scenarios (two with 4 clusters and two with 5 clusters) and it is observed that proposed algorithm has significantly improvement in clustering. Our next contribution in this chapter is model selection criteria for BAGGMM and MML is proposed for determining the optimal number of clusters using BAGGMM. The proposed model is applied to all the dataset in different clustering applications chosen to demonstrate the effectiveness of BAGGMM. From the set of experiments, it is observed that our models have shown extremely better performance in spam detection, object recognition, visual scene categorization and finding the optimal number of clusters.

Chapter 6

Conclusions

Clustering is the task of unsupervised categorization of observations or patterns into groups or clusters. Clustering algorithms aim to categorize elements of data into clusters or groups based on their similarity. Generally, the task of clustering faces several challenges and complexities in multidimensional feature space, which include unknown and unidentified shape of data, unknown number of clusters in data and existence of noise and redundant information in features of data which affect the modeling capabilities and performance of clustering task. Therefore, data clustering with high dimensional feature space has been considered an active area of research different fields which include information retrieval, data mining, pattern recognition, speech and image processing and many areas in last few decades. Many techniques in data clustering involves probability density function as a way to model the data and it has shown great success in clustering for many applications and different types of data including speech, images, videos and text. However, choosing an appropriate distribution to model the data in a particular application is a challenging task because most of the times, shape of data is unknown which also compromise the modeling capability in clustering. It is also observed that many real life applications have their data in a bounded support range, however, existing distributions to model this data are available for unbounded support range. In this thesis, special attention has been given to increase the modeling capabilities of finite mixture model by adopting bounded support distributions and applying them in different kinds of applications to model the data and improve the clustering performance, improve the feature extraction techniques which uses clustering and finding the optimal number of clusters.

In this thesis, first we proposed to adopt bounded Gaussian mixture model for data clustering to examine and analyze its effectiveness in different kinds of applications. Parameters of mixture model were estimated by maximum likelihood and learned by adopting EM algorithm. Initial experiments involve clustering speech and images datasets, which include categorizing between female and male speaker, recognizing spoken and handwritten digits in separate applications

and clustering different categories of fashion data in MNIST fashion dataset. The clustering was performed with data from different clusters in each application to examine the performance by varying the categories of data. These experiments were extended further by applying the bounded Gaussian mixture in feature representation for speech and images datasets. In these applications, speech and images features (MFCCs and SIFT, respectively) were used to create a BoW and mixture model is proposed to improve the process of code-book generation, to better represent the data and further improve the learning of clustering task. In the speech datasets, bag of audio words (BoAW) approach is used which is inspired by BoW and BoVW approaches from NLP and computer vision, respectively. In a clustering task, finding the optimal number of clusters is very crucial and minimum message length (MML) is proposed for bounded Gaussian mixture. The proposed model selection criterion is validated through 10 medical datasets and all above mentioned datasets from speech and images with different number of cluster and compared with 7 different criteria to examine it effectiveness. From the experiments on clustering, code-book generation and model selection, it is observed that proposed model has demonstrated its effectiveness as compared to models with unbounded support.

Second, we have extended the idea of bounded support to Laplace distribution and proposed a mixture of bounded Laplace distributions and parameters are estimated with maximum likelihood and EM algorithm along with Newton-Raphson method. The proposed model is validated for clustering initially through synthetic data (one dimensional and multidimensional datasets) with different numbers of clusters and 10 medical datasets. It is further applied in more advanced applications to categorize texture images and content based image retrieval (CBIR). Bounded support mixture is proposed for feature extraction from texture images in wavelet domain due to energy compacting property of wavelet transform, which makes the distribution of data very similar to a Laplace density. In this application, BLMM is also used in modeling the data after feature extraction which perform the task of image categorization in an unsupervised manner. The trained model from the clustering stage is further adopted for CBIR, where a similarity measure can be applied to compare the query image and clusters of trained model. For similarity measure, City-block distance, posterior probability and Kullback-Leibler (KL) divergence are introduced and a closed form solution for KL divergence is also proposed for Laplace distributions in the context of our clustering model. The experiments are conducted on 5 different datasets to demonstrate the effectiveness of proposed model in feature extraction, image categorization and CBIR. From the experiments on synthetic data, medical datasets, feature extraction, texture image categorization and CBIR, it is evident that proposed bounded support mixture has demonstrated its effectiveness in learning as compared to unbounded support models. With feature extraction using BLMM in wavelet domain, image categorization is also proposed using supervised learning approach and a

Naive Bayes classifier is introduced with Laplace distribution. The proposed model is validated through experiments on 3 texture images datasets and compared with Gaussian Naive Bayes classifier. These experiments also demonstrate the effectiveness of feature extraction using BLMM. The generalized Gaussian distribution provides a generalization of Gaussian and Laplace distributions, thus we have also introduced a Naive Bayes classifier with generalized Gaussian distribution which is validated through experiments conducted on 3 datasets from texture images. Classification performance demonstrate the effectiveness of proposed approach and feature extraction using BLMM in wavelet domain.

Third, the idea of bounded support mixtures is extended to ICA and a multivariate bounded generalized Gaussian mixture model with ICA was proposed. In an ICA mixture, it is assumed that observed data come from a mixture model and it can be categorized into mutually exclusive classes which mean that each class of data will be modeled through an ICA. The proposed ICA mixture model was applied to unsupervised keyword spotting and TIMIT speech corpus was used to create the experimental framework and compared with GMM. In the keyword spotting, ICA mixture model is first trained on a large amount of speech data and trained model is further used to generate posteriorgrams for reference keyword examples and test speech file. The posteriorgrams are compared with segmental dynamic time warping to find a match between test speech files and reference keywords. Proposed ICA mixture model is further explored with application to speaker classification in a semi-supervised hierarchical clustering framework. The experiments are conducted using TIMIT and TSP speech databases for male and female speaker categorization (TSP and TIMIT) and 10 speaker categorization (TSP). ICA mixture model is further explored with blind source separation on speech data and it is validated by conducting experiments on 3 speech datasets. In the last application with ICA mixture, BSS is applied as pre-processing stage for test data in unsupervised keyword spotting, where experimental framework of first application is extended to see the effectiveness of BSS in keyword spotting when test data has been mixed with noise and speech files. The experiments are conducted on data with source mixing where keyword spotting is performed with and without BSS as pre-processing. From all experiments in 4 different applications and many datasets, it is observed that ICA mixture model has demonstrated its success in modeling and learning in speech data applications.

Last but not least, two bounded support mixture models were proposed which are based on asymmetric distributions. The first proposed model is bounded asymmetric Gaussian mixture and second is bounded support asymmetric generalized Guassian mixture. In both models, maximum likelihood and EM along with Newton Raphson method are used for parameter estimation. The first model is applied to textual spam detection (spambase dataset), object categorization (Coral and Cal 101 datasets) and texture image clustering (VisTex dataset). From the experiments and results,

it is observed that BAGMM has improved data modeling capability as compared to AGMM. For the experiments on BAGGMM, image spam detection (Spam Hunter dataset), object clustering (ETHZ and GHIM datasets) and visual scene categorization (GHIM and 15-Scene datasets) are chosen to validate its effectiveness. The experiments are conducted on multi-cluster scenarios to examine the performance of proposed approach. For BAGGMM, a model selection criterion is also proposed with MML and experiments for its validity are conducted by choosing data from all applications mentioned in the clustering experiments for BAGGMM and compared with 7 different criteria to see its effectiveness. From the experiments on BAGGMM and MML, it is examined that proposed model has demonstrated its effectiveness.

In conclusion, proposed bounded support mixture can effectively be used as an alternative for data clustering with existing approaches having unbounded support. The proposed models have been validated through applications with speech, images and text datasets. The model are also used for feature extraction and selection of number of clusters and they have demonstrated their effectiveness in both task. As future work, attention could be devoted to the development of variational Bayesian inference and sampling-based approaches for model learning. The idea of bounded support mixture can also be extended to more distributions to improve the learning of clustering models.

Bibliography

[1] Y. Zhang and J. Glass, "Unsupervised Spoken Keyword Spotting via Segmental DTW on Gaussian Posteriorgrams," in *Automatic Speech Recognition Understanding, 2009. ASRU 2009. IEEE Workshop on*, Nov 2009, pp. 398–403.

[2] M. Azam and N. Bouguila, "Unsupervised Keyword Spotting using Bounded Generalized Gaussian Mixture Model with ICA," in *2015 IEEE GlobalSIP*, Dec 2015, pp. 1150–1154.

[3] Y. Zhang, "Unsupervised speech processing with applications to query-by-exampley-example spoken term detection," Ph.D. dissertation, MIT. Department of Electrical Engineering and Computer Science., 2013.

[4] X. Wu, X. Zhu, G. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, Jan 2014.

[5] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Information Fusion*, vol. 42, pp. 146–157, 2018.

[6] Y. Duan, J. S. Edwards, and Y. K. Dwivedi, "Artificial intelligence for decision making in the era of big data–evolution, challenges and research agenda," *International Journal of Information Management*, vol. 48, pp. 63–71, 2019.

[7] D. E. O'Leary, "Artificial intelligence and big data," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 96–99, 2013.

[8] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 67, 2016.

[9] K. Mahroof, "A human-centric perspective exploring the readiness towards smart warehousing: The case of a large retail distribution warehouse," *International Journal of Information Management*, vol. 45, pp. 176–190, 2019.

[10] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999. [Online]. Available: http://doi.acm.org/10.1145/331499.331504

[11] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014. [Online]. Available: http://science.sciencemag.org/content/344/6191/1492

[12] P. Purkait, T. J. Chin, A. Sadri, and D. Suter, "Clustering with hypergraphs: The case for large hyperedges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1697–1711, Sept 2017.

[13] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P. Patel, A. Tiwari, M. J. Er, W. Ding, and C.-T. Lin, "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, no. Supplement C, pp. 664 – 681, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231217311815

[14] S. Poddar and M. Jacob, "Clustering of data with missing entries using non-convex fusion penalties," *arXiv preprint arXiv:1709.01870*, 2017.

[15] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[16] D. Peel and G. McLachlan, "Robust Mixture Modelling using the t distribution," *Statistics and Computing*, vol. 10, no. 4, pp. 339–348, 2000.

[17] T. Elguebaly, "Unsupervised Selection and Estimation of Non-Gaussian Mixtures for High Dimensional Data Analysis," Ph.D. dissertation, Concordia University, 2014.

[18] T. M. Nguyen, Q. J. Wu, and H. Zhang, "Bounded Generalized Gaussian Mixture Model," *Pattern Recognition*, vol. 47, no. 9, 2014.

[19] M. Price, J. Glass, and A. Chandrakasan, "A 6 mW, 5,000-Word Real-Time Speech Recognizer Using WFST Models," *Solid-State Circuits, IEEE Journal of*, vol. 50, no. 1, pp. 102–112, Jan 2015.

[20] P. Jayashree and M. J. J. Premkumar, "Machine Learning in Automatic Speech Recognition: A Survey," *IETE Technical Review*, vol. 0, no. 0, pp. 1–12, 2015.

[21] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," 2018.

[22] N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," in *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1997, pp. 175–181.

[23] K. Blekas, A. Likas, N. P. Galatsanos, and I. E. Lagaris, "A spatially constrained mixture model for image segmentation," *IEEE transactions on Neural Networks*, vol. 16, no. 2, pp. 494–498, 2005.

[24] E. L. Thibodeau, K. E. Masyn, F. A. Rogosch, and D. Cicchetti, "Child maltreatment, adaptive functioning, and polygenic risk: A structural equation mixture model," *Development and psychopathology*, vol. 31, no. 2, pp. 443–456, 2019.

[25] C. Jung, C. Kim, S. W. Chae, and S. Oh, "Unsupervised segmentation of overlapped nuclei using bayesian classification," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 12, pp. 2825–2832, 2010.

[26] Y. Zhang and J. Glass, "Towards multi-speaker unsupervised speech pattern discovery," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, March 2010, pp. 4366–4369.

[27] D. Reynolds and R. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 1, pp. 72–83, Jan 1995.

[28] D. A. Reynolds, "Automatic speaker recognition using Gaussian mixture speaker models," *The Lincoln Laboratory Journal*, pp. 173–192, 1995.

[29] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1, pp. 19–41, 2000.

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

[31] G. McLachlan and D. Peel, *Finite mixture models*. John Wiley & Sons, 2004.

[32] T. Elguebaly and N. Bouguila, "Background subtraction using finite mixtures of asymmetric Gaussian distributions and shadow detection," *Machine Vision and Applications*, vol. 25, no. 5, pp. 1145–1162, 2014.

[33] G. Mclachlan and K. Basford, *Mixture Models: Inference and Applications to Clustering*, 01 1988, vol. 38.

[34] D. Titterington, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Distributions*. Wiley, New York, 1985.

[35] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382.

[36] L. Xu and M. I. Jordan, "On convergence properties of the em algorithm for gaussian mixtures," *Neural computation*, vol. 8, no. 1, pp. 129–151, 1996.

[37] R. M. Neal and G. E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in *Learning in graphical models*. Springer, 1998, pp. 355–368.

[38] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM review*, vol. 26, no. 2, pp. 195–239, 1984.

[39] S. J. Roberts, D. Husmeier, I. Rezek, and W. Penny, "Bayesian approaches to Gaussian mixture modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1133–1142, Nov 1998.

[40] T. Elguebaly and N. Bouguila, "A nonparametric Bayesian approach for enhanced pedestrian detection and foreground segmentation," in *CVPR 2011 WORKSHOPS*, June 2011, pp. 21–26.

[41] N. Bouguila, D. Ziou, and R. I. Hammoud, "A Bayesian Non-Gaussian Mixture Analysis: Application to Eye Modeling," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.

[42] A. Sefidpour and N. Bouguila, "Spatial color image segmentation based on finite non-Gaussian mixture models," *Expert Systems with Applications*, vol. 39, no. 10, pp. 8993 – 9001, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417412002680

[43] ——, "Spatial Finite Non-gaussian Mixture for Color Image Segmentation," in *Neural Information Processing*, B.-L. Lu, L. Zhang, and J. Kwok, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 514–521.

[44] N. Bouguila, D. Ziou, and S. Boutemedjet, "Simultaneous Non-gaussian Data Clustering, Feature Selection and Outliers Rejection," in *Pattern Recognition and Machine Intelligence*, S. O. Kuznetsov, D. P. Mandal, M. K. Kundu, and S. K. Pal, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 364–369.

[45] C. Liu and D. B. Rubin, "ML Estimation of the t distribution using EM and its Extensions, ECM and ECME," *Statistica Sinica*, vol. 5, no. 1, pp. 19–39, 1995.

[46] X. Wei and Z. Yang, "The Infinite Student's t-factor Mixture Analyzer for Robust Clustering and Classification," *Pattern Recognition*, vol. 45, no. 12, pp. 4346 – 4357, 2012.

[47] M. S. Allili, N. Bouguila, and D. Ziou, "Finite General Gaussian Mixture Modeling and Application to Image and Video Foreground Segmentation," *Journal of Electronic Imaging*, vol. 17, no. 1, pp. 013 005–013 005, 2008.

[48] M. Allili, "Wavelet Modeling Using Finite Mixtures of Generalized Gaussian Distributions: Application to Texture Discrimination and Retrieval," *Image Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 1452–1464, April 2012.

[49] M. Allili, N. Baaziz, and M. Mejri, "Texture Modeling Using Contourlets and Finite Mixtures of Generalized Gaussian Distributions and Applications," *Multimedia, IEEE Transactions on*, vol. 16, no. 3, pp. 772–784, April 2014.

[50] G. Liu, J. Wu, and S. Zhou, "Probabilistic Classifiers with a Generalized Gaussian Scale Mixture Prior," *Pattern Recognition*, vol. 46, no. 1, pp. 332 – 345, 2013.

[51] S. Choy and C. Tong, "Statistical Wavelet Subband Characterization Based on Generalized Gamma Density and Its Application in Texture Retrieval," *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 281–289, Feb 2010.

[52] A. Cord, C. Ambroise, and J.-P. Cocquerez, "Feature selection in robust clustering based on laplace mixture," *Pattern Recognition Letters*, vol. 27, no. 6, pp. 627–635, 2006.

[53] M. D. Ernst, "A multivariate generalized laplace distribution." *Comput. Statist.*, no. 13, pp. 227–232, 1998.

