Unsupervised and Semi-Supervised Learning

*Series Editor:* M. Emre Celebi

Nizar Bouguila
Wentao Fan *Editors*

# Mixture Models and Applications

Springer

# Unsupervised and Semi-Supervised Learning

**Series Editor**

M. Emre Celebi, Computer Science Department, Conway, Arkansas, USA

Springer's Unsupervised and Semi-Supervised Learning book series covers the latest theoretical and practical developments in unsupervised and semi-supervised learning. Titles – including monographs, contributed works, professional books, and textbooks – tackle various issues surrounding the proliferation of massive amounts of unlabeled data in many application domains and how unsupervised learning algorithms can automatically discover interesting and useful patterns in such data. The books discuss how these algorithms have found numerous applications including pattern recognition, market basket analysis, web mining, social network analysis, information retrieval, recommender systems, market research, intrusion detection, and fraud detection. Books also discuss semi-supervised algorithms, which can make use of both labeled and unlabeled data and can be useful in application domains where unlabeled data is abundant, yet it is possible to obtain a small amount of labeled data.

Topics of interest in include:

- Unsupervised/Semi-Supervised Discretization
- Unsupervised/Semi-Supervised Feature Extraction
- Unsupervised/Semi-Supervised Feature Selection
- Association Rule Learning
- Semi-Supervised Classification
- Semi-Supervised Regression
- Unsupervised/Semi-Supervised Clustering
- Unsupervised/Semi-Supervised Anomaly/Novelty/Outlier Detection
- Evaluation of Unsupervised/Semi-Supervised Learning Algorithms
- Applications of Unsupervised/Semi-Supervised Learning

While the series focuses on unsupervised and semi-supervised learning, outstanding contributions in the field of supervised learning will also be considered. The intended audience includes students, researchers, and practitioners.

More information about this series at http://www.springer.com/series/15892

Nizar Bouguila • Wentao Fan
Editors

# Mixture Models and Applications

Springer

*Editors*
Nizar Bouguila
Concordia Institute for Information Systems Engineering
Concordia University
QC, Montreal, Canada

Wentao Fan
Department of Computer Science and Technology
Huaqiao University
Xiamen, China

ISSN 2522-848X ISSN 2522-8498 (electronic)
Unsupervised and Semi-Supervised Learning
ISBN 978-3-030-23875-9 ISBN 978-3-030-23876-6 (eBook)
https://doi.org/10.1007/978-3-030-23876-6



This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

Increasingly, business, government agencies, and scientists are confronted with large amounts of heterogenous data that are critical for the daily activities, but not well enough analyzed to get the valuable information and knowledge that they potentially hide. The availability of large data sets has changed the scientific approaches to data mining. This has given rise to the need to develop efficient data modeling tools. Among these approaches, mixture models have become a tool of choice in the last years in many scientific domains [1–3]. This is mainly due to their ability to offer a well-principled approach to clustering. New challenges (e.g., Big Data), new approaches (e.g., deep learning), and new technologies (e.g., cloud computing, Internet of Things, etc.) have added new problems when deploying mixture models in real-life scenarios. And several new frameworks based on mixture models have been proposed. The importance of mixture models as a powerful learning machine is evident by the great plethora of papers dedicated to this subject. Such models are finding applications in almost any area of human endeavor. This includes applications in engineering, science, medicine, and business, just to name a few. At the same time, however, there are a lot of challenges related to the development and application of mixture models. Indeed, very few books present a comprehensive discussion about the application of such models to many real-life domains. The present edited book shows clearly that mixture models may be applied successfully in a variety of applications if well deployed.

The book contains 14 chapters that are grouped into 5 parts, namely, Gaussian-based models (3 chapters), generalized Gaussian-based models (2 chapters), spherical and count data clustering (3 chapters), bounded and semi-bounded data clustering (3 chapters), and image modeling and segmentation (3 chapters). In the first chapter, Parsons presents a Gaussian mixture model approach to classify response types. The parameter estimates obtained from fitting the proposed Gaussian model are used in a naive Bayesian classifier to perform the classification task. In Chap. 2, Berio et al. use Gaussian mixtures for the interactive generation of calligraphic trajectories. The authors exploit the stochastic nature of the Gaussian mixture combined with an optimal control to generate paths with natural variation. The merits of the approach are tested by generating curves and traces that are

similar from a geometrical and dynamical point of views to the ones that can be observed in art forms such as calligraphy or graffiti. In Chap. 3, Calinon presents an interesting overview of techniques used for the analysis, edition, and synthesis of continuous time series, with emphasis on motion data. The author exploits the fact that mixture models allow the decomposition of time signals as a superposition of basis functions. Several applications with radial, Bernstein, and Fourier basis functions are presented in this chapter. A generalization to the Gaussian mixture called multivariate bounded asymmetric Gaussian mixture model is proposed by Azam et al. in Chap. 4. The proposed model is learned via expectation maximization and applied to several real-life applications such as spam filtering and texture image clustering. Another generalization is proposed in Chap. 5 by Najar et al. and applied for online recognition of human action and facial expression as well as pedestrian detection from infrared images. In Chap. 6, Fan et al. tackle the problem of spherical data clustering by developing an infinite mixture model of von Mises distributions. A localized feature selection approach is integrated within the developed model to detect relevant features. The resulting model is learned via variational inference and applied to two challenging applications, namely, topic novelty detection and image clustering. A hybrid generative discriminative framework, based on an exponential approximation to two distributions dedicated to count data modeling, namely, the multinomial Dirichlet and the multinomial generalized Dirichlet, is developed in Chap. 7 by Zamzami and Bouguila. Several SVM kernels are developed within this hybrid framework and applied to the problem of analyzing activities in surveillance scenes. A challenging problem when considering the multinomial Dirichlet and the multinomial generalized Dirichlet distribution in statistical frameworks is the computation of the log-likelihood function. This problem is tackled in Chap. 8 by Daghyani et al. by approximating this function using Bernoulli polynomials. The approach is validated via two clustering problems: natural scene clustering and facial expression recognition. A unified approach for the estimation and selection of finite bivariate and multivariate beta mixture models is developed in Chap. 9 by Manouchehri and Bouguila. The approach is based on minimum message length and deployed to several problems (e.g., sentiment analysis, credit approval, etc.). In Chap. 10, Maanicshah et al. tackle the problem of positive vector clustering by developing a variational Bayesian algorithm to learn finite inverted Beta-Liouville mixture models. Applications such as image clustering and software defect detection are used to validate the model. In Chap. 11, Kalra et al. examine and analyze multimodal medical images by developing an unsupervised learning algorithm based on online variational inference for finite inverted Dirichlet mixture models. The algorithm is validated using challenging applications from the medical domain. Kalsi et al. tackle in Chap. 12 image segmentation problem by integrating spatial information within three mixture models based on inverted Dirichlet, inverted generalized Dirichlet, and inverted Beta-Liouville distributions. The same problem is approached in Chap. 13 by Chen et al. by developing a spatially constrained inverted Beta-Liouville mixture model applied to both simulated and real brain magnetic resonance imaging data. Finally, Chap. 14 by Channoufi et al. presents a flexible statistical model for unsupervised image modeling and

segmentation. The model is based on bounded generalized Gaussian mixtures learned using maximum likelihood estimation and minimum description length principle.

Montreal, QC, Canada — Nizar Bouguila
Xiamen, China — Wentao Fan

## References

1. McLachlan, G.J., Peel, D.: Finite Mixture Models. Wiley, New York (2000)
2. McNicholas, P.D.: Mixture Model-Based Classification. Chapman and Hall/CRC, Boca Raton (2016)
3. Schlattmann, P.: Medical Applications of Finite Mixture Models. Springer, Berlin (2009)

# Contents

# Contributors

**Rula Al-Azawi** Gulf College, Al Maabelah, Muscat, Oman

**Ali Al-Badi** Gulf College, Al Maabelah, Muscat, Oman

**Basim Alghabashi** Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, QC, Canada

**Roobaea Alroobaea** Taif University, Taif, Saudi Arabia

**Muhammad Azam** Department of Electrical and Computer Engineering (ECE), Concordia University, Montreal, QC, Canada

**Daniel Berio** Goldsmiths, University of London, London, UK

**Nizar Bouguila** Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

**Sami Bourouis** Taif University,Taif, Saudi Arabia

Université de Tunis El Manar, LR-SITI Laboratoire Signal, Image et Technologies de l'Information, Tunis, Tunisie

**Sylvain Calinon** Idiap Research Institute, Martigny, Switzerland

**Ines Channoufi** Université de Tunis El Manar, Ecole Nationale d'Ingénieurs de Tunis, LR-SITI Laboratoire Signal, Image et Technologies de l'Information, Tunis, Tunisie