[54] V. M. Dang, "Classification de données spatiales: modèles probabilistes et critères de partitionnement," Ph.D. dissertation, Compiègne, 1998.

[55] Z. Ji, Y. Huang, Q. Sun, and G. Cao, "A spatially constrained generative asymmetric gaussian mixture model for image segmentation," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 611–626, 2016.

[56] T. Elguebaly and N. Bouguila, "Simultaneous high-dimensional clustering and feature selection using asymmetric Gaussian mixture models ," *Image and Vision Computing*, vol. 34, pp. 27 – 41, 2015.

[57] G. Wang, Z. Wang, Y. Chen, and W. Zhao, "A robust non-rigid point set registration method based on asymmetric gaussian representation," *Computer vision and image understanding*, vol. 141, pp. 67–80, 2015.

[58] T. Elguebaly and N. Bouguila, "Finite asymmetric generalized Gaussian mixture models learning for infrared object detection," *Computer Vision and Image Understanding*, vol. 117, no. 12, pp. 1659 – 1671, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314213001379

[59] ——, "Improving codebook generation for action recognition using a mixture of Asymmetric Gaussians," in *2014 IEEE Symposium on Computational Intelligence for Multimedia, Signal and Vision Processing (CIMSIVP)*, Dec 2014, pp. 1–7.

[60] A. Farag, A. El-Baz, and G. Gimel'farb, "Precise Segmentation of Multimodal Images," *IEEE Transactions on Image Processing*, vol. 15, no. 4, pp. 952–968, April 2006.

[61] P. Hedelin and J. Skoglund, "Vector quantization based on gaussian mixture models," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 4, pp. 385–401, Jul 2000.

[62] J. Lindblom and J. Samuelsson, "Bounded Support Gaussian Mixture Modeling of Speech Spectra," *Speech and Audio Processing, IEEE Transactions on*, vol. 11, no. 1, pp. 88–99, Jan 2003.

[63] A. D. Subramaniam and B. D. Rao, "Pdf optimized parametric vector quantization of speech line spectral frequencies," in *2000 IEEE Workshop on Speech Coding. Proceedings. Meeting the Challenges of the New Millennium (Cat. No.00EX421)*, 2000, pp. 87–89.

[64] P. Hedelin, J. Skoglund, and J. Samuelsson, "Performance bounds for lpc spectrum quantization," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, vol. 2, Mar 1999, pp. 677–680 vol.2.

[65] T. M. Nguyen and Q. M. J. Wu, "Bounded asymmetrical student's-t mixture model," *IEEE T. Cybernetics*, vol. 44, no. 6, pp. 857–869, 2014. [Online]. Available: http://dx.doi.org/10.1109/TCYB.2013.2273714

[66] ——, "A non-parametric bayesian model for bounded data," *Pattern Recognition*, vol. 48, no. 6, pp. 2084–2095, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2014.12.019

[67] M. Azam and N. Bouguila, "Speaker verification using adapted bounded gaussian mixture model," in *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, July 2018, pp. 300–307.

[68] T. Huang, H. Peng, and K. Zhang, "Model selection for gaussian mixture models," *Statistica Sinica*, pp. 147–169, 2017.

[69] G. Celeux, S. Fruewirth-Schnatter, and C. P. Robert, "Model selection for mixture models-perspectives and strategies," *arXiv preprint arXiv:1812.09885*, 2018.

[70] J. G. Campbell, C. Fraley, F. Murtagh, and A. E. Raftery, "Linear flaw detection in woven textiles using model-based clustering," *Pattern Recognition Letters*, vol. 18, no. 14, pp. 1539–1548, 1997.

[71] A. Dasgupta and A. E. Raftery, "Detecting features in spatial point processes with clutter via model-based clustering," *Journal of the American Statistical Association*, vol. 93, no. 441, pp. 294–302, 1998.

[72] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis," *The computer journal*, vol. 41, no. 8, pp. 578–588, 1998.

[73] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[74] J. Rissanen, *Stochastic complexity in statistical inquiry*. World scientific, 1998, vol. 15.

[75] M. P. Windham and A. Cutler, "Information ratios for validating mixture analyses," *Journal of the American Statistical Association*, vol. 87, no. 420, pp. 1188–1192, 1992.

[76] H. Bozdogan, *Choosing the Number of Component Clusters in the Mixture-Model Using a New Informational Complexity Criterion of the Inverse-Fisher Information Matrix*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 40–54. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-50974-2_5

[77] J. D. Banfield and A. E. Raftery, "Model-based gaussian and non-gaussian clustering," *Biometrics*, pp. 803–821, 1993.

[78] C. Biernacki and G. Govaert, "Using the classification likelihood to choose the number of clusters," *Computing Science and Statistics*, pp. 451–457, 1997.

[79] C. Biernacki, G. Celeux, and G. Govaert, "An improvement of the NEC criterion for assessing the number of clusters in a mixture model," *Pattern Recognition Letters*, vol. 20, no. 3, pp. 267 – 272, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865598001445

[80] G. Celeux and G. Soromenho, "An entropy criterion for assessing the number of clusters in a mixture model," *Journal of Classification*, vol. 13, no. 2, pp. 195–212, 1996. [Online]. Available: http://dx.doi.org/10.1007/BF01246098

[81] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated completed likelihood," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, pp. 719–725, Jul 2000.

[82] J. J. Oliver, R. A. Baxter, and C. S. Wallace, "Unsupervised learning using mml," in *ICML*, 1996, pp. 364–372.

[83] C. S. Wallace and D. L. Dowe, "Minimum message length and kolmogorov complexity," *The Computer Journal*, vol. 42, no. 4, pp. 270–283, 1999.

[84] 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.

[85] M. Azam and N. Bouguila, "Bounded Laplace Mixture Model with Applications to Image Clustering and Content Based Image Retrieval," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec 2018, pp. 558–563.

[86] ——, "Texture image categorization in wavelet domain via naive bayes classifier based on laplace and generalized gaussian distribution," in *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)*, July 2019, pp. 143–150.

[87] M. Azam and N. Bouguila, *Speaker Classification via Supervised Hierarchical Clustering Using ICA Mixture Model*. Cham: Springer International Publishing, 2016, pp. 193–202. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-33618-3_20

[88] M. Azam and N. Bouguila, "Blind source separation as pre-processing to unsupervised keyword spotting via an ica mixture model," in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug 2018, pp. 833–836.

[89] M. Azam and N. Bouguila, "Bounded Generalized Gaussian Mixture Model with ICA," *Neural Processing Letters*, vol. 49, no. 3, pp. 1299–1320, Jun 2019. [Online]. Available: https://doi.org/10.1007/s11063-018-9868-7

[90] M. Azam, B. Alghabashi, and N. Bouguila, *Multivariate Bounded Asymmetric Gaussian Mixture Model*. Cham: Springer International Publishing, 2020, pp. 61–80. [Online]. Available: https://doi.org/10.1007/978-3-030-23876-6_4

[91] P. Kabal, "TSP speech database," Department of Electrical & Computer Engineering, McGill University, Montreal, Quebec, Canada, Tech. Rep., 2002.

[92] "Free spoken digit dataset," https://github.com/Jakobovski/free-spoken-digit-dataset.

[93] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[94] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017.

[95] F. B. Pokorny, F. Graf, F. Pernkopf, and B. W. Schuller, "Detection of negative emotions in speech signals using bags-of-audio-words," in *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, Sep. 2015, pp. 879–884.

[96] R. Grzeszick, A. Plinge, G. A. Fink, R. Grzeszick, A. Plinge, and G. A. Fink, "Bag-of-features methods for acoustic event detection and classification," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 25, no. 6, pp. 1242–1252, Jun. 2017. [Online]. Available: https://doi.org/10.1109/TASLP.2017.2690574

[97] S. Pancoast and M. Akbacak, "Bag-of-audio-words approach for multimedia event classification," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[98] Y. Liu, W.-L. Zhao, C.-W. Ngo, C.-S. Xu, and H.-Q. Lu, "Coherent bag-of audio words model for efficient large-scale video copy detection," in *Proceedings of the ACM international conference on image and video retrieval*. ACM, 2010, pp. 89–96.

[99] B. Ziolko, S. Manandhar, and R. C. Wilson, "Bag-of-words modelling for speech recognition," in *2009 International Conference on Future Computer and Communication*, April 2009, pp. 646–650.

[100] I. A. Sheikh, I. Illina, D. Fohr, and G. Linarès, "Learning to retrieve out-of-vocabulary words in speech recognition," *ArXiv*, vol. abs/1511.05389, 2015.

[101] I. Sheikh, I. Illina, D. Fohr, and G. Linares, "Improved neural bag-of-words model to retrieve out-of-vocabulary words in speech recognition," 2016.

[102] E. Spyrou, R. Nikopoulou, I. Vernikos, and P. Mylonas, "Emotion recognition from speech using the bag-of-visual words on audio segment spectrograms," *Technologies*, vol. 7, no. 1, p. 20, 2019.

[103] T. Elguebaly and N. Bouguila, "Semantic Scene Classification with Generalized Gaussian Mixture Models," in *Image Analysis and Recognition*, M. Kamel and A. Campilho, Eds. Cham: Springer International Publishing, 2015, pp. 159–166.

[104] 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: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[105] G. J. McLachlan, S. X. Lee, and S. I. Rathnayake, "Finite mixture models," *Annual review of statistics and its application*, vol. 6, pp. 355–378, 2019.

[106] R. Espindola and N. Ebecken, "On extending f-measure and g-mean metrics to multi-class problems," *WIT Transactions on Information and Communication Technologies*, vol. 35, 2005.

[107] G. Jurman and C. Furlanello, "A unifying view for performance measures in multi-class prediction," *arXiv preprint arXiv:1008.2908*, 2010.

[108] G. Jurman, S. Riccadonna, and C. Furlanello, "A comparison of mcc and cen error measures in multi-class prediction," *PloS one*, vol. 7, no. 8, p. e41882, 2012.

[109] J. Gorodkin, "Comparing two k-category assignments by a k-category correlation coefficient," *Computational biology and chemistry*, vol. 28, no. 5-6, pp. 367–374, 2004.

[110] K. E. Ihou and N. Bouguila, "A new latent generalized dirichlet allocation model for image classification," in *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Nov 2017, pp. 1–6.

[111] Pan Xiao, Nian Cai, Bochao Tang, Shaowei Weng, and Han Wang, "Efficient sift descriptor via color quantization," in *2014 IEEE International Conference on Consumer Electronics - China*, April 2014, pp. 1–3.

[112] X. Zhou, K. Wang, and J. Fu, "A method of sift simplifying and matching algorithm improvement," in *2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, Dec 2016, pp. 73–77.

[113] R. A. Baxter and J. J. Oliver, "Finding overlapping components with mml," *Statistics and Computing*, vol. 10, no. 1, pp. 5–16, 2000. [Online]. Available: http://dx.doi.org/10.1023/A:1008928315401

[114] C. S. Wallace and D. M. Boulton, "An information measure for classification," *The Computer Journal*, vol. 11, no. 2, pp. 185–194, 1968. [Online]. Available: http://comjnl.oxfordjournals.org/content/11/2/185.abstract

[115] J. H. Conway and N. J. A. Sloane, *Sphere packings, lattices and groups*. Springer Science & Business Media, 2013, vol. 290.

[116] 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, pp. 1716–1731, Oct 2007.

[117] Y. Agusta and D. L. Dowe, "Unsupervised learning of correlated multivariate gaussian mixture models using mml," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2003, pp. 477–489.

[118] P. Kasarapu and L. Allison, "Minimum message length estimation of mixtures of multivariate gaussian and von mises-fisher distributions," *Machine Learning*, vol. 100, no. 2-3, pp. 333–378, 2015.

[119] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, December 1974.

[120] H. Bozdogan, "Model selection and akaike's information criterion (aic): The general theory and its analytical extensions," *Psychometrika*, vol. 52, no. 3, pp. 345–370, 1987.

[121] M. A. Figueiredo, J. M. Leitão, and A. K. Jain, "On fitting mixture models," in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer, 1999, pp. 54–69.

[122] F. Khozeimeh, R. Alizadehsani, M. Roshanzamir, A. Khosravi, P. Layegh, and S. Nahavandi, "An expert system for selecting wart treatment method," *Computers in biology and medicine*, vol. 81, pp. 167–175, 2017.

[123] F. Khozeimeh, F. Jabbari Azad, Y. Mahboubi Oskouei, M. Jafari, S. Tehranian, R. Alizadehsani, and P. Layegh, "Intralesional immunotherapy compared to cryotherapy in the treatment of warts," *International journal of dermatology*, vol. 56, no. 4, pp. 474–478, 2017.

[124] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[125] M. A. Little, P. E. McSharry, S. J. Roberts, D. A. Costello, and I. M. Moroz, "Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection," *Biomedical engineering online*, vol. 6, no. 1, p. 23, 2007.

[126] M. Patrício, J. Pereira, J. Crisóstomo, P. Matafome, M. Gomes, R. Seiça, and F. Caramelo, "Using resistin, glucose, age and bmi to predict the presence of breast cancer," *BMC cancer*, vol. 18, no. 1, p. 29, 2018.

[127] M. Elter, R. Schulz-Wendtland, and T. Wittenberg, "The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process," *Medical physics*, vol. 34, no. 11, pp. 4164–4172, 2007.

[128] I.-C. Yeh, K.-J. Yang, and T.-M. Ting, "Knowledge discovery on rfm model using bernoulli sequence," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5866–5871, 2009.

[129] D. Gil, J. L. Girela, J. De Juan, M. J. Gomez-Torres, and M. Johnsson, "Predicting seminal quality with artificial intelligence methods," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12 564–12 573, 2012.

[130] L. A. Kurgan, K. J. Cios, R. Tadeusiewicz, M. Ogiela, and L. S. Goodenday, "Knowledge discovery approach to automated cardiac spect diagnosis," *Artificial intelligence in medicine*, vol. 23, no. 2, pp. 149–169, 2001.

[131] J. Hansen and T. Hasan, "Speaker Recognition by Machines and Humans: A tutorial review," *Signal Processing Magazine, IEEE*, vol. 32, no. 6, pp. 74–99, Nov 2015.

[132] J. Markowitz, "The Many Roles of Speaker Classification in Speaker Verification and Identification," in *Speaker Classification I*, ser. Lecture Notes in Computer Science, C. Müller, Ed. Springer Berlin Heidelberg, 2007, vol. 4343, pp. 218–225.

[133] D. A. Reynolds, "An overview of automatic speaker recognition technology," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, May 2002, pp. IV–4072–IV–4075.

[134] D. Reynolds, "Universal Background Models," in *Encyclopedia of Biometrics*, S. Z. Li and A. K. Jain, Eds. Boston, MA: Springer US, 2015, pp. 1547–1550. [Online]. Available: https://doi.org/10.1007/978-1-4899-7488-4_197

[135] T. Hasan and J. H. L. Hansen, "A Study on Universal Background Model Training in Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 1890–1899, Sept 2011.

[136] R. B. Dunn, D. A. Reynolds, and T. F. Quatieri, "Approaches to speaker detection and tracking in conversational speech," *Digital signal processing*, vol. 10, no. 1-3, pp. 93–112, 2000.

[137] D. E. Sturim, W. M. Campbell, and D. A. Reynolds, "Classification methods for speaker recognition," in *Speaker Classification I*. Springer, 2007, pp. 278–297.

[138] J. L. Marcano, M. A. Bell, and A. L. Beex, "Classification of ADHD and non-ADHD subjects using a universal background model," *Biomedical Signal Processing and Control*, vol. 39, pp. 204 – 212, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1746809417301556

[139] A. K. Sarkar and Z.-H. Tan, "Incorporating pass-phrase dependent background models for text-dependent speaker verification," *Computer Speech & Language*, vol. 47, pp. 259 – 271, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230816303795

[140] E. Khoury and M. Garland, "Dimensionality reduction of baum-welch statistics for speaker recognition," Mar. 22 2018, US Patent App. 15/709,232.

[141] J. L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, Apr 1994.

[142] M. K. Omar and J. W. Pelecanos, "Training universal background models for speaker recognition." in *Odyssey: The Speaker and Language Recognition Workshop*, 2010, p. 10.

[143] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT Acoustic Phonetic Continuous Speech Corpus CDROM," http://www.ldc.upenn.edu/Catalog/LDC93S1.html.