**Wenmin Chen** Department of Computer Science and Technology, Huaqiao University, Xiamen, China

**Yewang Chen** Department of Computer Science and Technology, Huaqiao University, Xiamen, China

**Ziyi Chen** Department of Computer Science and Technology, Huaqiao University, Xiamen, China

**Masoud Daghyani** Department of Electrical and Computer Engineering (ECE), Concordia University, Montreal, QC, Canada

**Wentao Fan** Department of Computer Science and Technology, Huaqiao University, Xiamen, China

**Alrence S. Halibas** Gulf College, Al Maabelah, Muscat, Oman

**Meeta Kalra** Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

**Jaspreet Singh Kalsi** Department of Electrical and Computer Engineering (ECE), Concordia University, Montreal, QC, Canada

**Frederic Fol Leymarie** Goldsmiths, University of London, London, UK

**Kamal Maanicshah** Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

**Narges Manouchehri** Department of Electrical and Computer Engineering (ECE), Concordia University, Montreal, QC, Canada

**Fatma Najar** Laboratoire RISC Robotique Informatique et Systèmes Complexes, Université de Tunis El Manar, ENIT, Tunis, Tunisie

**Hieu Nguyen** Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

**Michael Osadebey** Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway

**Owen E. Parsons** University of Cambridge, Cambridge, UK

**Marius Pedersen** Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway

**Nuha Zamzami** Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

**Bineng Zhong** Department of Computer Science and Technology, Huaqiao University, Xiamen, China

# Part I
# Gaussian-Based Models

# Chapter 1
# A Gaussian Mixture Model Approach to Classifying Response Types


**Owen E. Parsons**

**Abstract** Visual perception is influenced by prior experiences and learned expectations. One example of this is the ability to rapidly resume visual search after an interruption to the stimuli. The occurrence of this phenomenon within an interrupted search task has been referred to as *rapid resumption*. Previous attempts to quantify individual differences in the extent to which rapid resumption occurs across participants relied on using an operationally defined cutoff criteria to classify response types within the task. This approach is potentially limited in its accuracy and could be improved by turning to data-driven alternatives for classifying response types. In this chapter, I present an alternative approach to classifying participant responses on the interrupted search task by fitting a Gaussian mixture model to response distributions. The parameter estimates obtained from fitting this model can then be used in a naïve Bayesian classifier to allow for probabilistic classification of individual responses. The theoretical basis and practical application of this approach are covered, detailing the use of the Expectation-Maximisation algorithm to estimate the parameters of the Gaussian mixture model as well as applying a naïve classifier to data and interpreting the results.

## 1.1 Background

### *1.1.1 The Influence of Prior Information During Interrupted Visual Search*

Visual perception is widely regarded to involve processes of unconscious inference about the state of the external world which act upon incoming noisy sensory information [1]. Hermann von Helmholtz was an early pioneer of the view that visual perception involves higher order processing of ambiguous retinal images.

O. E. Parsons (✉)
University of Cambridge, Cambridge, UK
e-mail: oep20@cam.ac.uk



N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6_1

He suggested that vision was a process of finding the most likely state of visual stimuli based on both the sensory information being received and the previous experiences of the observer [2]. This view of vision, as a process of testing hypotheses about the state of the world, has since been strongly advocated and perception is now understood to be heavily influenced by our expectations of the external environment [3]. These expectations help to solve any ambiguities in the incoming sensory information and enable us to process visual scenes in a fast and efficient way.

Prior expectations have been shown to influence performance during visual search tasks [4–9]. One particular set of studies, which were carried out by Lleras and colleagues, demonstrated how periodically removing the search display during visual search tasks results in a unique distribution of response times [4, 10, 11]. These results illustrated the effects of previously acquired information on search performance. The initial paradigm within these studies required participants to complete a visual search task in which the search display was only visible for short intervals, while being intermittently interrupted by a blank screen [4]. By separating responses into those which occurred after a single presentation of the search display and those which occurred after two or more presentations, the authors found that the distributions of these two response types were distinct. Responses which immediately followed the first presentation of the search display showed a typical unimodal distribution, with all responses occurring after 500 ms from the onset of the search display. However, responses that followed subsequent presentations of the search display showed a clear bimodal distribution with a large proportion of responses occurring within 500 ms of the most recent presentation of the search display. This was interpreted as evidence for a predictive aspect of visual processing in the latter response type, as participants were able to use information acquired from previous exposures of the search display to facilitate their search performance on subsequent presentations.

Lleras and colleagues built on this initial finding by carrying out a number of different manipulations to the original task design in order to better understand the mechanisms of this phenomenon and to rule out alternative explanations for their results [4]. First, they implemented an adaptation of the original paradigm in which the participants had to search for two separate targets in parallel which occurred within distinct search displays that alternated on each presentation. This version of the task produced similar response distributions from participants as the original task, which provided evidence that the results they found in the original task were not simply the product of delayed responses following previous presentations of the display. The authors also experimented with increasing the display time of their search display from 100 ms to 500 ms, which resulted in a stronger influence of prior information on search performance as the participants had longer to accumulate visual information.

Importantly, they were able to rule out the possibility that the effects they observed in the original study were due to a confirmation bias. This refers to a potential strategy where participants would withhold their response following the initial presentation of the search display until they could confirm their decision after

viewing a subsequent presentation. The authors assessed whether this strategy was adopted by participants by inserting catch trials into the task (20% of time) in which the search display did not reappear following the initial presentation. The absence of further presentations of the search display forced participants to respond when they realised that they weren't going to be presented with any additional information. The results from this version of the task found that responses which occurred during these catch trials were likely to have been generated by random guessing, suggesting that a confirmation strategy was unlikely to have been the cause of the observed results in the original task.

### *1.1.2 Quantifying Individual Differences During the Interrupted Search Task*

It is common that distributions of responses which are obtained from single-condition behavioural tasks (tasks in which the behavioural paradigm is consistent across all trials) are assumed to be a result of a single underlying cognitive process. Distinct cognitive processes are more commonly seen in multiple-condition tasks where two types of condition are presented to participants. A classic example of a multiple-condition task is the Posner cueing task, in which trials may either have valid or invalid cues [12]. In tasks such as this, the data are normally stratified by the type of task condition to allow for statistical comparison. This is straightforward in multiple-condition tasks where the different response types occur as a direct result of task manipulation. However, a different approach is required in the case of single-condition tasks, such as the interrupted search task, as different response types occur throughout the task independently of any task manipulation. This means there are no directly observable labels that indicate which response type occurred in any given trial.

Previous attempts have been made to classify response types during the interrupted search task in order to quantify the effects of rapid resumption across individual participants. Lleras and colleagues carried out a subsequent study which looked at whether there were age-related differences in the extent to which individuals showed the effects of rapid resumption [10]. In their study, they focused on responses that occurred after subsequent presentations of the search display (in which rapid resumption could occur) and discarded responses that occurred immediately after the first presentation of the search display (in which rapid resumption could not occur). They classified trials where rapid resumption was thought to have occurred using a cutoff value of 500 ms, which was based on their operational definition of rapid resumption. This allowed for a comparison to be made between the reaction time distributions of the two different response types and for the relative proportion of each response type to be calculated. Using this method, they were able to calculate the ratio of trials in which rapid resumption did and did not occur and then used this to assess for age-related effects. While they found

increasing age led to an improvement in overall visual search performance, they were unable to find an association between age and the extent to which participants displayed the effects of rapid resumption.

The method developed by Lleras and colleagues has some potential issues regarding its validity and suitability for classifying response types in the interrupted search paradigm. First, the defined cutoff used to differentiate between response types is slightly arbitrary as it wasn't derived empirically from behavioural data. The cutoff used in this approach was chosen primarily based on visual inspection of data [4, 11] and is therefore unlikely to allow for optimal labelling of the different response types. By using more sophisticated statistical methods, empirical data could be used to classify response types more accurately. Second, the use of a cutoff point leads to binary classifications that might lose some of the richness of the behavioural data. To further illustrate the potential variance in performance that this method fails to capture, I generated simulated data for 3 different hypothetical response distributions (see Fig. 1.1). These 3 response distributions were created using distinct underlying generative models. Distributions (a) and (c) were each drawn from single Gaussians. While both of these had a mean reaction time of $\mu = 0.5$, they had differing variances of $\sigma = 0.07$ and $\sigma = 0.3$, respectively. Distribution (b) was drawn from a mixture of two Gaussians with the same variance ($\sigma = 0.1$) but different means ($\mu = 0.25$ and $\mu = 0.75$). Using the approach by Lleras and colleagues [10] to classify these different distributions (in terms of the proportion of rapid resumption responses that they contain) gives us the same value (0.5) for all 3 distributions. As they clearly have distinct underlying generative models, this result highlights how this method fails to capture certain types of variation in response distributions that may be indicative of differences in performance on the task.