[144] P. S. Bradley, O. L. Mangasarian, and W. N. Street, "Clustering via concave minimization," in *Advances in neural information processing systems*, 1997, pp. 368–374.

[145] N. Mitianoudis and T. Stathaki, "Overcomplete source separation using laplacian mixture models," *Signal Processing Letters, IEEE*, vol. 12, no. 4, pp. 277–280, April 2005.

[146] D. Bhowmick, A. Davison, D. R. Goldstein, and Y. Ruffieux, "A laplace mixture model for identification of differential expression in microarray experiments," *Biostatistics*, vol. 7, no. 4, pp. 630–641, 2006.

[147] F. Shi and I. W. Selesnick, "Multivariate quasi-laplacian mixture models forwavelet-based image denoising," in *2006 International Conference on Image Processing*, Oct 2006, pp. 2625–2628.

[148] T. Eltoft, T. Kim, and T.-W. Lee, "On the multivariate laplace distribution," *Signal Processing Letters, IEEE*, vol. 13, no. 5, pp. 300–303, 2006.

[149] B. C. Franczak, R. P. Browne, and P. D. McNicholas, "Mixtures of shifted asymmetric-claplace distributions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1149–1157, June 2014.

[150] C. Scricciolo, "Bayes and maximum likelihood for L-1-wasserstein deconvolution of laplace mixtures," *Statistical Methods & Applications*, vol. 27, no. 2, pp. 333–362, 2018.

[151] A. Najjar, C. Gagne, and D. Reinharz, "Two-step heterogeneous finite mixture model clustering for mining healthcare databases," in *2015 IEEE International Conference on Data Mining*, Nov 2015, pp. 931–936.

[152] L. Garg, S. McClean, B. Meenan, E. El-Darzi, and P. Millard, "Clustering patient length of stay using mixtures of gaussian models and phase type distributions," in *2009 22nd IEEE International Symposium on Computer-Based Medical Systems*, Aug 2009, pp. 1–7.

[153] I. Banerjee, C. Kurtz, A. E. Devorah, B. Do, D. L. Rubin, and C. F. Beaulieu, "Relevance feedback for enhancing content based image retrieval and automatic

prediction of semantic image features: Application to bone tumor radiographs," *Journal of Biomedical Informatics*, vol. 84, pp. 123 – 135, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S153204641830128X

[154] S. Bai, X. Bai, Q. Tian, and L. J. Latecki, "Regularized diffusion process on bidirectional context for object retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018.

[155] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1224–1244, May 2018.

[156] M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain, "Simultaneous feature selection and clustering using mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1154–1166, Sept 2004.

[157] M. B. Mayhew, B. K. Petersen, A. P. Sales, J. D. Greene, V. X. Liu, and T. S. Wasson, "Flexible, cluster-based analysis of the electronic medical record of sepsis with composite mixture models," *Journal of Biomedical Informatics*, vol. 78, pp. 33 – 42, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1532046417302691

[158] J. Sun, A. Zhou, S. Keates, and S. Liao, "Simultaneous bayesian clustering and feature selection through student's t mixtures model," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 1187–1199, April 2018.

[159] T. Amin, M. Zeytinoglu, and L. Guan, "Application of laplacian mixture model to image and video retrieval," *IEEE Transactions on Multimedia*, vol. 9, no. 7, pp. 1416–1429, 2007.

[160] S. Medasani and R. Krishnapuram, "Categorization of image databases for efficient retrieval using robust mixture decomposition," *Computer Vision and Image Understanding*, vol. 83, no. 3, pp. 216 – 235, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314201909269

[161] ——, "Categorization of image databases for efficient retrieval using robust mixture decomposition," in *Proceedings. IEEE Workshop on Content-Based Access of Image and Video Libraries (Cat. No.98EX173)*, Jun 1998, pp. 50–54.

[162] H. Yuan, X.-P. Zhang, and L. Guan, "Content-based image retrieval using a gaussian mixture model in the wavelet domain," in *Proc.SPIE*, vol. 5150, 2003, pp. 5150 – 5150 – 8. [Online]. Available: https://doi.org/10.1117/12.503262

[163] H. Yuan and X.-P. Zhang, "Texture image retrieval based on a gaussian mixture model and similarity measure using a kullback divergence," in *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, vol. 3, June 2004, pp. 1867–1870 Vol.3.

[164] H. Yuan and X. P. Zhang, "Statistical modeling in the wavelet domain for compact feature extraction and similarity measure of images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 3, pp. 439–445, March 2010.

[165] J. K. Romberg, H. Choi, and R. G. Baraniuk, "Bayesian tree-structured image modeling using wavelet-domain hidden markov models," *IEEE Transactions on Image Processing*, vol. 10, no. 7, pp. 1056–1068, Jul 2001.

[166] H. A. Chipman, E. D. Kolaczyk, and R. E. McCulloch, "Adaptive bayesian wavelet shrinkage," *Journal of the American Statistical Association*, vol. 92, no. 440, pp. 1413–1421, 1997.

[167] E. P. Simoncelli, "Statistical models for images: Compression, restoration and synthesis," in *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers, 1997*, vol. 1. IEEE, 1997, pp. 673–678.

[168] S. Gai, B. Zhang, C. Yang, and L. Yu, "Speckle noise reduction in medical ultrasound image using monogenic wavelet and laplace mixture distribution," *Digital Signal Processing*, vol. 72, pp. 192 – 207, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1051200417302233

[169] Y. Chen and J. Z. Wang, "Image categorization by learning and reasoning with regions," *Journal of Machine Learning Research*, vol. 5, no. Aug, pp. 913–939, 2004.

[170] Y. Song, Q. Li, D. Feng, J. J. Zou, and W. Cai, "Texture image classification with discriminative neural networks," *Computational Visual Media*, vol. 2, no. 4, pp. 367–377, 2016.

[171] V. Andrearczyk and P. F. Whelan, "Deep learning in texture analysis and its application to tissue image classification," in *Biomedical Texture Analysis*. Elsevier, 2017, pp. 95–129.

[172] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *International journal of computer vision*, vol. 62, no. 1-2, pp. 61–81, 2005.

[173] B. E. Usevitch, "A tutorial on modern lossy wavelet image compression: foundations of jpeg 2000," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 22–35, Sep 2001.

[174] B. E. Usevitch and M. T. Orchard, "Smooth wavelets, transform coding, and markov-1 processes," *IEEE Transactions on Signal Processing*, vol. 43, no. 11, pp. 2561–2569, Nov 1995.

[175] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: a power tool for interactive content-based image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 644–655, Sep 1998.

[176] F. Malik and B. Baharudin, "Analysis of distance metrics in content-based image retrieval using statistical quantized histogram texture features in the dct domain," *Journal of King Saud University - Computer and Information Sciences*, vol. 25, no. 2, pp. 207 – 218, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1319157812000444

[177] P. C. Cheeseman, J. C. Stutz *et al.*, "Bayesian classification (autoclass): theory and results." *Advances in knowledge discovery and data mining*, vol. 180, pp. 153–180, 1996.

[178] I. Rish, "An empirical study of the naive bayes classifier," Tech. Rep., 2001.

[179] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265–1278, Aug 2005.

[180] M. Fritz, E. Hayman, B. Caputo, and J. olof Eklundh, "The kth-tips database," 2004.

[181] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[182] "Salzburg texture image database (stex)," http://www.wavelab.at/sources/STex/, accessed: 2019-01-15.

[183] G. Kylberg, "The kylberg texture dataset v. 1.0," Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, Uppsala, Sweden, External report (Blue series) 35, September 2011. [Online]. Available: http://www.cb.uu.se/~gustaf/texture/

[184] F. Peng and D. Schuurmans, "Combining naive bayes and n-gram language models for text classification," in *European Conference on Information Retrieval*. Springer, 2003, pp. 335–350.

[185] F. Demichelis, P. Magni, P. Piergiorgi, M. A. Rubin, and R. Bellazzi, "A hierarchical naive bayes model for handling sample heterogeneity in classification problems: an application to tissue microarrays," *BMC bioinformatics*, vol. 7, no. 1, p. 514, 2006.

[186] J. Chen, H. Huang, S. Tian, and Y. Qu, "Feature selection for text classification with naïve bayes," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5432–5435, 2009.

[187] J. C. Griffis, J. B. Allendorfer, and J. P. Szaflarski, "Voxel-based gaussian naïve bayes classification of ischemic stroke lesions in individual t1-weighted mri scans," *Journal of neuroscience methods*, vol. 257, pp. 97–108, 2016.

[188] W. Lou, X. Wang, F. Chen, Y. Chen, B. Jiang, and H. Zhang, "Sequence based prediction of dna-binding proteins based on hybrid feature selection using random forest and gaussian naive bayes," *PLoS One*, vol. 9, no. 1, p. e86703, 2014.

[189] R. Marcos De Moraes and L. Dos Santos Machado, "Online assessment in medical simulators based on virtual reality using fuzzy gaussian naive bayes." *Journal of Multiple-Valued Logic & Soft Computing*, vol. 18, 2012.

[190] S. Raschka, "Naive bayes and text classification i-introduction and theory," *arXiv preprint arXiv:1410.5329*, 2014.

[191] Y. Tsuruoka and J. Tsujii, "Training a naive bayes classifier via the em algorithm with a class distribution constraint," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003, pp. 127–134.

[192] M. Collins, "The naive bayes model, maximum-likelihood estimation, and the em algorithm," *Lecture Notes*, 2012.

[193] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[194] N. Bouguila, K. Almakadmeh, and S. Boutemedjet, "A finite mixture model for simultaneous high-dimensional clustering, localized feature selection and outlier rejection," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6641–6656, 2012.

[195] T.-W. Lee and M. S. Lewicki, "The generalized Gaussian mixture model using ICA," in *International Workshop on ICA*, 2000, pp. 239–244.

[196] T. W. Lee, M. S. Lewicki, and T. J. Sejnowski, "ICA Mixture Models for Unsupervised Classification with non-Gaussian Sources and Automatic Context Switching in Blind Signal Separation," in *IEEE Transactions on Pattern Recognition and Machine Learning*, 2000.

[197] T.-W. Lee and M. S. Lewicki, "Unsupervised image classification, segmentation, and enhancement using ICA mixture models," *Image Processing, IEEE Transactions on*, vol. 11, no. 3, pp. 270–279, 2002.

[198] T.-W. Lee, M. S. Lewicki, and T. J. Sejnowski, "Unsupervised classification with non-gaussian mixture models using ica," *Advances in neural information processing systems*, pp. 508–514, 1999.

[199] A. Salazar, "ICA and ICAMM Methods," in *On Statistical Pattern Recognition in Independent Component Analysis Mixture Modelling*, ser. Springer Theses. Springer Berlin Heidelberg, 2013, vol. 4.

[200] R. A. Choudrey and S. J. Roberts, "Variational mixture of bayesian independent component analyzers," *Neural Computation*, vol. 15, no. 1, pp. 213–252, 2003.

[201] M. N. H. Mollah, M. Minami, and S. Eguchi, "Exploring latent structure of mixture ica models by the minimum β-divergence method," *Neural Computation*, vol. 18, no. 1, pp. 166–190, 2006.

[202] C. A. Shah, M. K. Arora, and P. K. Varshney, "Unsupervised classification of hyperspectral data: an ICA mixture model based approach," *International Journal of Remote Sensing*, vol. 25, no. 2, pp. 481–487, 2004.

[203] C. A. Shah, P. K. Varshney, and M. K. Arora, "ICA Mixture Model Algorithm for Unsupervised Classification of Remote Sensing Imagery," *International Journal of Remote Sensing*, vol. 28, no. 8, pp. 1711–1731, 2007.

[204] P. B. Ribeiro, R. A. F. Romero, P. R. Oliveira, H. Schiabel, and L. B. Vercosa, "Automatic segmentation of breast masses using enhanced ICA mixture model," *Neurocomputing*, vol. 120, no. 0, pp. 61 – 71, 2013, image Feature Detection and Description.

[205] J. Palmer, K. Kreutz-delgado, and S. Makeig, "An independent component analysis mixture model with adaptive source densities," 2006.

[206] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[207] X. Huang, A. Acero, and H. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[208] Y. Zhang, "Unsupervised spoken keyword spotting and learning of acoustically meaningful units," Master's thesis, Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science, 2009.

[209] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic Speech Recognition for Under-Resourced Languages: A Survey," *Speech Communication*, vol. 56, no. 0, pp. 85 – 100, 2014.

[210] S. Krauwer, "The Basic Language Resource Kit (BLARK) as the First Milestone for the Language Resources Roadmap," in *Proceedings of the 2003 International Workshop Speech and Computer (SPECOM 2003)*. Moscow State Linguistic University, 2003, pp. 8–15.

[211] V. Berment, "Methodes pour informatiser des langues et des groupes de langues peu dotees," Ph.D. dissertation, Joseph Fourier University Grenoble I, May 2004.

[212] "Vistawide World Languages and Cultures: General," http://www.vistawide.com/languages/language_statistics.htm, online accessed on 25-February-2015.

[213] Nuance Communications, Inc., "Nuance recognizer for speech," http://www.nuance.com/for-business/automatic-speech-recognition/automated-ivr/index.htm, online accessed 25-February-2015.

[214] I. Szoke, P. Schwarz, L. Burget, M. Fapso, M. Karafiat, J. Cernocky, and P. Matejka, "Comparison of Keyword Spotting Approaches for Informal Continuous Speech," in *In Proceedings, Interspeech*, Sep. 2005.

[215] McGraw-Hill, "Keyword spotting. (n.d.) mcgraw-hill dictionary of scientific & technical terms, 6e. (2003)," http://encyclopedia2.thefreedictionary.com/keyword+spotting, retrieved on March 31 2015.

[216] M. H. Siu, H. Gish, A. Chan, W. Belfield, and S. Lowe, "Unsupervised training of an hmm-based self-organizing unit recognizer with applications to topic classification and keyword discovery," *Computer Speech & Language*, vol. 28, no. 1, pp. 210 – 223, 2014.

[217] R. Rose and D. Paul, "A hidden markov model based keyword recognition system," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, Apr 1990, pp. 129–132 vol.1.

[218] Y. Takebayashi, H. Tsuboi, and H. Kanazawa, "Keyword-spotting in noisy continuous speech using word pattern vector subabstraction and noise immunity learning," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 2, Mar 1992, pp. 85–88 vol.2.

[219] L. Wilcox and M. Bush, "Training and search algorithms for an interactive wordspotting system," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 2, Mar 1992, pp. 97–100 vol.2.

[220] J. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden markov modeling for speaker-independent word spotting," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, May 1989, pp. 627–630 vol.1.

[221] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 26, no. 1, pp. 43–49, Feb 1978.

[222] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, Aug 1980.

[223] C. Myers and L. Rabiner, "A level building dynamic time warping algorithm for connected word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, no. 2, pp. 284–297, Apr 1981.

[224] A. Park and J. Glass, "Towards unsupervised pattern discovery in speech," in *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, Nov 2005, pp. 53–58.

[225] ——, "Unsupervised word acquisition from speech using pattern discovery," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, May 2006, pp. I–I.

[226] ——, "Unsupervised Pattern Discovery in Speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, Jan 2008.

[227] Y. Zhang and J. Glass, "An Inner-Product Lower-Bound Estimate for Dynamic Time Warping," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, May 2011, pp. 5660–5663.

[228] W. Li and Q. Liao, "Keyword-Specific Normalization Based Keyword Spotting for Spontaneous Speech," in *Chinese Spoken Language Processing (ISCSLP), 2012 8th International Symposium on*, Dec 2012, pp. 233–237.

[229] H. Wang, T. Lee, C. C. Leung, B. Ma, and H. Li, "Unsupervised Mining of Acoustic Subword Units with Segment-Level Gaussian Posteriorgrams," in *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*, 2013, pp. 2297–2301.

[230] S. Bourouis, M. A. Mashrgy, and N. Bouguila, "Bayesian learning of finite generalized inverted dirichlet mixtures: Application to object classification and forgery detection," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2329 – 2336, 2014.

[231] T. Bdiri, N. Bouguila, and D. Ziou, "Visual scenes categorization using a flexible hierarchical mixture model supporting users ontology," in *2013 IEEE 25th International Conference on Tools with Artificial Intelligence, Herndon, VA, USA, November 4-6, 2013*, 2013, pp. 262–267.

[232] ——, "Object clustering and recognition using multi-finite mixtures for semantic classes and hierarchy modeling," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1218–1235, 2014.

[233] E. Vincent, N. Bertin, R. Gribonval, and F. Bimbot, "From blind to guided audio source separation: How models and side information can improve the separation of sound," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 107–115, May 2014.

[234] J. J. Thiagarajan, K. N. Ramamurthy, and A. Spanias, "Mixing matrix estimation using discriminative clustering for blind source separation," *Digital Signal Processing*, vol. 23, no. 1, pp. 9 – 18, 2013.

[235] J. Cardoso, "Infomax and Maximum Likelihood for Blind Source Separation," *Signal Processing Letters, IEEE*, vol. 4, no. 4, April 1997.

[236] T. Peng, Y. Chen, and Z. Liu, "A time frequency domain blind source separation method for underdetermined instantaneous mixtures," *Circuits, Systems, and Signal Processing*, pp. 1–13, 2015. [Online]. Available: http://dx.doi.org/10.1007/s00034-015-0035-3

[237] B. Ans, J. Hérault, and C. Jutten, "Adaptive neural architectures: detection of primitives," *Proceedings of COGNITIVA*, vol. 85, pp. 593–597, 1985.

[238] J. Herault and C. Jutten, "Space or time adaptive signal processing by neural network models," in *Neural networks for computing*, vol. 151, no. 1. AIP Publishing, 1986, pp. 206–211.

[239] J. Herault, C. Jutten, and B. ANS, "Detection de grandeurs primitives dans un message composite par une architeture de calcul neuromimetique en apprentissage non supervise," *10 Colloque sur le traitement du signal et des images, 1985 ; p. 1017-1022*, 01 1985.

[240] C. Jutten and J. Herault, "Blind separation of sources, part 1: An adaptive algorithm based on neuromimetic architecture," *Signal Process.*, vol. 24, no. 1, pp. 1–10, Aug. 1991.

[241] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, 1st ed. Academic Press, 2010.

[242] C. Jutten, "Calcul neuromimétique et traitement du signal: analyse en composantes indépendantes," Ph.D. dissertation, Grenoble INPG, 1987.

[243] P. Comon, "Independent Component Analysis," *Higher-Order Statistics*, pp. 29–38, 1992.

[244] ——, "Independent component analysis, a new concept?" *Signal processing*, vol. 36, no. 3, pp. 287–314, 1994.

[245] T. W. Lee, M. Girolami, and T. J. Sejnowski, "Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources," 1999.

[246] U.-M. Bae, T.-W. Lee, and S.-Y. Lee, "Blind signal separation in teleconferencing using ica mixture model," *Electronics Letters*, vol. 36, no. 7, pp. 680–682, Mar 2000.

[247] F. Gu, H. Zhang, W. Wang, and S. Wang, "An expectation-maximization algorithm for blind separation of noisy mixtures using gaussian mixture model," *Circuits, Systems, and Signal Processing*, vol. 36, no. 7, pp. 2697–2726, 2017. [Online]. Available: http://dx.doi.org/10.1007/s00034-016-0424-2

[248] A. Alinaghi, P. J. Jackson, Q. Liu, and W. Wang, "Joint mixing vector and binaural model based stereo source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 9, pp. 1434–1448, Sept 2014.

[249] K. B. Petersen and O. Winther, "The em algorithm in independent component analysis," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 5, March 2005, pp. v/169–v/172 Vol. 5.

[250] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*. John Wiley & Sons, 2004, vol. 46.

[251] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2046–2057, Sept 2011.

[252] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, July 2006.

[253] P. Mowlaee, R. Saeidi, M. G. Christensen, and R. Martin, "Subjective and objective quality assessment of single-channel speech separation algorithms," in *2012 IEEE ICASSP*, March 2012, pp. 69–72.

[254] L. D. Persia, D. Milone, H. L. Rufiner, and M. Yanagida, "Perceptual evaluation of blind source separation for robust speech recognition," *Signal Processing*, vol. 88, no. 10, pp. 2578 – 2583, 2008.

[255] T. Hazen, W. Shen, and C. White, "Query-By-Example Spoken Term Detection using Phonetic Posteriorgram Templates," in *IEEE Workshop on ASRU 2009.*, Nov 2009, pp. 421–426.

[256] P. Nguyen, T. Le, D. Tran, X. Huang, and D. Sharma, "Fuzzy Support Vector Machines for Age and Gender Classification," in *INTERSPEECH*, 2010, pp. 2806–2809.

[257] R. Vergin, A. Farhat, and D. O'Shaughnessy, "Robust gender-dependent acoustic-phonetic modelling in continuous speech recognition based on a new automatic male/female classification," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 2, Oct 1996, pp. 1081–1084 vol.2.

[258] Y. Hu and P. Loizou, "Noizeus: A noisy speech corpus for evaluation of speech enhancement algorithms," 2007, online web resource. [Online]. Available: http://ecs.utdallas.edu/loizou/speech/noizeus/

[259] A. J. Bell and T. J. Sejnowski, "An Information-Maximization Approach to Blind Separation and Blind Deconvolution," *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[260] T. Kato, S. Omachi, and H. Aso, "Asymmetric gaussian and its application to pattern recognition," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 2002, pp. 405–413.

[261] H. Xu and B. Yu, "Automatic thesaurus construction for spam filtering using revised back propagation neural network," *Expert Systems with Applications*, vol. 37, no. 1, pp. 18 – 23, 2010.

[262] J. Hong, "The state of phishing attacks," *Commun. ACM*, vol. 55, no. 1, pp. 74–81, Jan. 2012.

[263] I. Park, R. Sharman, H. Raghav Rao, and S. Upadhyaya, "The effect of spam and privacy concerns on e-mail users' behavior," *ACM Transactions on Information and System Security - TISSEC*, 01 2016.

[264] S. Fu and N. Bouguila, "Asymmetric gaussian mixtures with reversible jump mcmc," in *2018 IEEE Canadian Conference on Electrical Computer Engineering (CCECE)*, May 2018, pp. 1–4.

[265] M. Wang, W. Zhang, Y. Zhang, and X. Ji, "Detecting image spam based on cross entropy," in *2011 Eighth Web Information Systems and Applications Conference*, Oct 2011, pp. 19–22.

[266] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[267] V. Viitaniemi and J. Laaksonen, "Techniques for still image scene classification and object detection," in *Artificial Neural Networks – ICANN 2006*, S. Kollias, A. Stafylopatis, W. Duch, and E. Oja, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 35–44.

[268] K. E. Ihou and N. Bouguila, "Variational-based latent generalized dirichlet allocation model in the collapsed space and applications," *Neurocomputing*, vol. 332, pp. 372 – 395, 2019.

[269] W. Fan, N. Bouguila, and D. Ziou, "Variational learning for finite dirichlet mixture models and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 5, pp. 762–774, May 2012.

[270] D. Yin, J. Pan, P. Chen, and R. Zhang, "Medical image categorization based on gaussian mixture model," in *2008 International Conference on BioMedical Engineering and Informatics*, vol. 2, May 2008, pp. 128–131.

[271] L. Fei-Fei, R. Fergus, and P. 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*, June 2004, pp. 178–178.

[272] G.-H. Liu, Z.-Y. Li, L. Zhang, and Y. Xu, "Image retrieval based on micro-structure descriptor," *Pattern Recognition*, vol. 44, no. 9, pp. 2123–2133, 2011.

[273] G.-H. Liu, J.-Y. Yang, and Z. Li, "Content-based image retrieval using computational visual attention model," *pattern recognition*, vol. 48, no. 8, pp. 2554–2566, 2015.

[274] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov 2004.

[275] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features. iccv (pp. 1458–1465)," *IEEE Computer Society*, 2005.

[276] A. D. Holub, M. Welling, and P. Perona, "Combining generative models and fisher kernels for object recognition," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 1. IEEE, 2005, pp. 136–143.

[277] A. Holub, M. Welling, and P. Perona, "Exploiting unlabelled data for hybrid object classification," in *Proc. Neural Information Processing Systems, Workshop Inter-Class Transfer*, vol. 7, 2005, p. 2.

[278] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "Svm-knn: Discriminative nearest neighbor classification for visual category recognition," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 2126–2136.

[279] J. Mutch and D. G. Lowe, "Multiclass object recognition with sparse, localized features," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1. IEEE, 2006, pp. 11–18.

[280] A. C. Berg, T. L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," in *CVPR (1)*. Citeseer, 2005, pp. 26–33.

[281] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," Massachusetts Institure of Technology Cambridge Deptartment of Brain and Cognitive Sceinces, Tech. Rep., 2006.

[282] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 2169–2178.

[283] M. J. Marin-Jimenez and N. P. De La Blanca, "Empirical study of multi-scale filter banks for object categorization." in *ICPR (1)*. Citeseer, 2006, pp. 578–581.

[284] G. Wang, Y. Zhang, and L. Fei-Fei, "Using dependent regions for object categorization in a generative framework," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1597–1604.

[285] M. Jian, L. Liu, and F. Guo, "Texture image classification using perceptual texture features and gabor wavelet features," in *2009 Asia-Pacific Conference on Information Processing*, vol. 2, July 2009, pp. 55–58.

[286] T. Braunl, D.-I. Stefan Feyrer, D.-I. Wolfgang Rapf, and D.-I. Michael Reinhardt, "Texture recognition," pp. 121–130, 01 2001.

[287] A. Chadha, S. Mallik, and R. Johar, "Comparative study and optimization of feature-extraction techniques for content based image retrieval," *arXiv preprint arXiv:1208.6335*, 2012.

[288] X. Tang, "Texture information in run-length matrices," *IEEE transactions on image processing*, vol. 7, no. 11, pp. 1602–1609, 1998.

[289] F. R. De Siqueira, W. R. Schwartz, and H. Pedrini, "Multi-scale gray level co-occurrence matrices for texture description," *Neurocomputing*, vol. 120, pp. 336–345, 2013.

[290] D. A. Clausi and M. E. Jernigan, "A fast method to determine co-occurrence texture features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 1, pp. 298–300, Jan 1998.

[291] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, Nov 1973.

[292] "Mit media lab. vistex texture database, (1995)." https://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html.

[293] Z. Song, S. Ali, and N. Bouguila, "Bayesian learning of infinite asymmetric gaussian mixture models for background subtraction," in *Image Analysis and Recognition*, F. Karray, A. Campilho, and A. Yu, Eds. Cham: Springer International Publishing, 2019, pp. 264–274.

[294] T. M. Nguyen, Q. M. J. Wu, D. Mukherjee, and H. Zhang, "Bounded asymmetric mixture model for medical image segmentation," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, 2013, pp. 1031–1035. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.2013.6637806

[295] T. Elguebaly and N. Bouguila, "Model-based approach for high-dimensional non-Gaussian visual data clustering and feature weighting," *Digital Signal Processing*, vol. 40, pp. 63 – 79, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1051200415000718

[296] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, Sep. 1999, pp. 1150–1157 vol.2.

[297] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vision*, vol. 42, no. 3, pp. 145–175, May 2001. [Online]. Available: https://doi.org/10.1023/A:1011139631724

[298] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, June 2005, pp. 524–531 vol. 2.

[299] M. Girolami, "An alternative perspective on adaptive independent component analysis algorithms," *Neural Computation*, vol. 10, no. 8, pp. 2103–2114, 1998.

Appendix A

BGMM

A.1 Estimation of $\hat{p}_j$

$$\frac{\partial \log[\Phi(\mathscr{X},Z,\Theta,\Lambda)]}{\partial p_j} = \frac{\partial}{\partial p_j}\sum_{i=1}^{N} Z_{ij}\left\{\log p_j + \log p(\vec{X}_i|\xi_j)\right\} + \frac{\partial}{\partial p_j}\Lambda\left(1-\sum_{j=1}^{K} p_j\right) \quad (A.1)$$

$$\frac{\partial \log[\Phi(\mathscr{X},Z,\Theta,\Lambda)]}{\partial p_j} = \frac{\partial}{\partial p_j}\sum_{i=1}^{N} Z_{ij}\log p_j + \frac{\partial}{\partial p_j}\Lambda\left(1-\sum_{j=1}^{K} p_j\right) = \frac{\sum_{i=1}^{N} Z_{ij}}{p_j} - \Lambda \quad (A.2)$$

$$\frac{\partial \log[\Phi(\mathscr{X},Z,\Theta,\Lambda)]}{\partial p_j} = 0 \quad \Rightarrow \quad p_j = \frac{\sum_{i=1}^{N} Z_{ij}}{\Lambda} \quad (A.3)$$

Taking the derivative of the log-likelihood with respect to Λ, we obtain

$$1-\sum_{j=1}^{K} p_j = 0 \quad \Rightarrow \quad \sum_{j=1}^{K} p_j = 1 \quad \Rightarrow \quad \sum_{j=1}^{K} p_j = \sum_{j=1}^{K}\frac{\sum_{i=1}^{N} Z_{ij}}{\Lambda} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{K} Z_{ij}}{\Lambda} = 1 \quad (A.4)$$

Since $\sum_{j=1}^{K} Z_{ij} = 1$, we obtain $\Lambda = N$, then p_j will become:

$$\hat{p}_j = \frac{\sum_{i=1}^{N} Z_{ij}}{\Lambda} = \frac{\sum_{i=1}^{N} Z_{ij}}{N} \quad (A.5)$$

A.2 Derivation of $\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \vec{\mu}_j}$

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \vec{\mu}_j} = \frac{\partial}{\partial \mu_{jd}}\sum_{i=1}^{N} Z_{ij}\left\{\log p_j + \log f(\vec{X}_i|\xi_j) + \log \mathrm{H}(\vec{X}_i|j) - \log\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right\} \quad (A.6)$$

$$\frac{\partial}{\partial \mu_{jd}} \sum_{i=1}^{N} Z_{ij} \log p_j = 0 \tag{A.7}$$

$$\frac{\partial}{\partial \mu_{jd}} \sum_{i=1}^{N} Z_{ij} \log \mathrm{H}(X_{id}|j) = 0 \tag{A.8}$$

$$\frac{\partial \mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta)}{\partial \vec{\mu}_j} = \sum_{i=1}^{N} \hat{Z}_{ij} \left\{ \Sigma_j^{-1}(\vec{X}_i - \vec{\mu}_j) - \frac{\int_{\partial_j} \Sigma_j^{-1} f(\vec{\mathrm{u}}|\xi_j)(\vec{\mathrm{u}} - \vec{\mu}_j) d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \right\} \tag{A.9}$$

A.3 Estimation of $\hat{\mu}_{jd}$

$$\frac{\partial \mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta)}{\partial \vec{\mu}_j} = 0 \tag{A.10}$$

$$\sum_{i=1}^{N} \hat{Z}_{ij} \left\{ \Sigma_j^{-1}(\vec{X}_i - \vec{\mu}_j) - \frac{\int_{\partial_j} \Sigma_j^{-1} f(\vec{\mathrm{u}}|\xi_j)(\vec{\mathrm{u}} - \vec{\mu}_j) d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \right\} = 0 \tag{A.11}$$

$$\hat{\vec{\mu}}_j = \frac{\sum_{i=1}^{N} \hat{Z}_{ij} \left\{ \vec{X}_i - \frac{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)(\vec{\mathrm{u}} - \vec{\mu}_j) d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \right\}}{\sum_{i=1}^{N} \hat{Z}_{ij}} \tag{A.12}$$

A.4 Derivation of $\frac{\partial \mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta)}{\partial \Sigma_j}$

$$\frac{\partial \mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta)}{\partial \Sigma_j} = \frac{\partial}{\partial \Sigma_j} \sum_{i=1}^{N} Z_{ij} \left\{ \log p_j + \log f(\vec{X}_i|\xi_j) + \log \mathrm{H}(\vec{X}_i|j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \tag{A.13}$$

$$\begin{aligned} \frac{\partial \mathscr{L}(\mathscr{X}, \mathscr{Z}|\Theta)}{\partial \Sigma_j} = \sum_{i=1}^{N} \hat{Z}_{ij} \Bigg\{ & -\frac{1}{2}\Sigma_j^{-1} + \frac{1}{2}(\vec{X}_i - \vec{\mu}_j)\Sigma_j^{-2}(\vec{X}_i - \vec{\mu}_j)^T \\ & - \frac{\int_{\partial_j} (-\frac{1}{2}\Sigma_j^{-1} + \frac{1}{2}(\vec{\mathrm{u}} - \vec{\mu}_j)\Sigma_j^{-2}(\vec{\mathrm{u}} - \vec{\mu}_j)^T) f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \Bigg\} \end{aligned} \tag{A.14}$$