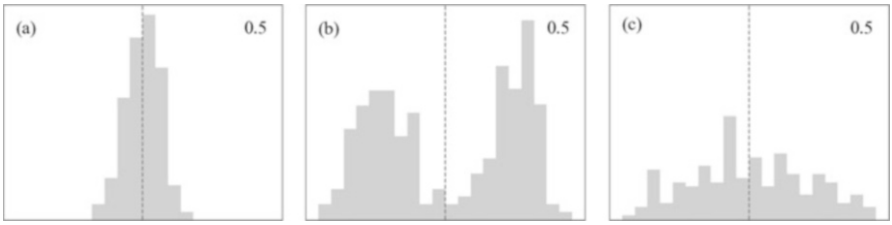



**Fig. 1.1** Simulated reaction time distributions. Distribution (**a**) was drawn from a single Gaussian of $\mu = 0.5$ and $\sigma = 0.07$. Distribution (**b**) was drawn equally ($\lambda = 0.5$) from a mixture of two Gaussians with the same variance ($\sigma = 0.1$) but different means ($\mu = 0.25$ and $\mu = 0.75$). Distribution (**c**) was drawn from a single Gaussian of $\mu = 0.5$ and $\sigma = 0.3$. The ratio of rapid to non-rapid responses, using the method suggested by Lleras et al. [10], is shown in the top right corner of each plot

### 1.1.3 An Alternative Approach to Classifying Response Types During Interrupted Search

One way in which the evaluation of performance in the interrupted search task could be improved is through the use of an empirical data-driven approach to classify response types. The following chapter presents a novel method which uses behavioural data to drive response classification. Considering the data obtained from the interrupted search task, the overall distribution of responses can be viewed as being comprised of two separate distributions. When distributions are derived from two or more distinct processes, the underlying probabilistic structure can be captured using mixture modelling [13]. Based on the evidence put forward by Lleras and colleagues [4, 11], there is a strong reason to believe that there are two distinct response types that occur within the interrupted search paradigm, these being (1) those responses which involve rapid resumption and (2) responses which don't involve rapid resumption.

In terms of the true underlying cognitive mechanisms responsible for the different response types, there is no direct way of observing which response type occurred in any given trial. Therefore, the response type can be described as a *latent variable* (or a *hidden variable*), a variable which is not directly observable but can be inferred from other observed variables. The main observed variable that can be used in the present study is reaction time. The method used by Lleras and colleagues was essentially a way of using a simple classification rule to infer the latent variable, response type, from the observed variable, reaction time. The main concern with this approach, as outlined earlier, is the suitability of the classification rule used to infer the latent variable from the observed data. Here, I present a novel data-driven approach that uses reaction times from trials to infer the most likely response type for any given trial.

### 1.1.4 Aims of This Chapter

This chapter aims to clearly present a method of applying the Expectation-Maximisation algorithm to fit a Gaussian mixture model to behavioural data and demonstrating how this can then be used to classify response types based on which generative process they were likely to have been produced by. Here, I will focus primarily on applying the outcomes from this approach to assessing whether the original cutoff point suggested by Lleras and colleagues is valid. The results produced by this novel method will also be compared with the results from the method used by Lleras et al. to assess whether the classifications produced by the two methods differ significantly. However, as outlined above, this approach also has the potential to provide a number of additional advantages such as individualised modelling of classification criteria as well as potential quantification

of the confidence of classifications. While I will not apply these approaches in the present chapter, the additional benefits of such approaches will be considered during the discussion.

## 1.2 Methods

### *1.2.1 Data Collection*

The dataset presented here was collected as part of a larger study which used a reproduced version of the original task presented by Lleras et al. [4]. A summary of the procedures used for this experiment are presented in the appendix. For the present analysis, only participant responses that occurred following subsequent presentations of the search display were included.

### *1.2.2 Overview of Approach*

This alternative approach to estimating the latent variable from the observed data will be based on extracting the parameters for the separate unimodal distributions of the different response types and then using these parameters to calculate which distribution was more likely to have generated each individual response. The outline of this approach is shown in Fig. 1.2. The overall response distribution for the combined distributions is assumed to be a bimodal distribution, as illustrated by Fig. 1.2a. The first step is to estimate the distribution parameters of the two individual Gaussian distributions that would generate similar data to the observed bimodal distribution. This step is shown in Fig. 1.2b. Once these parameters have been estimated, individual data points can be assessed to determine which of the two Gaussians they were more likely to have been generated by. Two example data points, $x_i$ and $x_j$, are shown in Fig. 1.2c. Both of these example data points are more likely to have been generated by the rightmost Gaussian distribution, as indicated in Fig. 1.2d. One additional advantage of the new approach is that the likelihood to which these data points are expected to have been generated by a given distribution and not the other can also be quantified. In this instance, $x_j$ will be more likely to have been generated by the highlighted Gaussian than $x_i$. The exact details and methodology of approach will be outlined in greater detail below.

### *1.2.3 Gaussian Mixture Models*

One particular example of a latent variable model is the Gaussian mixture model. A mixture model is an example of a hidden model, in which observations are generated

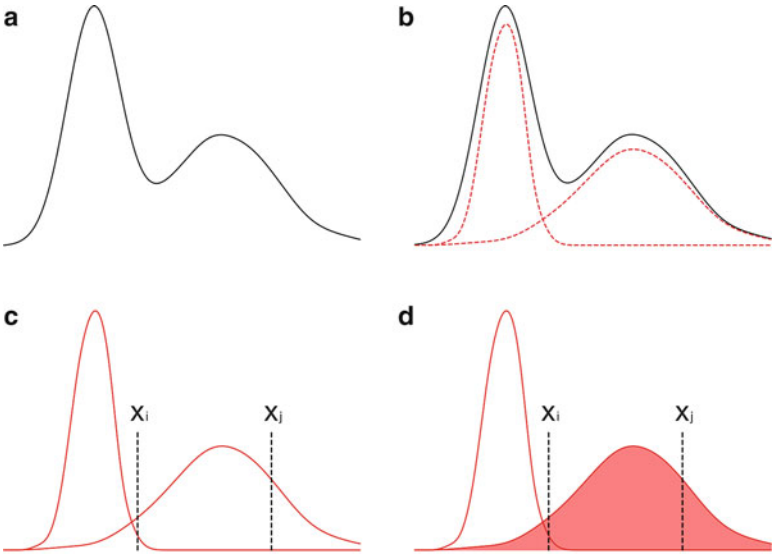



**Fig. 1.2** Demonstration of the procedure used to classify data generated by a bimodal distribution. Diagram (**a**) shows a hypothetical binomial distribution. A Gaussian mixture model can be used to estimate the parameters of the different components of the binomial distribution as shown in diagram (**b**). These can be used to label data points such as $x_i$ and $x_j$ based on which distribution they were most likely to have been drawn from, as shown in diagrams (**c**) and (**d**)

from a mixture of two distinct generative models [14]. A Gaussian mixture model is a common example of this, which consists of a mixture model comprising of two or more Gaussian distributions. The Gaussian distribution can be expressed as:

$$\mathcal{N}(x|\mu,\sigma) = \frac{1}{\sigma(2\pi)^{1/2}} \exp - \left( \frac{(x-\mu)^2}{\sigma^2} \right) \tag{1.1}$$

where $\mu$ is the expected value of the dataset $x$, and $\sigma^2$ is the variance of the dataset. A mixture model can be defined as such:

$$p(x|\{\theta_k\}) = \sum_{k=1}^{K} \lambda_k p_k(x|\theta_k) \tag{1.2}$$

Here, $\lambda_k$ represents the relative weights of the different components (for a model with $k$ components) where $\sum \lambda_k = 1$ and $p_k(x|\theta_k)$ represents the respective components of the subpopulations with $\theta_k$ referring to the parameter set for component $k$. Note that this assumes that $\lambda_k > 0$ for all values of $k$, otherwise the model contains non-contributive subpopulations which can be ignored. Gaussian mixture models are a specific case of mixture models in which the distributions for the subpopulations are Gaussian. This can be written as:

$$p(x|\{\theta_k\}) = \sum \lambda_k \mathcal{N}(x|\mu_k, \Sigma_k) \tag{1.3}$$

Within the mixture model, each individual Gaussian density $\mathcal{N}(x|\mu_k, \Sigma_k)$ is referred to as a component of the mixture and has specific values for its mean $\mu_k$ and covariance $\Sigma_k$. The parameters $\lambda_k$ are the mixing coefficients, which are the relative weights of each distribution within the mixture model. Integrating equation (1.3) with respect to $x$, while incorporating the fact that both $p(x)$ and each of the individual Gaussian components are normalised, gives:

$$\sum_{k=1}^{K} \lambda_k = 1 \tag{1.4}$$

By definition, both $p(x) \geq 0$ and $\mathcal{N}(x|\mu_k, \Sigma_k) \geq 0$. This indicates that $\lambda_k \geq 0$ for all values of $k$. These statements can be combined with Eq. (1.4) to show that the mixing coefficients meet the criteria to be probabilities:

$$0 \leq \lambda_k \leq 1 \tag{1.5}$$

It can also be stated across all the components $k$ that:

$$p(x) = \sum_{k=1}^{K} p(k)p(x|k) \tag{1.6}$$

So, it is clear that $\lambda_k$ is equivalent to $p(k)$, which is the prior probability of a data point coming from the $k_{th}$ component. Additionally, the density $\mathcal{N}(x|\mu_k, \Sigma_k) = p(x|k)$ can be regarded as the probability of data point $x$ given component $k$. The properties of the Gaussian mixture distribution are defined by the parameters $\lambda$, $\mu$ and $\Sigma$, which refer to sets containing the parameters of the individual components $\boldsymbol{\lambda} \equiv \{\lambda_1, \ldots, \lambda_K\}$, $\boldsymbol{\mu} \equiv \{\mu_1, \ldots, \mu_K\}$ and $\boldsymbol{\Sigma} \equiv \{\Sigma_1, \ldots, \Sigma_K\}$.

In the present study, there is no direct information that indicates which of the two underlying processes generate any given response. In order to be able to estimate which underlying process is the most likely cause of individual responses, knowledge of the specific characteristics of the distributions for the different subpopulations is required. In the case of a Gaussian mixture model, estimates need to be obtained for the number of subpopulations, $k$, the characteristics of each Gaussian, $\mu_k$ and $\Sigma_k$, as well as the relative weight of each subpopulation distribution to the overall population, $\lambda_k$. A standard approach for estimating parameters such as these is to find the maximum likelihood. This involves finding values of parameters for which the likelihood function is maximised. The log likelihood function can be written as:

$$\log p(X|\lambda, \mu, \Sigma) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \lambda_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\} \tag{1.7}$$

This equation includes a summation term within the logarithm. This leads to it not being possible to solve the derivative of this in closed-form and so it is necessary to turn to the Expectation-Maximisation algorithm to estimate the parameter values.

### *1.2.4 Expectation-Maximisation Algorithm*

The Expectation-Maximisation algorithm is an iterative method which can be used to find the maximum likelihood estimate in models that contain latent variables [15]. It works by starting with initial parameter estimates and then iterates through an *Expectation Step* and a *Maximisation Step* until the estimates for the parameters converge on a stable solution. The Expectation Step assumes the current parameter estimates are fixed and uses these to compute the expected values of the latent variables in the model. The Maximisation Step takes the expected values of the latent variables and finds updated values for the previous parameter estimates that maximise the likelihood function.

In the case of a Gaussian mixture model, the Expectation Step assumes that the values of all the 3 parameters for the Gaussians in the model are fixed and then computes the probability that each given data point is drawn from each of the individual Gaussians in the model. This property, the probability that a data point is drawn from a specific distribution, is referred to as the *responsibility* of the distribution to a given data point. Once the responsibility values are calculated, the Maximisation Step assumes these responsibilities are fixed and then attempts to maximise the likelihood function across all the model parameters.

The responsibilities are equivalent to the posterior probabilities for a given component within the model and can be calculated as follows:

$$\gamma(z_k) = p(z_k = 1|x) = \frac{p(z_k = 1) \cdot p(x|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1) \cdot p(x|z_j = 1)} \tag{1.8}$$

$$= \frac{\lambda_k \cdot \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \lambda_j \cdot \mathcal{N}(x|\mu_j, \Sigma_j)} \tag{1.9}$$

where $\sum_{j=1}^{K} \lambda_j \cdot \mathcal{N}(x|\mu_j, \Sigma_j)$ is the normaliser term across all components. The responsibility of a component of the model to a data point is equivalent to the normalised probability of a given data point belonging to a specific Gaussians within the mixture model, then weighted by the estimated mixture proportions ($\lambda_k$). This is the posterior probability for a specific distribution given the observed data, $x$. Using this, it is possible to calculate the distribution of the prior mixture weights. The responsibilities can be summed and normalise to estimate the contribution of the individual Gaussians to the observed data:

$$\lambda_k = \frac{1}{N} \sum_i \gamma(z_k) \tag{1.10}$$

The responsibilities of each data point to the different distributions in the model can be used to estimate the mean and standard deviation of the Gaussians:

$$\mu_k = \frac{\sum_i \gamma(z_k) x_i}{\sum_i \gamma(z_k)} \tag{1.11}$$

and

$$\sigma_k = \frac{\sum_i \gamma(z_k)(x_i - \mu_k)(x_i - \mu_k)}{\sum_i \gamma(z_k)} \tag{1.12}$$

It would be straightforward to calculate the posteriors for the components within the model if the distribution parameters were known and, similarly, it would be easy to calculate the parameters were the posterior know. The Expectation-Maximisation algorithm overcomes this issue of circularity by alternating between fixing either the posterior or the parameters while maximising the likelihood. Initially, the parameters are fixed and then the posterior distribution is calculated for the hidden variables. Then, the posterior distribution is fixed, and the parameters are optimised. These steps are repeated in an alternating fashion until the likelihood value converges.

### *1.2.5 Estimation of Mixture Model Parameters*

I used the Expectation-Maximisation algorithm to estimate the parameters for the individual distributions of responses where rapid resumption did occur and responses where rapid resumption did not occur. Once the parameters of these two distributions had been estimated, I would be able to not only reclassify all participant responses using an empirically derived criterion but also quantify the relative likelihood of each individual classification. The Expectation-Maximisation algorithm was carried out by initialising the parameters and then iterating through the Expectation and Maximisation Steps until the parameters converged. The individual steps of the Expectation-Maximisation algorithm are detailed below.

#### 1.2.5.1 Initialisation

The means $\mu_k$, covariances $\Sigma_k$ and mixing coefficients $\lambda_k$ were initialised by using the values obtained from the classification method suggested by Lleras and colleagues [10] to classify data points across all participants and then estimate the distribution parameters for the two response types based on these classifications.

#### 1.2.5.2 Expectation Step

The responsibilities (posteriors) for the individual components were evaluated using the current estimates for the parameter values:

$$\gamma(z_{nk}) = \frac{\lambda_k \cdot \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \lambda_j \cdot \mathcal{N}(x_n|\mu_j, \Sigma_j)} \tag{1.13}$$

#### 1.2.5.3 Maximisation Step

The parameters were then updated by re-estimating them based on the current values for the responsibilities. This can be done using Eqs. (1.10)–(1.12), giving the following update equations:

$$\mu_k^{\text{new}} = \frac{1}{N_k} \cdot \sum_{n=1}^{N} \gamma(z_{nk}) \cdot x_n \tag{1.14}$$

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \cdot (x_n - \mu_k) \cdot (x_n - \mu_k)^T \tag{1.15}$$

$$\lambda_k^{\text{new}} = \frac{N_k}{N} \tag{1.16}$$

where:

$$N_k = \sum_{n=1}^{N} \cdot \gamma(z_{nk}) \tag{1.17}$$

#### 1.2.5.4 Convergence Criteria

Convergence was checked for both the model parameters and log likelihood. The convergence criteria were all set as $10^{-15}$. During each iteration of the Expectation-Maximisation algorithm, the updated parameter and log likelihood estimates were compared to the previous estimates to assess whether the change in values met the convergence criteria. The log likelihood was estimated as follows:

$$\log p(X|\mu, \Sigma, \lambda) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \lambda_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\} \tag{1.18}$$

If any of the parameters or the log likelihood satisfied the convergence criteria, then the algorithm terminated, otherwise the next iteration was started.