A.5 Estimation of $\hat{\Sigma}_j$

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \Sigma_j} = 0 \tag{A.15}$$

$$\sum_{i=1}^{N} \hat{Z}_{ij} \left\{ -\frac{1}{2}\Sigma_j^{-1} + \frac{1}{2}(\vec{X}_i - \vec{\mu}_j)\Sigma_j^{-2}(\vec{X}_i - \vec{\mu}_j)^T - \frac{\int_{\partial_j}(-\frac{1}{2}\Sigma_j^{-1} + \frac{1}{2}(\vec{\mathrm{u}} - \vec{\mu}_j)\Sigma_j^{-2}(\vec{\mathrm{u}} - \vec{\mu}_j)^T) f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \right\} = 0 \tag{A.16}$$

$$\hat{\Sigma}_j = \frac{\sum_{i=1}^{N} \hat{Z}_{ij} \left\{ (\vec{X}_i - \vec{\mu}_j)(\vec{X}_i - \vec{\mu}_j)^T - \frac{\int_{\partial_j}(-\Sigma_j + (\vec{\mathrm{u}} - \vec{\mu}_j)(\vec{\mathrm{u}} - \vec{\mu}_j)^T) f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \right\}}{\sum_{i=1}^{N} \hat{Z}_{ij}} \tag{A.17}$$

A.6 Derivatives for MML

In this appendix, we compute the solutions for Eqs. (2.30 & 2.31) used for MML algorithm.

$$\frac{\partial^2 \mathscr{L}(\Theta, Z, \mathscr{X}_j)}{\partial \mu_j^2} = \sum_{i=l}^{l+n_j-1} \Sigma_j^{-1} \left\{ -1 + \frac{\Sigma_j^{-1} \left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)(\vec{\mathrm{u}} - \vec{\mu}_j) d\mathrm{u} \right)^2}{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right)^2} - \frac{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) \left((\vec{\mathrm{u}} - \vec{\mu}_j)\Sigma_j^{-1}(\vec{\mathrm{u}} - \vec{\mu}_j)^T - 1 \right) d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \right\} \tag{A.18}$$

$$\begin{aligned} \frac{\partial^2 \mathscr{L}(\Theta, Z, \mathscr{X}_j)}{\partial \Sigma_j^2} = \sum_{i=l}^{l+n_j-1} \Bigg\{ & \frac{1}{2}\Sigma_j^{-2} - (\vec{X} - \vec{\mu}_j)\Sigma_j{}^{-3}(\vec{X} - \vec{\mu}_j)^T \\ & + \frac{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) \{ -\frac{1}{2}\Sigma_j^{-1} + \frac{1}{2}(\vec{\mathrm{u}} - \vec{\mu}_j)\Sigma_j^{-2}(\vec{\mathrm{u}} - \vec{\mu}_j) \} d\mathrm{u} \right)^2}{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right)^2} \\ & - \frac{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) \left[\left(-\frac{1}{2}\Sigma_j^{-1} + \frac{1}{2}(\vec{\mathrm{u}} - \vec{\mu}_j)\Sigma_j^{-2}(\vec{\mathrm{u}} - \vec{\mu}_j)^T \right)^2 + \left(\frac{1}{2}\Sigma_j^{-2} + (\vec{\mathrm{u}} - \vec{\mu}_j)\Sigma_j^{-3}(\vec{\mathrm{u}} - \vec{\mu}_j)^T \right) \right] d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \Bigg\} \end{aligned} \tag{A.19}$$

Appendix B

BLMM

B.1 Derivation of $\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \mu_{jd}}$

For a particular mixture j and dimension d, the data log-likelihood is differentiated with respect to μ_{jd} as below.

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \mu_{jd}} = \frac{\partial}{\partial \mu_{jd}} \sum_{i=1}^{N} Z_{ij} \left\{ \log p_j + \log f(\vec{X}_i|\xi_j) + \log \mathrm{H}(\vec{X}_i|j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \tag{B.1}$$

$$\begin{aligned} \frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \mu_{jd}} &= \frac{\partial}{\partial \mu_{jd}} \sum_{i=1}^{N} Z_{ij} \left\{ \log f(\vec{X}_i|\xi_j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \\ &= \sum_{i=1}^{N} Z_{ij} \left\{ \left[\frac{(\mathrm{X}_{id} - \mu_{jd})}{b_{jd} \left| \mathrm{X}_{id} - \mu_{jd} \right|} \right] - \frac{\int_{\partial_j} \left(f(\mathrm{u}|\xi_j) \left[\frac{(\mathrm{u}-\mu_{jd})}{b_{jd}|\mathrm{u}-\mu_{jd}|} \right] \right) d\mathrm{u}}{\int_{\partial_j} f(\mathrm{u}|\xi_j) d\mathrm{u}} \right\} \end{aligned} \tag{B.2}$$

B.2 Estimation of $\hat{\mu}_{jd}$

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \mu_{jd}} = 0 \tag{B.3}$$

$$\sum_{i=1}^{N} Z_{ij} \left\{ \left[\frac{(\mathrm{X}_{id} - \mu_{jd})}{b_{jd} \left| \mathrm{X}_{id} - \mu_{jd} \right|} \right] - \frac{\int_{\partial_j} \left(f(\mathrm{u}|\xi_j) \left[\frac{(\mathrm{u}-\mu_{jd})}{b_{jd}|\mathrm{u}-\mu_{jd}|} \right] \right) d\mathrm{u}}{\int_{\partial_j} f(\mathrm{u}|\xi_j) d\mathrm{u}} \right\} = 0 \tag{B.4}$$

$$\hat{\mu}_{jd} = \frac{\sum_{i=1}^{N} Z_{ij} \left\{ \left[\frac{\mathrm{X}_{id}}{b_{jd}|\mathrm{X}_{id}-\mu_{jd}|} \right] - \frac{\int_{\partial_j} \left(f(\mathrm{u}|\xi_j) \left[\frac{(\mathrm{u}-\mu_{jd})}{b_{jd}|\mathrm{u}-\mu_{jd}|} \right] \right) d\mathrm{u}}{\int_{\partial_j} f(\mathrm{u}|\xi_j) d\mathrm{u}} \right\}}{\sum_{i=1}^{N} \left\{ \left[\frac{Z_{ij}}{b_{jd}|\mathrm{X}_{id}-\mu_{jd}|} \right] \right\}} \tag{B.5}$$

B.3 Derivation of $\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial b_{jd}}$

For a particular mixture j and dimension d, the data log-likelihood is differentiated with respect to b_{jd} as below.

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial b_{jd}} = \frac{\partial}{\partial b_{jd}} \sum_{i=1}^{N} Z_{ij} \left\{ \log p_j + \log f(\vec{X}_i|\xi_j) + \log \mathrm{H}(\vec{X}_i|j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \tag{B.6}$$

$$\begin{aligned} \frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial b_{jd}} &= \frac{\partial}{\partial b_{jd}} \sum_{i=1}^{N} Z_{ij} \left\{ \log f(\vec{X}_i|\xi_j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \\ &= \sum_{i=1}^{N} Z_{ij} \left\{ \left[\frac{-1}{b_{jd}} + \frac{|\mathrm{X}_{id}-\mu_{jd}|}{b_{jd}^2} \right] - \frac{\int_{\partial_j} \left(\frac{-1}{b_{jd}} f(\mathrm{u}|\xi_j) + \frac{|\mathrm{u}-\mu_{jd}|}{b_{jd}^2} f(\mathrm{u}|\xi_j) \right) d\mathrm{u}}{\int_{\partial_j} f(\mathrm{u}|\xi_j) d\mathrm{u}} \right\} \end{aligned} \tag{B.7}$$

B.4 Derivation of $\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial b_{jd}^2}$

The second order derivative can be computed from first order derivative as follows:

$$\begin{aligned} \frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial b_{jd}^2} &= \frac{\partial^2}{\partial b_{jd}^2} \sum_{i=1}^{N} Z_{ij} \left\{ \log f(\vec{X}_i|\xi_j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \\ &= \frac{\partial}{\partial b_{jd}} \sum_{i=1}^{N} Z_{ij} \left\{ \left[\frac{-1}{b_{jd}} + \frac{|\mathrm{X}_{id}-\mu_{jd}|}{b_{jd}^2} \right] - \frac{\int_{\partial_j} \left(\frac{-1}{b_{jd}} f(\mathrm{u}|\xi_j) + \frac{|\mathrm{u}-\mu_{jd}|}{b_{jd}^2} f(\mathrm{u}|\xi_j) \right) d\mathrm{u}}{\int_{\partial_j} f(\mathrm{u}|\xi_j) d\mathrm{u}} \right\} \\ &= \sum_{i=1}^{N} Z_{ij} \left\{ \left[\frac{1}{b_{jd}^2} - \frac{2|\mathrm{X}_{id}-\mu_{jd}|}{b_{jd}^3} \right] - \frac{\partial}{\partial b_{jd}} \left[\frac{\int_{\partial_j} \frac{-1}{b_{jd}} f(\mathrm{u}|\xi_j) d\mathrm{u}}{\int_{\partial_j} f(\mathrm{u}|\xi_j) d\mathrm{u}} \right] - \frac{\partial}{\partial b_{jd}} \left[\frac{\int_{\partial_j} \left(\frac{|\mathrm{u}-\mu_{jd}|}{b_{jd}^2} f(\mathrm{u}|\xi_j) \right) d\mathrm{u}}{\int_{\partial_j} f(\mathrm{u}|\xi_j) d\mathrm{u}} \right] \right\} \end{aligned} \tag{B.8}$$

$$\frac{\partial}{\partial b_{jd}}\left[\frac{\int_{\partial_j}\frac{-1}{b_{jd}}f(\mathrm{u}|\xi_j)d\mathrm{u}}{\int_{\partial_j}f(\mathrm{u}|\xi_j)d\mathrm{u}}\right]=\frac{\int_{\partial_j}\left(\frac{1}{b_{jd}^2}f(\mathrm{u}|\xi_j)-\frac{1}{b_{jd}}f(\mathrm{u}|\xi_j)\frac{(\mathrm{u}-\mu_{jd})}{b_{jd}|\mathrm{u}-\mu_{jd}|}\right)d\mathrm{u}}{\left(\int_{\partial_j}f(\mathrm{u}|\xi_j)d\mathrm{u}\right)} - \frac{\left(\int_{\partial_j}\frac{-1}{b_{jd}}f(\mathrm{u}|\xi_j)d\mathrm{u}\right)\int_{\partial_j}f(\mathrm{u}|\xi_j)\frac{(\mathrm{u}-\mu_{jd})}{b_{jd}|\mathrm{u}-\mu_{jd}|}d\mathrm{u}}{\left(\int_{\partial_j}f(\mathrm{u}|\xi_j)d\mathrm{u}\right)^2} \tag{B.9}$$

$$\frac{\partial}{\partial b_{jd}}\left[\frac{\int_{\partial_j}\left(\frac{|\mathrm{u}-\mu_{jd}|}{b_{jd}^2}f(\mathrm{u}|\xi_j)\right)d\mathrm{u}}{\int_{\partial_j}f(\mathrm{u}|\xi_j)d\mathrm{u}}\right]=\frac{\left(\int_{\partial_j}\frac{-2|\mathrm{u}-\mu_{jd}|}{b_{jd}^3}f(\mathrm{u}|\xi_j)d\mathrm{u}+\int_{\partial_j}\left(\frac{-|\mathrm{u}-\mu_{jd}|}{b_{jd}^3}f(\mathrm{u}|\xi_j)+\frac{|\mathrm{u}-\mu_{jd}|^2}{b_{jd}^4}f(\vec{\mathrm{u}}|\xi_j)\right)d\mathrm{u}\right)}{\left(\int_{\partial_j}f(\mathrm{u}|\xi_j)d\mathrm{u}\right)} - \frac{\left(\int_{\partial_j}\frac{|\mathrm{u}-\mu_{jd}|}{b_{jd}^2}f(\mathrm{u}|\xi_j)d\mathrm{u}\right)\int_{\partial_j}\left(\frac{-1}{b_{jd}}f(\mathrm{u}|\xi_j)+\frac{|\mathrm{u}-\mu_{jd}|}{b_{jd}^2}f(\mathrm{u}|\xi_j)\right)d\mathrm{u}}{\left(\int_{\partial_j}f(\mathrm{u}|\xi_j)d\mathrm{u}\right)^2} \tag{B.10}$$

Appendix C

ICA Mixture Model

C.1 Derivation of $\frac{\partial[\mathscr{L}(\Theta,\mathscr{Z},\mathscr{X})]}{\partial\mu_{jd}}$

$$\frac{\partial[\mathscr{L}(\Theta,\mathscr{Z},\mathscr{X})]}{\partial\mu_{jd}} = \frac{\partial}{\partial\mu_{jd}}\sum_{i=1}^{N} Z_{ij}\left\{\log p_j + \log f_{ggd}(X_{id}|\xi_j) + \log \mathrm{H}(X_{id}|j) - \log\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u}\right\} \tag{C.1}$$

$$\frac{\partial[\mathscr{L}(\Theta,\mathscr{Z},\mathscr{X})]}{\partial\mu_{jd}} = \frac{\partial}{\partial\mu_{jd}}\sum_{i=1}^{N} Z_{ij}\left\{\log f_{ggd}(X_{id}|\xi_j) - \log\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u}\right\} \tag{C.2}$$

$$\frac{\partial}{\partial\mu_{jd}}\sum_{i=1}^{N} Z_{ij}\log f_{ggd}(X_{id}|\xi_j) = A(\lambda_{jd})\frac{\lambda_{jd}}{\sigma_{jd}^{\lambda_{jd}}}\sum_{i=1}^{N} Z_{ij}\left[\left|X_{id}-\mu_{jd}\right|^{(\lambda_{jd}-2)}\left(X_{id}-\mu_{jd}\right)\right] \tag{C.3}$$

$$\frac{\partial}{\partial\mu_{jd}}\sum_{i=1}^{N} Z_{ij}\log\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u} = A(\lambda_{jd})\frac{\lambda_{jd}}{\sigma_{jd}^{\lambda_{jd}}}\sum_{i=1}^{N} Z_{ij}\left[\frac{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)\mathrm{sign}\left(\mathrm{u}-\mu_{jd}\right)\left|\mathrm{u}-\mu_{jd}\right|^{\lambda_{jd}-1}d\mathrm{u}}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u}}\right] \tag{C.4}$$

$$\begin{aligned}\frac{\partial[\mathscr{L}(\Theta,\mathscr{Z},\mathscr{X})]}{\partial\mu_{jd}} = A(\lambda_{jd})\frac{\lambda_{jd}}{\sigma_{jd}^{\lambda_{jd}}}\sum_{i=1}^{N} Z_{ij}\Bigg\{&\left[\left|X_{id}-\mu_{jd}\right|^{(\lambda_{jd}-2)}\left(X_{id}-\mu_{jd}\right)\right] \\ &-\left[\frac{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)\mathrm{sign}\left(\mathrm{u}-\mu_{jd}\right)\left|\mathrm{u}-\mu_{jd}\right|^{\lambda_{jd}-1}d\mathrm{u}}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u}}\right]\Bigg\}\end{aligned} \tag{C.5}$$

C.2 Estimation of $\hat{\mu}_{jd}$

$$A(\lambda_{jd})\frac{\lambda_{jd}}{\sigma_{jd}^{\lambda_{jd}}}\sum_{i=1}^{N}Z_{ij}\left\{\left[\left|X_{id}-\mu_{jd}\right|^{(\lambda_{jd}-2)}\left(X_{id}-\mu_{jd}\right)\right]-\left[\frac{\int_{\partial_j}f_{ggd}(\mathrm{u}|\xi_j)\mathrm{sign}\left(\mathrm{u}-\mu_{jd}\right)\left|\mathrm{u}-\mu_{jd}\right|^{\lambda_{jd}-1}d\mathrm{u}}{\int_{\partial_j}f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u}}\right]\right\}=0 \tag{C.6}$$

$$\begin{aligned}\hat{\mu}_{jd}=\frac{1}{\sum_{i=1}^{N}Z_{ij}\left|X_{id}-\mu_{jd}\right|^{(\lambda_{jd}-2)}}\sum_{i=1}^{N}Z_{ij}\Bigg\{&\left[\left|X_{id}-\mu_{jd}\right|^{(\lambda_{jd}-2)}X_{id}\right]\\&-\left[\frac{\int_{\partial_j}f_{ggd}(\mathrm{u}|\xi_j)\mathrm{sign}\left(\mathrm{u}-\mu_{jd}\right)\left|\mathrm{u}-\mu_{jd}\right|^{\lambda_{jd}-1}d\mathrm{u}}{\int_{\partial_j}f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u}}\right]\Bigg\}\end{aligned} \tag{C.7}$$

C.3 Derivation of $\frac{\partial[\mathscr{L}(\Theta,\mathscr{Z},\mathscr{X})]}{\partial\sigma_{jd}}$

$$\frac{\partial[\mathscr{L}(\Theta,\mathscr{Z},\mathscr{X})]}{\partial\sigma_{jd}}=\frac{\partial}{\partial\sigma_{jd}}\sum_{i=1}^{N}Z_{ij}\left\{\log p_j+\log f_{ggd}(X_{id}|\xi_j)+\log\mathrm{H}(X_{id}|j)-\log\int_{\partial_j}f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u}\right\} \tag{C.8}$$

$$\frac{\partial[\mathscr{L}(\Theta,\mathscr{Z},\mathscr{X})]}{\partial\sigma_{jd}}=\frac{\partial}{\partial\sigma_{jd}}\sum_{i=1}^{N}Z_{ij}\left\{\log f_{ggd}(X_{id}|\xi_j)-\log\int_{\partial_j}f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u}\right\} \tag{C.9}$$

$$\frac{\partial}{\partial\sigma_{jd}}\sum_{i=1}^{N}Z_{ij}\log f_{ggd}(X_{id}|\xi_j)=\frac{1}{\sigma_{jd}}\sum_{i=1}^{N}Z_{ij}\left[-1+A(\lambda_{jd})\left|X_{id}-\mu_{jd}\right|^{\lambda_{jd}}\lambda_{jd}(\sigma_{jd})^{-\lambda_{jd}}\right] \tag{C.10}$$

$$\frac{\partial}{\partial\sigma_{jd}}\sum_{i=1}^{N}Z_{ij}\log\int_{\partial_j}f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u}=\frac{1}{\sigma_{jd}}\sum_{i=1}^{N}Z_{ij}\left[\frac{\int_{\partial_j}f_{ggd}(\mathrm{u}|\xi_j)\left\{-1+A(\lambda_{jd})\left|X_{id}-\mu_{jd}\right|^{\lambda_{jd}}\lambda_{jd}(\sigma_{jd})^{-\lambda_{jd}}\right\}d\mathrm{u}}{\int_{\partial_j}f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u}}\right] \tag{C.11}$$

$$\begin{aligned}\frac{\partial[\mathscr{L}(\Theta,\mathscr{Z},\mathscr{X})]}{\partial\sigma_{jd}}=\frac{1}{\sigma_{jd}}\sum_{i=1}^{N}Z_{ij}\Bigg\{&\left[-1+A(\lambda_{jd})\left|X_{id}-\mu_{jd}\right|^{\lambda_{jd}}\lambda_{jd}(\sigma_{jd})^{-\lambda_{jd}}\right]\\&-\left[\frac{\int_{\partial_j}f_{ggd}(\mathrm{u}|\xi_j)\left\{-1+A(\lambda_{jd})\left|X_{id}-\mu_{jd}\right|^{\lambda_{jd}}\lambda_{jd}(\sigma_{jd})^{-\lambda_{jd}}\right\}d\mathrm{u}}{\int_{\partial_j}f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u}}\right]\Bigg\}\end{aligned} \tag{C.12}$$

C.4 Estimation of $\hat{\sigma}_{jd}$

$$\frac{1}{\sigma_{jd}}\sum_{i=1}^{N} Z_{ij}\left\{\left[-1+A(\lambda_{jd})\left|X_{id}-\mu_{jd}\right|^{\lambda_{jd}}\lambda_{jd}(\sigma_{jd})^{-\lambda_{jd}}\right] - \left[\frac{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)\left\{-1+A(\lambda_{jd})\left|X_{id}-\mu_{jd}\right|^{\lambda_{jd}}\lambda_{jd}(\sigma_{jd})^{-\lambda_{jd}}\right\}d\mathrm{u}}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u}}\right]\right\} = 0 \tag{C.13}$$

$$\hat{\sigma}_{jd} = \left(\frac{\sum_{i=1}^{N} Z_{ij}\left[A(\lambda_{jd})\left|X_{id}-\mu_{jd}\right|^{\lambda_{jd}}\lambda_{jd}\right]}{\sum_{i=1}^{N} Z_{ij}\left\{1+\left[\frac{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)\left\{-1+A(\lambda_{jd})\left|X_{id}-\mu_{jd}\right|^{\lambda_{jd}}\lambda_{jd}(\sigma_{jd})^{-\lambda_{jd}}\right\}d\mathrm{u}}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)d\mathrm{u}}\right]\right\}}\right)^{1/\lambda_{jd}} \tag{C.14}$$

C.5 Estimation of Shape Parameter $\hat{\lambda}_{jd}$ with Gradient Ascent

For the estimation of parameters in ICA mixture model, unit variance and zero mean is assumed. For the purpose of estimation of shape parameter, same assumption is adopted and the problem will become the estimation of shape parameter from the data. The Eq. (4.4) with the assumption of zero mean and unit variance will become:

$$f_{ggd}(\vec{X}_i|\xi_j) = \prod_{d=1}^{D}\frac{\lambda_{jd}\sqrt{\Gamma(3/\lambda_{jd})}}{2\Gamma(1/\lambda_{jd})\sqrt{\Gamma(1/\lambda_{jd})}}\exp\left(-A(\lambda_{jd})\left|X_{id}\right|^{\lambda_{jd}}\right),\ \text{with } A(\lambda_{jd}) = \left[\frac{\Gamma(3/\lambda_{jd})}{\Gamma(1/\lambda_{jd})}\right]^{\lambda_{jd}/2} \tag{C.15}$$

The term $\frac{\partial}{\partial\lambda_{jd}}\log p(X_{id}|\xi_j)$ in the estimation of shape parameter using gradient ascent in an ICA mixture model can be computed as below.

$$\frac{\partial}{\partial\lambda_{jd}}\log p(X_{id}|\xi_j) = \frac{\partial}{\partial\lambda_{jd}}\log\left[\frac{f_{ggd}(X_{id}|\xi_j)\mathrm{H}(X_{id}|j)}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)dX}\right] \tag{C.16}$$

$$\frac{\partial}{\partial\lambda_{jd}}\log p(X_{id}|\xi_j) = \frac{\partial}{\partial\lambda_{jd}}\log f_{ggd}(X_{id}|\xi_j) - \frac{\partial}{\partial\lambda_{j}}\log\int_{\partial_j} f_{ggd}(\mathrm{u}|\xi_j)dX \tag{C.17}$$

$$\begin{aligned} h(X_{id}|\xi_j) &= \frac{\partial}{\partial\lambda_{jd}}\log f_{ggd}(X_{id}|\xi_j) \\ &= \left[\frac{1}{\lambda_{jd}} + \frac{3}{2\lambda_{jd}}\left[\Psi(1/\lambda_{jd}) - \Psi(3/\lambda_{jd})\right]\right] - A(\lambda_{jd})\left|X_{id}\right|^{\lambda_{jd}}\log\left|X_{id}\right| \\ &\quad - A(\lambda_{jd})\left(\frac{1}{2}\log\frac{\Gamma(3/\lambda_{jd})}{\Gamma(1/\lambda_{jd})} + \frac{1}{2\lambda_{jd}}\left[\Psi(1/\lambda_{jd}) - 3\Psi(3/\lambda_{jd})\right]\right)\left|X_{id}\right|^{\lambda_{jd}} \end{aligned} \tag{C.18}$$

$$\frac{\partial}{\partial \lambda_{jd}} \log \int_{\partial_j} f_{ggd}(X_d|\xi_j)dX = \frac{\frac{\partial}{\partial \lambda_{jd}} \int_{\partial_j} f_{ggd}(X_d|\xi_j)dX}{\int_{\partial_j} f_{ggd}(X_d|\xi_j)dX} = \frac{\int_{\partial_j} f_{ggd}(X_d|\xi_j)h(X_d|\xi_j)dX}{\int_{\partial_j} f_{ggd}(X_d|\xi_j)dX} \tag{C.19}$$

The term $\int_{\partial_j} f_{ggd}(X_d|\xi_j)h(X_d|\xi_j)dX$ can be approximated similar to Eq. (4.12).

$$\int_{\partial_j} f_{ggd}(X_d|\xi_j)h(X_d|\xi_j)dX \approx \frac{1}{M}\sum_{m=1}^{M} h(s_{j_{md}}|\xi_j)\mathrm{H}(s_{j_{md}}|j) \tag{C.20}$$

$$\begin{aligned}\frac{\partial}{\partial \lambda_{jd}} \log p(X_{id}|\xi_j) = &\left[\frac{1}{\lambda_{jd}} + \frac{3}{2\lambda_{jd}}\left[\Psi(1/\lambda_{jd}) - \Psi(3/\lambda_{jd})\right]\right] - A(\lambda_{jd})\,|X_{id}|^{\lambda_{jd}} \log|X_{id}| \\ &- A(\lambda_{jd})\left(\frac{1}{2}\log\frac{\Gamma(3/\lambda_{jd})}{\Gamma(1/\lambda_{jd})} + \frac{1}{2\lambda_{jd}}\left[\Psi(1/\lambda_{jd}) - 3\Psi(3/\lambda_{jd})\right]\right)|X_{id}|^{\lambda_{jd}} \\ &- \frac{\sum_{m=1}^{M} h(s_{j_{md}}|\xi_j)\mathrm{H}(s_{j_{md}}|j)}{\sum_{m=1}^{M}\mathrm{H}(s_{j_{md}}|j)}\end{aligned} \tag{C.21}$$

$$\hat{\lambda}_{jd} = \lambda_{jd} + \alpha\left(p(j|\vec{X}_i)\frac{\partial}{\partial \lambda_{jd}}\log p(X_{id}|\xi_j)\right) \tag{C.22}$$

C.6 Independent Component Analysis Learning Algorithm

The derivative of $\log p(\vec{X}_i|\xi_j)$ in Section 4.2.1.2 can be computed using ICA [196, 245, 299]. Assume that s is an M-dimensional zero mean vector that has mutually independent components and $\vec{s}$ corresponds to M independent scaler-valued sources which is expressed as $\vec{\mathrm{s}} = [s_1, ..., s_M]^T$. A data vector $\vec{X}_i = [X_{i1}, ..., X_{iD}]^T$ is observed at each time point i, such that $\vec{X}_i = \mathrm{A}\vec{\mathrm{s}}_i$, where A is $D \times M$ scalar matrix. In the proposed algorithm we shall consider the case where, the number of sources is equal to the number of sensors $D = M$. The goal of ICA is to estimate the a linear transformation W of the dependent sensor signal X that makes the output u as independent as possible such that u is an estimate of the sources as: $\mathrm{u}_i = \mathrm{W}\vec{X}_i = \mathrm{WA}\vec{\mathrm{s}}_i$. The sources can be recovered exactly when the W is the inverse of A up to a scale and permutation level. The probability density function of the observations X can be represented as: $p(X) = |\det(\mathrm{W})|p(\mathrm{u})$, where $p(\mathrm{u})$ is the hypothesized distribution of $p(\vec{s})$ The log-likelihood of the above probability density function is given by:

$$L(\mathrm{u}, \mathrm{W}) = \log(\det(\mathrm{W})) + \log(p(\mathrm{u})) \tag{C.23}$$

By maximizing the log-likelihood with respect to W , learning algorithm for W can be determined as:

$$\Delta \mathrm{W} \propto \left[(\mathrm{W}^T)^{-1} - \phi(\mathrm{u})X^T\right] \tag{C.24}$$

where

$$\varphi(\mathrm{u}) = \left[-\frac{\frac{\partial p(\mathrm{u})}{\partial(\mathrm{u})}}{p(\mathrm{u})}\right] = \left[-\frac{\frac{\partial p(u_1)}{\partial(u_1)}}{p(u_1)}, \ldots, -\frac{\frac{\partial p(u_N)}{\partial(u_N)}}{p(u_N)}\right]^T \tag{C.25}$$

An efficient way to maximize the log-likelihood is to follow the gradient ascent.

$$\Delta \mathrm{W} \propto \frac{\partial L(\mathrm{u},\mathrm{W})}{\partial \mathrm{W}} \mathrm{W}^T \mathrm{W} = \left[\mathrm{I} - \phi(\mathrm{u})\mathrm{u}^T\right] \mathrm{W} \tag{C.26}$$

If we choose $g(u)$ to be a logistic function ($g(u) = \tanh(\mathrm{u})$)

$$\varphi(\mathrm{u}) = \left[-\frac{\frac{\partial p(\mathrm{u})}{\partial(\mathrm{u})}}{p(\mathrm{u})}\right], \quad p(\mathrm{u}) = \frac{\partial g}{\partial \mathrm{u}} = \frac{\partial}{\partial \mathrm{u}}\tanh(\mathrm{u}) = 1 - \tanh(\mathrm{u})^2 \tag{C.27}$$

$$\frac{\partial}{\partial \mathrm{u}} p(\mathrm{u}) = \frac{\partial^2}{\partial \mathrm{u}^2}\tanh(\mathrm{u}) = -2\tanh(\mathrm{u})(1 - \tanh(\mathrm{u})^2) \tag{C.28}$$

$$\varphi(\mathrm{u}) = \frac{2\tanh(\mathrm{u})(1 - \tanh(\mathrm{u})^2)}{(1 - \tanh(\mathrm{u})^2)} = 2\tanh(\mathrm{u}) \tag{C.29}$$

The learning rule for ICA will become:

$$\Delta \mathrm{W} \propto \mathrm{W}[\mathrm{I} - 2\tanh(\mathrm{u})\mathrm{u}^T] \tag{C.30}$$

Appendix D

BAGMM

D.1 Derivation of $\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \mu_{jd}}$

For a particular mixture j and dimension d, the data log-likelihood is differentiated with respect to μ_{jd} as below.