### *1.2.6 Log Probability Ratio*

Once the parameters for the distributions of rapid and non-rapid responses had been estimated, log probability ratios were calculated for all trials across each participant individually. The log probability ratios could be used to classify responses as either rapid or non-rapid which in turn allowed for an updated calculation of the proportion of rapid responses for all participants. This updated measure will be referred to as *RR-Model* which can then be compared to the *RR-Basic* scores that were calculated using the cutoff method outlined by Lleras and colleagues [10]. Additionally, the log probability ratios allow for a measure of the cumulative confidence of classifications to be calculated for individual participants. For the current dataset, the set of latent variables (which refer to the components of the Gaussian mixture model) is $\boldsymbol{Z} \equiv \{z_R, z_S\}$ where $z_R$ and $z_S$ are multinominal vectors such that $z_R = 1$ is a classification of a rapid response and $z_S = 1$ is a classification of a slow response. For any given response $x_i$, the probabilities of the observed responses can be formulated as either being classified as a rapid response or a slow response (non-rapid response). These can be written, respectively, as:

$$P(z_R = 1 \mid x) = \frac{P(x \mid z_R = 1)\, P(z_R = 1)}{P(x)} \tag{1.19}$$

and

$$P(z_S = 1 \mid x) = \frac{P(x \mid z_S = 1)\, P(z_S = 1)}{P(x)} \tag{1.20}$$

As a binary classification (two classes) has been used and slow trials are defined as any trials in which rapid resumption has not occurred, it can also be stated that:

$$P(z_R = 1 \mid x) + P(z_S = 1 \mid x) = 1 \tag{1.21}$$

Equations (1.19) and (1.20) can then be combined with Eq. (1.21). This gives us an equation for the normaliser term $P(x)$:

$$P(x \mid z_R = 1)P(z_R = 1) + P(x \mid z_S = 1)P(z_S = 1) = P(x) \tag{1.22}$$

This can be rearranged to give the probability that data point $x$ will be classified as a rapid response:

$$P(z_R = 1 \mid x) = \frac{1}{\frac{P(x|z_S=1)P(z_S=1)}{P(x|z_R=1)P(z_R=1)} + 1} \tag{1.23}$$

All the terms within this equation are computable from the observed data. The prior probabilities $P(z_R = 1)$ and $P(z_S = 1)$ can be estimated from the observed data. The posterior terms $P(x|z_R = 1)$ and $P(x|z_S = 1)$ can then be calculated by assuming that:

$$P(x \mid z_k = 1) = \mathcal{N}(x \mid \mu_k, \sigma_k^2) \tag{1.24}$$

where $z_k = 1$ is the response type (either rapid, $z_R = 1$, or slow, $z_S = 1$) and $\mu_k$ and $\sigma_k^2$ are the estimates for the mean and standard deviation of the given response distribution which were calculated using the Expectation-Maximisation algorithm. From this it is possible to expand Eq. (1.24) using Eq. (1.1) to calculate the log probability ratio, which is the ratio of log likelihood probabilities for the Gaussian components of the model.

$$\begin{aligned}\log \frac{P(z_R = 1|x)}{P(z_S = 1|x)} = -\frac{1}{2}\left(\frac{(x-\mu_r)^2}{\sigma_r^2} - \frac{(x-\mu_s)^2}{\sigma_s^2} + \log \sigma_r^2 - \log \sigma_s^2\right) \\ + \log P(z_R = 1) - \log P(z_S = 1)\end{aligned} \tag{1.25}$$

The final form shows the 3 mains components of the log probability ratio: the variance-weighted Euclidean distances from the means $\left(\frac{(x-\mu_r)^2}{\sigma_r^2} - \frac{(x-\mu_s)^2}{\sigma_s^2}\right)$, the log variances $\left(\log \sigma_r^2 - \log \sigma_s^2\right)$ and the difference in log prior probabilities $(\log P(\omega_r) - \log P(\omega_s))$. A log probability ratio of 0 would suggest that the observed response was equally likely to have been generated by either distribution, with positive values suggesting stronger evidence that the response was a rapid response and negative values suggesting that the observed response was a non-rapid response. These values can be accumulated across all responses for each individual participant using a sequential probability ratio test. This approach rests on the assumption that the outcome on the $n$th trial is independent of the outcome on the $n - 1$th trial. To verify whether this assumption holds, regression analyses can be used to determine whether previous trial response type has an effect on current trial response type. Additionally, it is worth considering that the accumulation of *directional* log probability ratios is not entirely informative as distributions which are evenly balanced across the classification boundary will have values close to zero regardless of the likelihood of the individual trial classifications. Returning to the simulated distributions in Fig. 1.1, accumulation of the direction log probability ratios would still not be able to differentiate between these 3 distributions. Therefore, the absolute values of the log probability ratio for each individual trial could also be considered. These absolute values of the log probability ratios can be accumulated across both response types combined, to give an overall measure of classification confidence for

each participant, or for each of the response types individually, to create 2 distinct within-subjects measures.

## 1.3 Results

### *1.3.1 Parameter Estimation of Response Distributions*

The Expectation-Maximisation algorithm was initialised using the values detailed in the methods section. The algorithm found a two-Gaussian fit for the response distribution. The parameters for the two Gaussians were $\mu_a = 0.324$, $\sigma_a = 0.155$ and $\mu_b = 0.73$, $\sigma_b = 0.119$ with a $\lambda$ of 0.551. To ensure the parameter estimates were accurate, the Expectation-Maximisation algorithm was run 100 times. The algorithm consistently converged on the same values with an average of 631.92 iterations ($SD = 30.85$) taken to converge. The fit of the estimated Gaussians to the observed data is shown in Fig. 1.3.

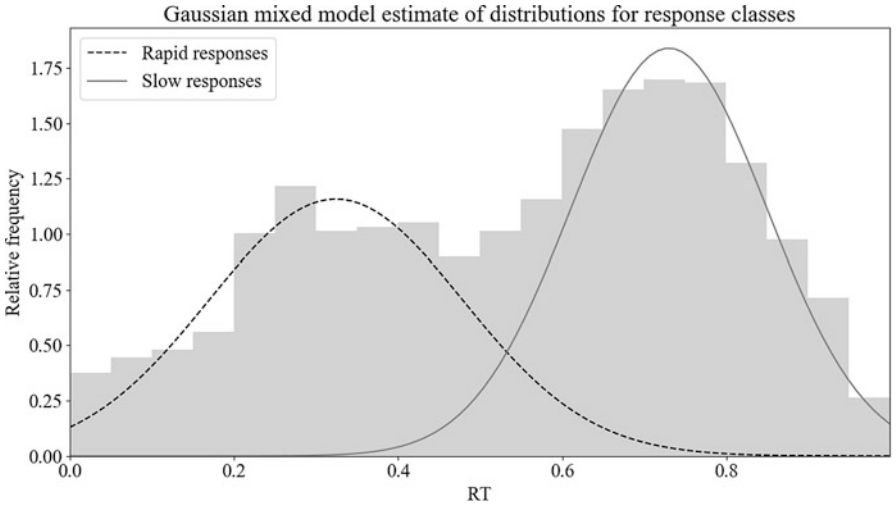



**Fig. 1.3** Histogram showing the percentage of responses within different time bins for standard responses. Data shown for all responses pooled across participants. The two curves show the estimated distributions from the Gaussian mixture model. The Gaussian for rapid responses is shown as a dashed line and the Gaussian for slow responses is shown as the solid line

### 1.3.2 Evaluation of Previous Classification Criteria

To contrast the current method with the method presented by Lleras et al. [10], the threshold value between the two response type classifications was calculated using the parameters obtained here. This value represents the exact within-epoch reaction time for which faster responses would be classified as rapid responses and slower responses would be classified as slow responses. This is calculated by using the model parameters and finding the reaction time for which the absolute log probability ratio is minimised. The calculated threshold reaction time was 432 ms, which indicates that the difference between the value used by Lleras and colleagues and the empirically derived value estimated by the model presented here is a relatively modest 68 ms. While this suggests that the cutoff used by Lleras and colleagues was a fairly good estimate, it is not clear whether the methods produce significantly different values for the ratio of rapid to non-rapid responses.

### 1.3.3 Comparison of Classification Methods

Participant data were then modelled by applying a naïve Bayesian classifier using the estimated parameters. For each trial, a classification likelihood value was calculated using the within-epoch reaction time and the estimated parameters for the underlying Gaussians obtained using the Expectation-Maximisation algorithm. The log probability ratio was calculated for all responses and was used to classify the response type for each individual trial. Trials with positive log probability ratios were classified as rapid responses, and trials with negative log probability ratios were classified as slow responses. No trials were calculated as having a log probability ratio of exactly 0 and so there were no ambiguous cases.