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \mu_{jd}} = \frac{\partial}{\partial \mu_{jd}} \sum_{i=1}^{N} Z_{ij} \left\{ \log p_j + \log f(\vec{X}_i|\xi_j) + \log \mathrm{H}(\vec{X}_i|j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \quad \text{(D.1)}$$

$$\begin{aligned}
\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \mu_{jd}} &= \frac{\partial}{\partial \mu_{jd}} \sum_{i=1}^{N} Z_{ij} \left\{ \log f(\vec{X}_i|\xi_j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \quad \text{(D.2)} \\
&= \sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} Z_{ij} \left[\frac{(\mathrm{X}_{id}-\mu_{jd})}{\sigma^2_{l_{jd}}} \right] + \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} Z_{ij} \left[\frac{(\mathrm{X}_{id}-\mu_{jd})}{\sigma^2_{r_{jd}}} \right] \\
&\quad - \sum_{i=1,\mathrm{x}_{d}<\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^2_{l_{jd}}} \times \left\{ \frac{\int_{\partial_j} \frac{2}{\sqrt{2\pi}(\sigma_{l_{jd}}+\sigma_{r_{jd}})} \left(\exp\left[-\frac{(\mathrm{u}-\mu_{jd})^2}{2\sigma^2_{l_{jd}}} \right] \right) (\mathrm{u}-\mu_{jd}) d\mathrm{u}}{\int_{\partial_j} f(\mathrm{u}|\xi_j) d\mathrm{u}} \right\} \\
&\quad - \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^2_{r_{jd}}} \times \left\{ \frac{\int_{\partial_j} \frac{2}{\sqrt{2\pi}(\sigma_{l_{jd}}+\sigma_{r_{jd}})} \left(\exp\left[-\frac{(\mathrm{u}-\mu_{jd})^2}{2\sigma^2_{r_{jd}}} \right] \right) (\mathrm{u}-\mu_{jd}) d\mathrm{u}}{\int_{\partial_j} f(\mathrm{u}|\xi_j) d\mathrm{u}} \right\}
\end{aligned}$$

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \mu_{jd}} = \sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} Z_{ij} \left[\frac{(\mathrm{X}_{id}-\mu_{jd})}{\sigma^2_{l_{jd}}} \right] + \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} Z_{ij} \left[\frac{(\mathrm{X}_{id}-\mu_{jd})}{\sigma^2_{r_{jd}}} \right] \quad \text{(D.3)}$$

$$- \sum_{i=1,\mathrm{x}_d<\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^2_{l_{jd}}} \times \left\{ \frac{\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})d\mathrm{u}}{\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)d\mathrm{u}} \right\}$$
$$- \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^2_{r_{jd}}} \times \left\{ \frac{\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})d\mathrm{u}}{\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)d\mathrm{u}} \right\}$$

D.2 Estimation of $\hat{\mu}_{jd}$

D.2.1 For the case $X_{id} < \mu_{jd}$, Estimation of $\hat{\mu}_{jd}$

$$\sum_{i=1,X_{id}<\mu_{jd}}^{N} Z_{ij} \left[\frac{(\mathrm{X}_{id}-\mu_{jd})}{\sigma^2_{l_{jd}}} \right] - \sum_{i=1,\mathrm{x}_d<\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^2_{l_{jd}}} \times \left\{ \frac{\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})d\mathrm{u}}{\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)d\mathrm{u}} \right\} = 0 \tag{D.4}$$

$$\left\{\hat{\mu}_{jd}\right\}_{\mathrm{X}_{id}<\mu_{jd}} = \frac{\sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} Z_{ij} \left\{ \mathrm{X}_{id} - \frac{\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})d\mathrm{x}}{\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)d\mathrm{u}} \right\}}{\sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} Z_{ij}} \tag{D.5}$$

D.2.2 For the case $X_{id} \geq \mu_{jd}$, Estimation of $\hat{\mu}_{jd}$

$$\sum_{i=1,X_{id}\geq\mu_{jd}}^{N} Z_{ij} \left[\frac{(\mathrm{X}_{id}-\mu_{jd})}{\sigma^2_{r_{jd}}} \right] - \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^2_{r_{jd}}} \times \left\{ \frac{\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})d\mathrm{u}}{\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)d\mathrm{u}} \right\} = 0 \tag{D.6}$$

$$\left\{\hat{\mu}_{jd}\right\}_{\mathrm{X}_{id}\geq\mu_{jd}} = \frac{\sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} Z_{ij} \left\{ \mathrm{X}_{id} - \frac{\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})d\mathrm{x}}{\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)d\mathrm{u}} \right\}}{\sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} Z_{ij}} \tag{D.7}$$

For $X_{id} < \mu_{jd}$ and $X_{id} \geq \mu_{jd}$, the derivation of $\hat{\mu}_{jd}$ can be generalized as:

$$\hat{\mu}_{jd} = \frac{\sum_{i=1}^{N} Z_{ij} \left\{ \mathrm{X}_{id} - \frac{\int_{\partial_j} f(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})d\mathrm{x}}{\int_{\partial_j} f(\mathrm{u}|\xi_j)d\mathrm{u}} \right\}}{\sum_{i=1}^{N} Z_{ij}} \tag{D.8}$$

D.3 Derivation of $\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma_{l_{jd}}}$

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma_{l_{jd}}} = \frac{\partial}{\partial \sigma_{l_{jd}}} \sum_{i=1}^{N} Z_{ij} \left\{ \log p_j + \log f(\vec{X}_i|\xi_j) + \log \mathrm{H}(\vec{X}_i|j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \quad \text{(D.9)}$$

$$\begin{aligned} \frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma_{l_{jd}}} &= \frac{\partial}{\partial \sigma_{l_{jd}}} \sum_{i=1}^{N} Z_{ij} \left\{ \log f(\vec{X}_i|\xi_j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \quad \text{(D.10)} \\ &= \sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} Z_{ij} \left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma_{l_{jd}}^3} \right) - \sum_{i=1,\mathrm{x}_d<\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma_{l_{jd}}^3} \left\{ \frac{\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^2 d\mathrm{u}}{\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j) d\mathrm{u}} \right\} \end{aligned}$$

D.4 Derivation of $\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma^2{}_{l_{jd}}}$

$$\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma^2{}_{l_{jd}}} = \frac{\partial^2}{\partial \sigma^2{}_{l_{jd}}} \sum_{i=1}^{N} Z_{ij} \left\{ \log f(\vec{X}_i|\xi_j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \quad \text{(D.11)}$$

$$\begin{aligned} \frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma^2{}_{l_{jd}}} &= -3 \sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} Z_{ij} \left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma_{l_{jd}}^4} \right) \quad \text{(D.12)} \\ &- \sum_{i=1,\mathrm{x}_d<\mu_{jd}}^{N} Z_{ij} \left(\frac{-2}{\sigma_{l_{jd}}^3(\sigma_{l_{jd}}+\sigma_{r_{jd}})} \right) \left\{ \frac{\left(\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^2 d\mathrm{u} \right)}{\left(\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j) d\mathrm{u} \right)} \right\} \\ &- \sum_{i=1,\mathrm{x}_d<\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma_{l_{jd}}^6} \left\{ \frac{\left(\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^4 d\mathrm{u} \right)}{\left(\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j) d\mathrm{u} \right)} \right\} - \sum_{i=1,\mathrm{x}_d<\mu_{jd}}^{N} \frac{-3\, Z_{ij}}{\sigma_{l_{jd}}^4} \left\{ \frac{\left(\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^2 d\mathrm{u} \right)}{\left(\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j) d\mathrm{u} \right)} \right\} \\ &- \sum_{i=1,\mathrm{x}_d<\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma_{l_{jd}}^6} \left\{ \frac{\left(\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^2 d\mathrm{u} \right)^2}{\left(\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j) d\mathrm{u} \right)^2} \right\} \end{aligned}$$

Similar to the approximations for first order derivative in Section 5.2.3.2, second order derivative

can be approximated as follows:

$$\begin{aligned}
\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial {\sigma^2}_{l_{jd}}} &= -3 \sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} Z_{ij} \left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma^4_{l_{jd}}} \right) \\
&- \sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} Z_{ij} \left(\frac{-2}{\sigma^3_{l_{jd}}(\sigma_{l_{jd}}+\sigma_{r_{jd}})} \right) \left\{ \frac{\frac{1}{M}\sum_{m=1}^{M}(\mathrm{l}_{m_{jd}}-\mu_{jd})^2 \mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j)}{\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j)} \right\} \\
&- \sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^6_{l_{jd}}} \left\{ \frac{\frac{1}{M}\sum_{m=1}^{M}(\mathrm{l}_{m_{jd}}-\mu_{jd})^4 \mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j)}{\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j)} \right\} \\
&- \sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} \frac{-3\,Z_{ij}}{\sigma^4_{l_{jd}}} \left\{ \frac{\frac{1}{M}\sum_{m=1}^{M}(\mathrm{l}_{m_{jd}}-\mu_{jd})^2 \mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j)}{\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j)} \right\} \\
&- \sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^6_{l_{jd}}} \left\{ \frac{\left(\frac{1}{M}\sum_{m=1}^{M}(\mathrm{l}_{m_{jd}}-\mu_{jd})^2 \mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j)\right)^2}{(\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j))^2} \right\}
\end{aligned} \tag{D.13}$$

D.5 Derivation of $\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma_{r_{jd}}}$

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma_{r_{jd}}} = \frac{\partial}{\partial \sigma_{r_{jd}}} \sum_{i=1}^{N} Z_{ij} \left\{ \log p_j + \log f(\vec{X}_i|\xi_j) + \log \mathrm{H}(\vec{X}_i|j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \tag{D.14}$$

$$\begin{aligned}
\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma_{r_{jd}}} &= \frac{\partial}{\partial \sigma_{r_{jd}}} \sum_{i=1}^{N} Z_{ij} \left\{ \log f(\vec{X}_i|\xi_j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \qquad \text{(D.15)} \\
&= \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} Z_{ij} \left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma^3_{r_{jd}}} \right) - \sum_{i=1,\mathrm{x}_d\geq\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^3_{r_{jd}}} \left\{ \frac{\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^2 d\mathrm{u}}{\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j) d\mathrm{u}} \right\}
\end{aligned}$$

D.6 Derivation of $\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial {\sigma^2}_{r_{jd}}}$

$$\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial {\sigma^2}_{r_{jd}}} = \frac{\partial^2}{\partial {\sigma^2}_{r_{jd}}} \sum_{i=1}^{N} Z_{ij} \left\{ \log f(\vec{X}_i|\xi_j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \tag{D.16}$$

$$\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial {\sigma^2}_{r_{jd}}} = -3 \sum_{i=1,X_{id}\geq\mu_{jd}}^{N} Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma^4_{r_{jd}}}\right) \tag{D.17}$$
$$- \sum_{i=1,\mathrm{x}_d\geq\mu_{jd}}^{N} Z_{ij}\left(\frac{-2}{\sigma^3_{r_{jd}}(\sigma_{l_{jd}}+\sigma_{r_{jd}})}\right)\left\{\frac{\left(\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^2 d\mathrm{u}\right)}{(\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)d\mathrm{u})}\right\}$$
$$- \sum_{i=1,\mathrm{x}_d\geq\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^6_{r_{jd}}}\left\{\frac{\left(\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^4 d\mathrm{u}\right)}{(\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)d\mathrm{u})}\right\} - \sum_{i=1,\mathrm{x}_d\geq\mu_{jd}}^{N} \frac{-3\,Z_{ij}}{\sigma^4_{r_{jd}}}\left\{\frac{\left(\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^2 d\mathrm{u}\right)}{(\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)d\mathrm{u})}\right\}$$
$$- \sum_{i=1,\mathrm{x}_d\geq\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^6_{r_{jd}}}\left\{\frac{\left(\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^2 d\mathrm{u}\right)^2}{(\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j)d\mathrm{u})^2}\right\}$$

Similar to the approximations for first order derivative in Section 5.2.3.3, second order derivative can be approximated as follows:

$$\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial {\sigma^2}_{r_{jd}}} = -3 \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma^4_{r_{jd}}}\right)$$
$$- \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} Z_{ij}\left(\frac{-2}{\sigma^3_{r_{jd}}(\sigma_{l_{jd}}+\sigma_{r_{jd}})}\right)\left\{\frac{\frac{1}{M}\sum_{m=1}^{M}(\mathrm{r}_{m_{jd}}-\mu_{jd})^2\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j)}{\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j)}\right\}$$
$$- \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^6_{r_{jd}}}\left\{\frac{\frac{1}{M}\sum_{m=1}^{M}(\mathrm{r}_{m_{jd}}-\mu_{jd})^4\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j)}{\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j)}\right\} \tag{D.18}$$
$$- \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} \frac{-3\,Z_{ij}}{\sigma^4_{r_{jd}}}\left\{\frac{\frac{1}{M}\sum_{m=1}^{M}(\mathrm{r}_{m_{jd}}-\mu_{jd})^2\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j)}{\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j)}\right\}$$
$$- \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} \frac{Z_{ij}}{\sigma^6_{r_{jd}}}\left\{\frac{\left(\frac{1}{M}\sum_{m=1}^{M}(\mathrm{r}_{m_{jd}}-\mu_{jd})^2\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j)\right)^2}{(\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j))^2}\right\}$$

Appendix E

BAGGMM

E.1 Derivation of $\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \mu_{jd}}$

For a particular mixture j and dimension d, the data log-likelihood is differentiated with respect to μ_{jd} as below.

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \mu_{jd}} = \frac{\partial}{\partial \mu_{jd}} \sum_{i=1}^{N} Z_{ij} \left\{ \log p_j + \log f(\vec{X}_i|\xi_j) + \log \mathrm{H}(\vec{X}_i|j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \quad \text{(E.1)}$$

$$\begin{aligned}
\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \mu_{jd}} &= \frac{\partial}{\partial \mu_{jd}} \sum_{i=1}^{N} Z_{ij} \left\{ \log f(\vec{X}_i|\xi_j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \quad \text{(E.2)} \\
&= A(\lambda_{jd})\lambda_{jd} \left[\sum_{i=1, X_{id} \geq \mu_{jd}}^{N} Z_{ij} \frac{\left(X_{id} - \mu_{jd}\right)^{\lambda_{jd}-1}}{\sigma_{r_{jd}}^{\lambda_{jd}}} - \sum_{i=1, X_{id} < \mu_{jd}}^{N} Z_{ij} \frac{\left(\mu_{jd} - X_{id}\right)^{\lambda_{jd}-1}}{\sigma_{l_{jd}}^{\lambda_{jd}}} \right. \\
&\left. - \sum_{i=1, X_{id} \geq \mu_{jd}}^{N} Z_{ij} \left\{ \frac{\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j) \frac{(\mathrm{u}-\mu_{jd})^{\lambda_{jd}-1}}{\sigma_{r_{jd}}^{\lambda_{jd}}} d\mathrm{u}}{\int_{\partial_j} \mathrm{g}_2(\mathrm{u}|\xi_j) d\mathrm{u}} \right\} + \sum_{i=1, X_{id} < \mu_{jd}}^{N} Z_{ij} \left\{ \frac{\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j) \frac{(\mu_{jd}-\mathrm{u})^{\lambda_{jd}-1}}{\sigma_{l_{jd}}^{\lambda_{jd}}} d\mathrm{u}}{\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j) d\mathrm{x}} \right\} \right]
\end{aligned}$$

E.2 Derivation of $\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial {\mu^2}_{jd}}$

$$\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial {\mu^2}_{jd}} = \frac{\partial^2}{\partial {\mu^2}_{jd}} \sum_{i=1}^{N} Z_{ij} \left\{ \log f(\vec{X}_i|\xi_j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \quad \text{(E.3)}$$

$$= A(\lambda_{jd})\lambda_{jd}(\lambda_{jd}-1)\left[-\sum_{i=1,X_{ik}<\mu_{jd}}^{N}\hat{Z}_{ij}\frac{(\mu_{jd}-X_{ik})^{\lambda_{jd}-2}}{\sigma_{l_{jd}}^{\lambda_{jd}}}-\sum_{i=1,X_{ik}\geq\mu_{jd}}^{N}\hat{Z}_{ij}\frac{(X_{ik}-\mu_{jd})^{\lambda_{jd}-2}}{\sigma_{r_{jd}}^{\lambda_{jd}}}\right]$$

$$-A(\lambda_{jd})\frac{\lambda_{jd}}{\sigma_{jd}^{\lambda_{jd}}}\left[\sum_{i=1,X_{id}<\mu_{jd}}^{N}Z_{ij}\left\{\frac{\int_{\partial_j}-A(\lambda_{jd})\mathrm{g}_1(\mathrm{u}|\xi_j)\frac{(\mu_{jd}-\mathrm{u})^{2(\lambda_{jd}-1)}}{\sigma_{l_{jd}}^{\lambda_{jd}}}d\mathrm{u}}{\int_{\partial_j}\mathrm{g}_1(\mathrm{u}|\xi_j)d\mathrm{u}}+\frac{\int_{\partial_j}(\lambda_{jd}-1)\mathrm{g}_1(\mathrm{u}|\xi_j)(\mu_{jd}-\mathrm{u})^{\lambda_{jd}-2}d\mathrm{u}}{\int_{\partial_j}\mathrm{g}_1(\mathrm{u}|\xi_j)d\mathrm{u}}\right\}\right.$$

$$\left.+A(\lambda_{jd})\frac{\lambda_{jd}}{\sigma_{jd}^{\lambda_{jd}}}\left(\frac{\int_{\partial_j}\mathrm{g}_1(\mathrm{u}|\xi_j)(\mu_{jd}-\mathrm{u})^{\lambda_{jd}-1}d\mathrm{u}}{\int_{\partial_j}\mathrm{g}_1(\mathrm{u}|\xi_j)d\mathrm{u}}\right)^2\right]$$

$$+A(\lambda_{jd})\frac{\lambda_{jd}}{\sigma_{jd}^{\lambda_{jd}}}\left[\sum_{i=1,X_{id}\geq\mu_{jd}}^{N}\hat{Z}_{ij}\left\{\frac{\int_{\partial_j}A(\lambda_{jd})\mathrm{g}_2(\mathrm{u}|\xi_j)\frac{(\mathrm{u}-\mu_{jd})^{2(\lambda_{jd}-1)}}{\sigma_{r_{jd}}^{\lambda_{jd}}}d\mathrm{u}}{\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)d\mathrm{u}}+\frac{\int_{\partial_j}(\lambda_{jd}-1)\mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^{\lambda_{jd}-2}d\mathrm{u}}{\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)d\mathrm{u}}\right\}\right.$$

$$\left.+A(\lambda_{jd})\frac{\lambda_{jd}}{\sigma_{jd}^{\lambda_{jd}}}\left(\frac{\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^{\lambda_{jd}-1}d\mathrm{u}}{\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)d\mathrm{u}}\right)^2\right]$$