Rapid resumption ratios were calculated using the procedure described above and will be referred to as the RR-Model scores. These scores were then compared to rapid resumption ratios calculated using the method suggested by Lleras et al. [10], which will be referred to as the RR-Basic scores. As expected, a Pearson correlation test revealed a high level of correlation between scores obtained using the two methods ($r = 0.958$, $p < 0.001$). A repeated measures ANOVA was then conducted, with method of calculation (RR-Basic v RR-Model) as the within-subject measure and the calculated rapid resumption ratio values as the outcome variable. There was a significant effect of method ($F_{(1,23)} = 178.8$, $p < 0.001$, $\eta^2 = 0.886$) indicating that scores obtained by the two methods did statistically differ from each other. Participant data are displayed as violin plots [16] in Fig. 1.4. This suggests that, in the present dataset, the Gaussian mixture model approach that was developed produced results that differed to the method used by Lleras et al.

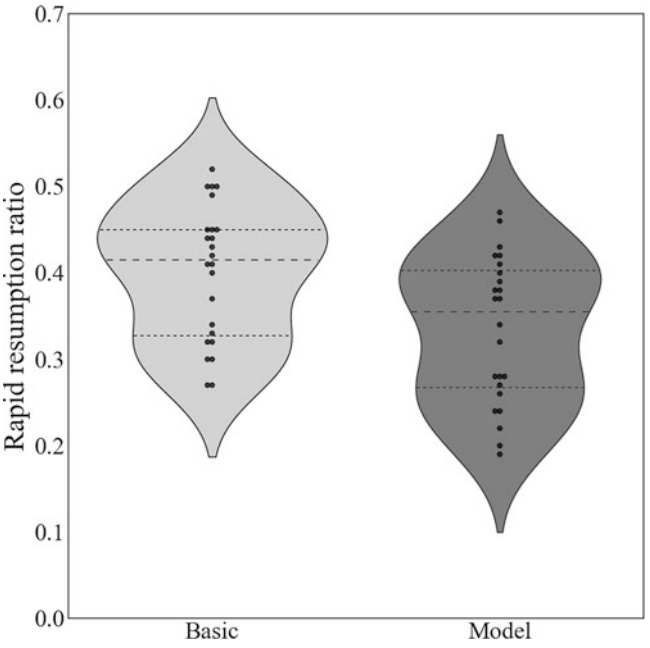



**Fig. 1.4** Violin plots showing the calculated rapid resumption ratio using the RR-Basic (light grey) and RR-Model (dark grey) methods. Median values as well as upper/lower quartiles are shown for each method by the horizontal dashed lines

## 1.4 Discussion

In this chapter, a Gaussian mixture model was successfully fit to the pooled participant response data. This allowed for the estimation of the parameters of the distinct distributions for responses in which rapid resumption occurred and responses in which rapid resumption did not occur. The Expectation-Maximisation algorithm converged on a 2-component model, suggesting that the response distributions for responses in the task that occurred after subsequent presentations of the search display were indeed bimodal. Importantly, the model parameters that were found were then used to calculate a more accurate classification threshold (the exact time at which a response is equally likely to be a rapid response or a slow response) by minimising the absolute log probability ratio between the two Gaussian distributions. This classification threshold was found to be moderately close to the value used by Lleras and colleagues [10]. The value calculated by the model was 432 ms, which was not dramatically different from the approximate value (500 ms) suggested by Lleras et al. However, a significant difference between the ratio scores calculated using the two methods was reported. This suggests that a slight adjustment to the originally suggested cutoff value of 500 ms should be adopted.

One limitation of the approach used by Lleras et al. is that fixed parameters are used to classify the response types across all participants rather than adopting an individualised approach. It may be the case that the best estimates for the parameters of the underlying response distributions vary across individuals and therefore using an approach that estimates the model parameters based on data from the entire sample might lead to less accurate modelling of the response types. The approach presented here could allow for Gaussian mixture models to be fit to single participant data which could overcome this issue. The modelling approach developed in this chapter has the added advantage of allowing for the likelihood of each trial classification to be considered. This method provides a richer set of measures which could be sensitive to variation in task performance that would be overlooked by only considering the relative proportions of response types. While individualised classification or the calculation of classification confidence values were not explored in this chapter, both of these approaches are viable using the method that was developed here. The focus of the present chapter was primarily to assess the cutoff criteria used by Lleras et al., which was achieved. It is hoped that the procedures outlined in this chapter will provide guidance for others who wish to apply similar methods to their datasets.

## Appendix: Additional Methods

### *Participants*

A total of 27 healthy male participants completed the interrupted visual search task. All participants were right handed and had normal or corrected-to-normal vision. The mean age of the group was 30.42($SD = 9.18$), and the mean performance IQ (measured using the WASI [17]) was 114.11($SD = 11.97$). These participants were recruited as the control group in a larger study that was conducted. Participants were recruited from the Cambridge Psychology Volunteers Database or through classified adverts on websites such as Gumtree.

### *Stimuli Presentation*

Stimuli were presented using the Psychtoolbox extension [18, 19] in MATLAB [20]. Stimuli were displayed on a 24″ monitor running at a resolution of 1920 × 1080. Participants were sat with a viewing distance of 60 cm from the screen in a darkened room.

Overall the stimuli presented and procedure used in this study closely match the methods outlined in experiment 1 from Lleras et al. [4]. Participants were required to locate a target T shape within an array of L shapes. Trials either contained 16 visual

items (1 target and 15 distractors) or 32 visual items (1 target and 31 distractors). An even amount of 16 and 32 item trials were presented to each participant in a random order. The effects of distractor density were not considered as part of the analysis presented here.

Items were presented within a centrally positioned white square which subtended a 9° visual angle. The area of the screen outside of the central square was coloured grey. Item positions were generated by randomly placing them inside an invisible 6 × 6 grid. The height and width of each invisible cell within the grid was 1.5°. During display generation, items were initially placed centrally within their grid positions and then a random amount of jitter (±0.2) was applied to this initial position in order to avoid the objects being collinearly aligned.

After generating item positions, one of the items was selected at random to be the target item, and the others were presented as distractor items. All items were generated using two lines of equal length at 90 degrees to each other, with target 'T' shapes placing the second line in the middle of the first line and distractor 'L' shapes placing the second line at the end of the first line. Each of the line segments within the items subtended 0.5° of visual angle. The orientation of each item was randomly selected from four possible options (at 90 degree rotations). Items could be either blue or red in colour and were balanced to ensure an equal number of items of each colour in the display.

## *Procedure*

During each trial, a new search display was generated using the methods detailed above. Trials were preceded by a fixation cross in the centre of the screen for 500 ms. The search display was shown for 100 ms at a time with a 900 ms blank display period in between. Blank display periods showed a white square without any of the search items present. Each cycle of a 100 ms search display presentation and 900 ms blank display will be referred to as an epoch [21]. Trials terminated after a total of 8000 ms without a response or as soon as the participant responded. This meant that on each trial the search display would be visible for a maximum of 8 times (8 epochs). Participants were shown feedback on each trial which stayed on the screen for 1000 ms. This procedure is demonstrated in Fig. 1.5.

Participants were given instructions on the screen which were repeated verbally by the experimenter. Once the participants were happy with the instructions, they were given 15 practice trials to do. After completing the practice trials, all participants completed a control task designed to assess their base reaction time. The control task consisted of 30 trials in which a target object appeared without the addition of any distractor objects. Participants were asked to report the colour of the target shape (red or blue) as quickly as possible by pressing the 'z' key for a blue target or the 'm' key for a red target. Coloured stickers were placed on the keys to indicate which key corresponded to which colour.

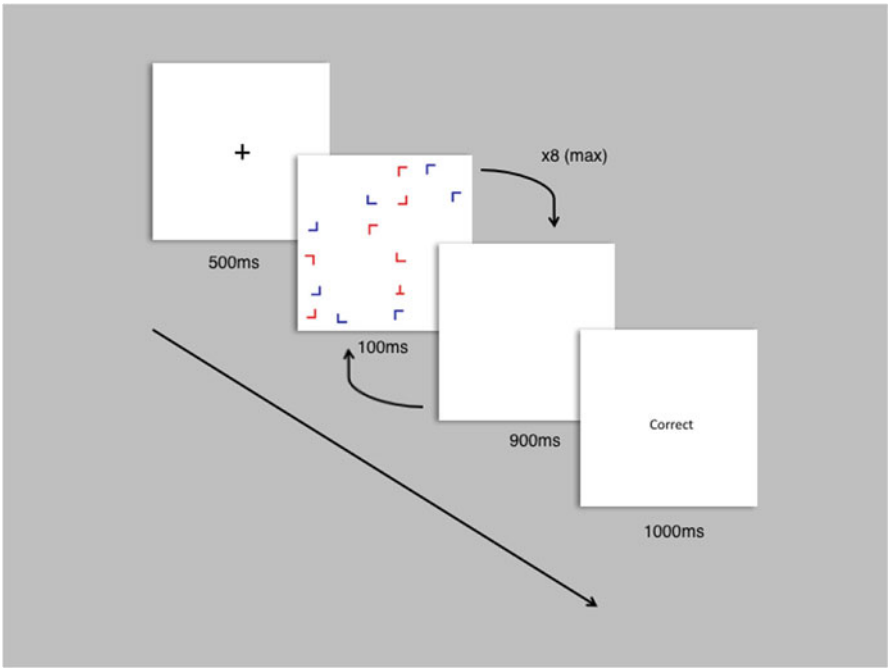



**Fig. 1.5** Diagram showing the stimuli sequence for any given trial. At the start of each trial, participants are presented with a fixation cross for 500 ms. After the fixation cross, the search display is presented for 100 ms followed by a 900 ms interval with a blank screen. The search display is shown a maximum of 8 times in total. Feedback is given for 1000 ms ('Correct' or 'Incorrect') once the participant responds or the trial times out (8000 ms from initial presentation of the search display)

After completing the control task, participants were given a short break before starting the main task. In the main task, participants were again required to report the colour of a target T shape. However, these T shapes were now presented alongside distractor L shapes. Participants completed a total of 10 blocks of 30 trials. Each block was followed by a 30 s rest period. The duration of the full session including the instructions, practice trials, control task and main task was approximately 30 min. Two participants were removed from further analysis for having median reaction times in the control task that were greater than 2 standard deviations from the group mean. An additional participant was removed for having an error rate in the main task that was greater than 2 standard deviations from the group mean. This left a final sample of 24 subjects for the main analyses. Data for all response trials were pooled together for all the participants. Only correct responses were included in this dataset.

## References


1. Bar, M.: Visual objects in context. Nat. Rev. Neurol. **5**(8), 617 (2004)
2. von Helmholtz, H.: Concerning the perceptions in general. In: Treatise on Physiological Optics (1866)
3. Seriès, P., Seitz, A.: Learning what to expect (in visual perception). Front. Hum. Neurosci. **7**, 668 (2013)
4. Lleras, A., Rensink, R.A., Enns, J.T.: Rapid resumption of interrupted visual search: New insights on the interaction between vision and memory. Psychol. Sci. **16**(9), 684–688 (2005)
5. Chun, M.M.: Contextual cueing of visual attention. Trends Cogn. Sci. **4**(5), 170–178 (2000)
6. Kunar, M.A., Flusberg, S.J., Horowitz, T.S., Wolfe, J.M.: Does contextual cueing guide the deployment of attention? J. Exp. Psychol. Hum. Percept. Perform. **33**(4), 816–828 (2007)
7. Makovski, T.: What is the context of contextual cueing? Psychon. Bull. Rev. **23**(6), 1982–1988 (2016)
8. Spaak, E., Fonken, Y., Jensen, O., de Lange, F.P.: The neural mechanisms of prediction in visual search. Cereb. Cortex (New York, NY: 1991) **26**(11), 4327–4336 (2016)
9. Vaskevich, A., Luria, R.: Adding statistical regularity results in a global slowdown in visual search. Cognition **174**, 19–27 (2018)
10. Lleras, A., Porporino, M., Burack, J.A., Enns, J.T.: Enns. Rapid resumption of interrupted search is independent of age-related improvements in visual search. J. Exp. Child Psychol. **109**(1), 58–72 (2011)
11. Lleras, A., Rensink, R.A., Enns, J.T.: Consequences of display changes during interrupted visual search: Rapid resumption is target specific. Percept. Psychophys. **69**(6), 980–993 (2007)
12. Posner, M.I.: Orienting of attention. Q. J. Exp. Psychol. **32**(1), 3–25 (1980)
13. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, Berlin (2006)
14. McLachlan, G.J., Lee, S.X., Rathnayake, S.I.: Finite mixture models. Ann. Rev. Stat. Appl. **6**(1), 355–378 (2019)
15. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B Methodol. **39**, 1–38 (1977)
16. Hintze, J.L., Nelson, R.D.: Violin plots: a box plot-density trace synergism. Am. Stat. **52**(2), 181–184 (1998)
17. Wechsler, D.: Wechsler Adult Intelligence Scale (WAIS–IV), vol. 22, 4th edn, p. 498. NCS Pearson, San Antonio (2008)
18. Brainard, D.H.: The psychophysics toolbox. Spat. Vis. **10**(4), 433–436 (1997)
19. Kleiner, M., Brainard, D., Pelli, D., Ingling, A., Murray, R., Broussard, C.: What's new in psychtoolbox-3. Perception **36**(14), 1 (2007)
20. MATLAB User's Guide MathWorks. MathWorks, South Natick (1989)
21. Rensink, R.A.: Visual search for change: A probe into the nature of attentional processing. Vis. Cogn. **7**(1–3), 345–376 (2000)

# Chapter 2
# Interactive Generation of Calligraphic Trajectories from Gaussian Mixtures


**Daniel Berio, Frederic Fol Leymarie, and Sylvain Calinon**

**Abstract** The chapter presents an approach for the interactive definition of curves and motion paths based on Gaussian mixture model (GMM) and optimal control. The input of our method is a mixture of multivariate Gaussians defined by the user, whose centers define a sparse sequence of key-points, and whose covariances define the precision required to pass through these key-points. The output is a dynamical system generating curves that are natural looking and reflect the kinematics of a movement, similar to that produced by human drawing or writing. In particular, the stochastic nature of the GMM combined with optimal control is exploited to generate paths with natural variations, which are defined by the user within a simple interactive interface. Several properties of the Gaussian mixture are exploited in this application. First, there is a direct link between multivariate Gaussian distributions and optimal control formulations based on quadratic objective functions (linear quadratic tracking), which is exploited to extend the GMM representation to a controller. We then exploit the option of tying the covariances in the GMM to modulate the style of the calligraphic trajectories. The approach is tested to generate curves and traces that are geometrically and dynamically similar to the ones that can be seen in art forms such as calligraphy or graffiti.

## 2.1 Introduction

The hand-drawn curves that can be observed in art forms such as calligraphy [32] and graffiti [8] are often the result of skillful and expressive movements that require years to master. Even after practice, the same trace executed twice will always be different due to motor variability. Mimicking this type of curves and variability with

D. Berio (✉) · F. F. Leymarie
Goldsmiths, University of London, London, UK
e-mail: d.berio@gold.ac.uk; ffl@gold.ac.uk

S. Calinon
Idiap Research Institute, Martigny, Switzerland
e-mail: sylvain.calinon@idiap.ch



N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6_2

conventional geometric computer aided design (GCAD) methods can be difficult. These methods typically describe a curve through the concatenation of piecewise polynomials, which interpolate or approximate the vertices of a control polygon defined by a user. This approach is well suited for geometric design applications. However, the manual positioning of control points can become unintuitive and overly complex when the task at hand requires mimicking the curvilinear patterns that would be produced by the movements of an experienced artist's hand. To this end, we propose a *movement centric* approach to curve design, in which a curve is defined through the synthesis of a movement underlying its production rather than only considering its static geometric trace.

In this chapter, we demonstrate how tools from statistics and optimal control, together with insights from computational motor control, can be combined into a curve generation method that produces synthetic traces that are visually and kinematically similar to the ones made by a human when drawing or writing. The *input* of our method is a *Gaussian mixture model* (GMM)[1] describing a spatial distribution. The output of our method is a distribution of smooth trajectories, with variations and kinematics that are similar to the ones that typically characterize human hand motions. We generate smooth trajectories by forcing a dynamical system to track the centers of each GMM component with a precision determined by the respective covariances. The trajectory evolution is determined by an optimization with a quadratic objective, which is formulated as a trade-off between tracking accuracy and control effort. The latter is expressed as the square magnitude of a $n$th order derivative of position, such as jerk (3rd) or snap (4th), which results in smooth trajectories that are consistent with known principles from computational motor control [18, 41]. Accompanying source codes for the chapter are available at http://doc.gold.ac.uk/autograff/.