E.3 Derivation of $\frac{\partial\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial\sigma_{l_{jd}}}$

$$\frac{\partial\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial\sigma_{l_{jd}}}=\frac{\partial}{\partial\sigma_{l_{jd}}}\sum_{i=1}^{N}Z_{ij}\left\{\log p_j+\log f(\vec{X}_i|\xi_j)+\log\mathrm{H}(\vec{X}_i|j)-\log\int_{\partial_j}f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right\}\quad\text{(E.4)}$$

$$\frac{\partial\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial\sigma_{l_{jd}}}=\sum_{i=1,X_{id}<\mu_{jd}}^{N}\hat{Z}_{ij}\frac{A(\lambda_{jd})\lambda_{jd}}{\sigma_{l_{jd}}}\left(\frac{\mu_{jd}-X_{id}}{\sigma_{l_{jd}}}\right)^{\lambda_{jd}}\quad\text{(E.5)}$$

$$\sum_{i=1,X_{id}<\mu_{jd}}^{N}\hat{Z}_{ij}\left\{\frac{\int_{\partial_j}\mathrm{g}_1(\mu_j|\xi_j)\frac{A(\lambda_{jd})\lambda_{jd}(\mu_{jd}-\mathrm{u})^{\lambda_{jd}}}{\sigma_{l_{jd}}^{\lambda_{jd}+1}}d\mathrm{u}}{\int_{\partial_j}\mathrm{g}_1(\mu_j|\xi_j)d\mathrm{u}}\right\}\quad\text{(E.6)}$$

E.4 Derivation of $\frac{\partial^2\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial{\sigma^2}_{l_{jd}}}$

$$\frac{\partial^2\mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial{\sigma^2}_{l_{jd}}}=\frac{\partial^2}{\partial{\sigma^2}_{l_{jd}}}\sum_{i=1}^{N}Z_{ij}\left\{\log f(\vec{X}_i|\xi_j)-\log\int_{\partial_j}f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right\}\quad\text{(E.7)}$$

$$\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial {\sigma^2}_{l_{jd}}} = -\sum_{i=1,X_{id}<\mu_{jd}}^{N} \hat{Z}_{ij} \frac{A(\lambda_{jd})\lambda_{jd}(\lambda_{jd}+1)}{\sigma_{l_{jd}}^2} \left(\frac{\mu_{jd}-\mathrm{X}_{id}}{\sigma_{l_{jd}}}\right)^{\lambda_{jd}} \tag{E.8}$$

$$-\sum_{i=1,X_{id}<\mu_{jd}}^{N} \hat{Z}_{ij} A(\lambda_{jd}) \frac{\lambda_{jd}}{\sigma_{l_{jd}}^{\lambda_{jd}+1}} \left(1+\frac{1}{(\sigma_{l_{jd}}+\sigma_{r_{jd}})}\right) \frac{\int_{\partial_j}(\mu_{jd}-X_{jd})^{\lambda_{jd}} \mathrm{g}_1(\mu_j|\xi_j)d\mathrm{u}}{\int_{\partial_j} \mathrm{g}_1(\mu_j|\xi_j)d\mathrm{u}}$$

$$-\sum_{i=1,X_{id}<\mu_{jd}}^{N} \hat{Z}_{ij} \left(A(\lambda_{jd}) \frac{\lambda_{jd}}{\sigma_{l_{jd}}^{\lambda_{jd}+1}}\right)^2 \frac{\int_{\partial_j}(\mu_{jd}-X_{jd})^{2\lambda_{jd}} \mathrm{g}_1(\mu_j|\xi_j)d\mathrm{u}}{\int_{\partial_j} \mathrm{g}_1(\mu_j|\xi_j)d\mathrm{u}}$$

$$+\sum_{i=1,X_{id}<\mu_{jd}}^{N} \hat{Z}_{ij} A(\lambda_{jd}) \frac{\lambda_{jd}(\lambda_{jd}+1)}{\sigma_{l_{jd}}^{\lambda_{jd}+2}} \frac{\int_{\partial_j}(\mu_{jd}-X_{jd})^{\lambda_{jd}} \mathrm{g}_1(\mu_j|\xi_j)d\mathrm{u}}{\int_{\partial_j} \mathrm{g}_1(\mu_j|\xi_j)d\mathrm{u}}$$

$$+\sum_{i=1,X_{id}<\mu_{jd}}^{N} \hat{Z}_{ij} \left(A(\lambda_{jd}) \frac{\lambda_{jd}(\lambda_{jd}+1)}{\sigma_{l_{jd}}^{\lambda_{jd}+2}}\right)^2 \frac{\left(\int_{\partial_j}(\mu_{jd}-X_{jd})^{\lambda_{jd}} \mathrm{g}_1(\mu_j|\xi_j)d\mathrm{u}\right)^2}{\left(\int_{\partial_j} \mathrm{g}_1(\mu_j|\xi_j)d\mathrm{u}\right)^2}$$

E.5 Derivation of $\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma_{r_{jd}}}$

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma_{r_{jd}}} = \frac{\partial}{\partial \sigma_{r_{jd}}} \sum_{i=1}^{N} Z_{ij} \left\{ \log p_j + \log f(\vec{X}_i|\xi_j) + \log \mathrm{H}(\vec{X}_i|j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u} \right\} \tag{E.9}$$

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \sigma_{r_{jd}}} = \sum_{i=1,X_{id}\geq\mu_{jd}}^{N} \hat{Z}_{ij} \frac{A(\lambda_{jd})\lambda_{jd}}{\sigma_{r_{jd}}} \left(\frac{X_{id}-\mu_{jd}}{\sigma_{r_{jd}}}\right)^{\lambda_{jd}} \tag{E.10}$$

$$\sum_{i=1,X_{id}\geq\mu_{jd}}^{N} \hat{Z}_{ij} \left\{ \frac{\int_{\partial_j} \mathrm{g}_2(\mu_j|\xi_j) \frac{A(\lambda_{jd})\lambda_{jd}(\mathrm{u}-\mu_{jd})^{\lambda_{jd}}}{\sigma_{r_{jd}}^{\lambda_{jd}+1}} d\mathrm{u}}{\int_{\partial_j} \mathrm{g}_2(\mu_j|\xi_j)d\mathrm{u}} \right\} \tag{E.11}$$

E.6 Derivation of $\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial {\sigma^2}_{r_{jd}}}$

$$\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial {\sigma^2}_{r_{jd}}} = \frac{\partial^2}{\partial {\sigma^2}_{r_{jd}}} \sum_{i=1}^{N} Z_{ij} \left\{ \log f(\vec{X}_i|\xi_j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u} \right\} \tag{E.12}$$

$$\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial {\sigma^2}_{r_{jd}}} = -\sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} \hat{Z}_{ij}\frac{A(\lambda_{jd})\lambda_{jd}(\lambda_{jd}+1)}{\sigma_{r_{jd}}^2}\left(\frac{\mu_{jd}-\mathrm{X}_{id}}{\sigma_{r_{jd}}}\right)^{\lambda_{jd}} \tag{E.13}$$
$$-\sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} \hat{Z}_{ij}A(\lambda_{jd})\frac{\lambda_{jd}}{\sigma_{r_{jd}}^{\lambda_{jd}+1}}\left(1+\frac{1}{(\sigma_{l_{jd}}+\sigma_{r_{jd}})}\right)\frac{\int_{\partial_j}(\mu_{jd}-X_{jd})^{\lambda_{jd}}\mathrm{g}_2(\mu_j|\xi_j)d\mathrm{u}}{\int_{\partial_j}\mathrm{g}_2(\mu_j|\xi_j)d\mathrm{u}}$$
$$-\sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} \hat{Z}_{ij}\left(A(\lambda_{jd})\frac{\lambda_{jd}}{\sigma_{r_{jd}}^{\lambda_{jd}+1}}\right)^2\frac{\int_{\partial_j}(\mu_{jd}-X_{jd})^{2\lambda_{jd}}\mathrm{g}_2(\mu_j|\xi_j)d\mathrm{u}}{\int_{\partial_j}\mathrm{g}_2(\mu_j|\xi_j)d\mathrm{u}}$$
$$+\sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} \hat{Z}_{ij}A(\lambda_{jd})\frac{\lambda_{jd}(\lambda_{jd}+1)}{\sigma_{r_{jd}}^{\lambda_{jd}+2}}\frac{\int_{\partial_j}(\mu_{jd}-X_{jd})^{\lambda_{jd}}\mathrm{g}_2(\mu_j|\xi_j)d\mathrm{u}}{\int_{\partial_j}\mathrm{g}_2(\mu_j|\xi_j)d\mathrm{u}}$$
$$+\sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} \hat{Z}_{ij}\left(A(\lambda_{jd})\frac{\lambda_{jd}(\lambda_{jd}+1)}{\sigma_{r_{jd}}^{\lambda_{jd}+2}}\right)^2\frac{\left(\int_{\partial_j}(\mu_{jd}-X_{jd})^{\lambda_{jd}}\mathrm{g}_2(\mu_j|\xi_j)d\mathrm{u}\right)^2}{\left(\int_{\partial_j}\mathrm{g}_2(\mu_j|\xi_j)d\mathrm{u}\right)^2}$$

E.7 Derivation of $\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \lambda_{jd}}$

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \lambda_{jd}} = \frac{\partial}{\partial \lambda_{jd}}\sum_{i=1}^{N} Z_{ij}\left\{\log p_j + \log f(\vec{X}_i|\xi_j) + \log \mathrm{H}(\vec{X}_i|j) - \log\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right\} \tag{E.14}$$
$$= \sum_{i=1}^{N} Z_{ij}\left\{\frac{\partial}{\partial \lambda_{jd}}\log f(\vec{X}_i|\xi_j) - \frac{\partial}{\partial \lambda_{jd}}\log\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}\right\}$$

$$h(X_{id}|\xi_j) = \frac{\partial}{\partial \lambda_{jd}}\log f(X_{id}|\xi_j) \tag{E.15}$$
$$= \left[\frac{1}{\lambda_{jd}} - \frac{2}{3}\left(\frac{\psi(3/\lambda_{jd})-\psi(1/\lambda_{jd})}{\lambda_{jd}^2}\right)\right]$$
$$+\sum_{i=1,X_{jd}<\mu_{jd}}^{N} Z_{ij}A(\lambda_{jd})\left(\frac{\mu_{jd}-X_{ik}}{\sigma_{l_{jd}}}\right)^{\lambda_{jd}}\left[\left(\frac{3\psi(3/\lambda_{jd})-\psi(1/\lambda_{jd})}{2\lambda_{jd}}\right)-\log\left(\frac{\mu_{jd}-X_{ik}}{\sigma_{l_{jd}}}\right)\right] \tag{E.16}$$
$$+\sum_{i=1,X_{jd}\geq\mu_{jd}}^{N} Z_{ij}A(\lambda_{jd})\left(\frac{X_{ik}-\mu_{jd}}{\sigma_{r_{jd}}}\right)^{\lambda_{jd}}\left[\left(\frac{3\psi(3/\lambda_{jd})-\psi(1/\lambda_{jd})}{2\lambda_{jd}}\right)-\log\left(\frac{X_{ik}-\mu_{jd}}{\sigma_{l_{jd}}}\right)\right] \tag{E.17}$$

$$\frac{\partial}{\partial \lambda_{jd}}\log\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u} = \frac{\frac{\partial}{\partial \lambda_{jd}}\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}} = \frac{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)h(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j)d\mathrm{u}} \tag{E.18}$$

$$\frac{\partial \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \lambda_{jd}} = \sum_{i=1}^{N} Z_{ij} \left\{ h(X_{id}|\xi_j) - \frac{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) h(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} \right\} \tag{E.19}$$

E.8 Derivation of $\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \lambda^2{}_{jd}}$

$$\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \lambda^2{}_{jd}} = \frac{\partial^2}{\partial \lambda^2{}_{jd}} \sum_{i=1}^{N} Z_{ij} \left\{ \log f(\vec{X}_i|\xi_j) - \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right\} \tag{E.20}$$

$$\begin{aligned}
h'(\vec{X}_i|\xi_j) &= \frac{\partial}{\partial \lambda_{jd}} \left(\frac{\partial}{\partial \lambda_{jd}} \log f(\vec{X}_i|\xi_j) \right) \\
&= \left[\frac{1}{\lambda_{jd}^2} + \frac{3\psi'(1/\lambda_{jd})}{2\lambda_{jd}^4} + 3\frac{\psi(1/\lambda_{jd}) - \psi(3/\lambda_{jd})}{2\lambda_{jd}^3} - \frac{9\psi'(2/\lambda_{jd})}{2\lambda_{jd}^4} \right] \\
&+ A(\lambda_{jd}) \sum_{i=1, X_{jd}<\mu_{jd}}^{N} Z_{ij} \left(\frac{\mu_{jd} - X_{ik}}{\sigma_{l_{jd}}} \right)^{\lambda_{jd}} \times \\
&\left[\left(\frac{9\psi'(3/\lambda_{jd}) - \psi'(1/\lambda_{jd})}{2\lambda_{jd}^3} + \frac{3\psi(3/\lambda_{jd}) - \psi(1/\lambda_{jd})}{2\lambda_{jd}^2} \right) \right. \\
&\left. + \left(\frac{3\psi(3/\lambda_{jd}) - \psi(1/\lambda_{jd})}{2\lambda_{jd}} - \log \left(\frac{\mu_{jd} - X_{ik}}{\sigma_{l_{jd}}} \right)^2 \right) \right] \\
&+ A(\lambda_{jd}) \sum_{i=1, X_{jd}\geq\mu_{jd}}^{N} Z_{ij} \left(\frac{X_{ik} - \mu_{jd}}{\sigma_{l_{jd}}} \right)^{\lambda_{jd}} \times \\
&\left[\left(\frac{9\psi'(3/\lambda_{jd}) - \psi'(1/\lambda_{jd})}{2\lambda_{jd}^3} + \frac{3\psi(3/\lambda_{jd}) - \psi(1/\lambda_{jd})}{2\lambda_{jd}^2} \right) \right. \\
&\left. + \left(\frac{3\psi(3/\lambda_{jd}) - \psi(1/\lambda_{jd})}{2\lambda_{jd}} - \log \left(\frac{X_{ik} - \mu_{jd}}{\sigma_{l_{jd}}} \right)^2 \right) \right]
\end{aligned} \tag{E.21}$$

$$\frac{\partial}{\partial \lambda_{jd}} \left(\frac{\partial}{\partial \lambda_j} \log \int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right) = \left\{ \frac{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) \{h^2(\vec{\mathrm{u}}|\xi_j) + h'(\vec{\mathrm{u}}|\xi_j)\} d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} - \frac{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) h(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right)^2}{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right)^2} \right\} \tag{E.22}$$

$$\frac{\partial^2 \mathscr{L}(\mathscr{X},\mathscr{Z}|\Theta)}{\partial \lambda^2{}_{jd}} = \sum_{i=1}^{N} Z_{ij} \left[h'(\vec{X}_i|\xi_j) - \left\{ \frac{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) \{h^2(\vec{\mathrm{u}}|\xi_j) + h'(\vec{\mathrm{u}}|\xi_j)\} d\mathrm{u}}{\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u}} - \frac{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) h(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right)^2}{\left(\int_{\partial_j} f(\vec{\mathrm{u}}|\xi_j) d\mathrm{u} \right)^2} \right\} \right] \tag{E.23}$$