## 2.2 Background

The proposed approach is informed by a number of observations and ideas from the domain of computational motor control. Target-directed hand movements are characterized by an archetypal "bell" shaped velocity profile [17, 34, 36]. A number of mathematical models of handwriting movements describe trajectories as the time superposition of multiple target-directed sub-movements [16, 37], where each sub-movement is in turn characterized by a bell-shaped velocity profile. The speed and curvature of human hand movements tend to show an inverse relation [15, 21] with this relation taking the form of a power law for certain types of movements [27, 44]. The duration of hand movements tends to be approximately invariant to scale, a principle that is also known as *isochrony*. Also the duration

[1]We refer the reader to the chapter by O.E. Parsons in this same book [Chapter 1] for an introduction and in-depth description of GMMs and relevant estimation methods.

of sub-movements tends to be approximately equal, a principle that is commonly referred to as *local isochrony* [25], and which is consistent with the hypothesis of central pattern generators [14]. Human hand movements are smooth and appear to obey optimality principles, which can be well modeled as the minimization of an objective function [19]. Popular models express this objective as the minimization of the square-magnitude of higher order positional derivatives [12, 18, 33, 42] or torque [43]. Human movements show inherent variability [6], which tends to increase in parts of the movement that are not critical to the required precision of a task. Todorov and Jordan [41] propose the framework of optimal feedback control and suggest that deviations from an average (smooth) trajectory are corrected only when they interfere with the required task precision. Our method allows to model this principle by explicitly defining the required precision of trajectory segments with full covariance matrices.

Egerstedt and Martin [13] discuss the equivalence between several forms of splines and control theoretic formulations of dynamical systems. The authors show that smoothing splines correspond to the output of a controller found by minimizing a quadratic cost function similar to the one used in our method. In a related line of work, Fujioka et al. [22] optimize the placement of B-spline [10] control points in order to mimic smoothing effects observable in Japanese calligraphy. With our method we extend these principles to a more generic case, in which movement precision as well as coordination patterns are encoded as full covariance matrices, and where the output of the method is a distribution rather than a single trajectory.

In conventional computer graphics applications, hand-drawn curves are usually specified interactively through a sketch-based interface. A user traces a curve with a mouse, trackpad, or tablet. The trace is then processed in order to avoid digitization artefacts and hesitations, with a procedure commonly referred to as curve "neatening" or "fairing" [2, 30, 31, 40]. However, the output of these methods is usually a piecewise polynomial curve set with several control points, and this makes it difficult to later edit or vary the overall trace.

Non photorealistic animation and rendering (NPAR) is the subfield of computer graphics aimed at the simulation of artistic techniques/styles and at clarity of representation [26]. A few methods from this domain also target the generation of curves through the simulation of a hand movement. Haeberli [23] uses a mass-spring system to generate calligraphic curves from input mouse movements. House and Singh [24] use a proportional-integral–derivative (PID) controller to generate sketch-based stylizations of 3D models. AlMeraj et al. [1] mimic the quality of hand-drawn pencil lines with a model of human movements that minimizes changes in acceleration [19]. The method we describe in this chapter allows to achieve similar artistic rendering effects or to generate curves that are similar to the ones produced by a sketch-based interface. We provide a user-friendly interface to input a sparse sequence of key-points, bearing similarities to the interfaces used in conventional GCAD methods.

## 2.3 Trajectory Generation

The input to our method is a GMM with $M$ multivariate components $\mathcal{N}\left(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right)$ defined in a Cartesian space of dimension $D$. The output is a distribution of smooth motions $\mathcal{N}\left(\boldsymbol{x}, \boldsymbol{\Sigma}^{\boldsymbol{x}}\right)$, where each motion tracks the centers $\boldsymbol{\mu}_i$ of the input with a precision defined by the corresponding covariances $\boldsymbol{\Sigma}_i$. Considering a sequence of centers of the mixture components gives a series of *key-points* $(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M)$, results in a descriptor that is similar to the control polygon used in conventional curve generation methods such as Bézier curves or splines. At the same time, the covariance structure of the GMM provides explicit control over the variability and smoothness of the trajectory in the neighborhood of each key-point, together with local or global control of the curvilinear evolution of the trajectory (Fig. 2.1).

Trajectories are generated by optimizing the evolution of a dynamical system that tracks each GMM component sequentially for a given amount of time. A decrease in the variance of a component corresponds to an increased precision requirement and thus forces the trajectory to pass near the component center (Fig. 2.2a). A sufficiently low variance then produces an interpolatory behavior. An increase in the variance corresponds with a lower precision requirement and thus produces a smoothing effect that is similar to the one achieved with approximating splines (Fig. 2.2b). However, the use of full covariances allows more complex spatial constraints to be captured such as forcing a movement to follow a given direction or to pass through a narrow region of space (Fig. 2.2c). The resulting trajectories

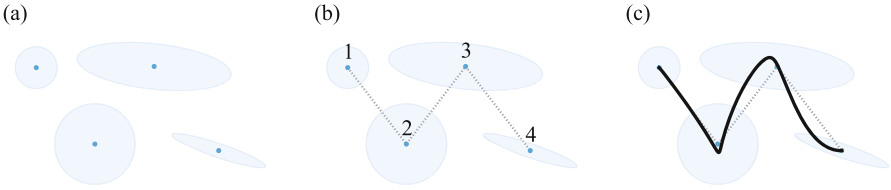



**Fig. 2.1** The trajectory generation method in a nutshell. An input GMM (**a**) is considered as a sequence (**b**). The ordered components are then used to guide the evolution of a dynamical system (**c**)

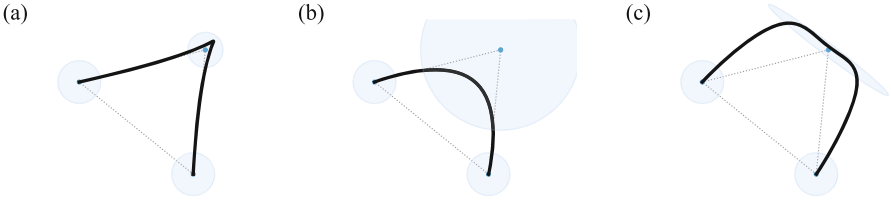



**Fig. 2.2** Variations of a trajectory by manipulating one covariance matrix. (**a**) Using an isotropic covariance with low variance (high precision). (**b**) An increase in variance produces a smoothing effect. (**c**) A full covariance can be used to force the trajectory to remain in a restricted (here nearly flat) region of space

are smooth and have kinematics that are similar to the ones that would be seen in a movement made by a drawing hand, with desirable features such as bell-shaped speed profiles and an inverse relation between speed and curvature.

### 2.3.1 Dynamical System

The model for our trajectory generation mechanism is a discrete linear time-invariant system of order $n$ defined with the state equation:

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t \,, \tag{2.1}$$

where at each time step $t$ the state,

$$\boldsymbol{x}_t = \left[\mathbf{x}^\top, \dot{\mathbf{x}}^\top, \ddot{\mathbf{x}}^\top \ldots, {}^{(n-1)}\mathbf{x}^\top\right]^\top , \tag{2.2}$$

concatenates the position and its derivatives up to order $n-1$. The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ describe the time invariant response of the system to an input command $\boldsymbol{u}_t$. For the examples presented here, we utilize a chain of $n$ integrators commanded by its $n$-th order derivatives. The system matrices for the continuous version of this system are then given by:

$$\boldsymbol{A}^c = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \quad , \quad \boldsymbol{B}^c = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \boldsymbol{I} \end{bmatrix} , \tag{2.3}$$

with $\boldsymbol{I}$ a $D \times D$ identity matrix. The discrete time versions of the system matrices can be computed with a forward Euler discretization:

$$\boldsymbol{A} = \Delta t \boldsymbol{A}^c + \boldsymbol{I} \quad \text{and} \quad \boldsymbol{B} = \Delta t \boldsymbol{B}^c, \tag{2.4}$$

where $\Delta t$ is the duration of one time step, or with higher order approximation methods such as zero order hold (ZOH).

The positions along the trajectory are then given by:

$$\boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{x}_t, \quad \text{where} \quad \boldsymbol{C} = \left[\boldsymbol{I}, \mathbf{0}, \ldots, \mathbf{0}, \mathbf{0}\right]. \tag{2.5}$$

From a control perspective, the sensor matrix $\boldsymbol{C}$ determines what elements of the state are observed in a feedback system. For our use case of curve generation, this

formulation of $\boldsymbol{C}$ limits the parameters of our method to the position components of the state, which greatly simplifies the user interaction with the method.

### 2.3.2 Optimization Objective

We generate a trajectory of $T$ time steps by computing an optimal controller that minimizes a quadratic cost, which penalizes a trade-off between deviations from a reference state sequence $\{\bar{\boldsymbol{x}}_t\}_{t=1}^{T}$ (*tracking cost*) and the magnitude of a control command sequence $\{\boldsymbol{u}_t\}_{t=1}^{T-1}$ (*control cost*). The optimization objective is expressed with the cost:

$$J = \sum_{t=1}^{T} (\bar{\boldsymbol{x}}_t - \boldsymbol{x}_t)^{\top} \boldsymbol{Q}_t (\bar{\boldsymbol{x}}_t - \boldsymbol{x}_t) + \sum_{t=1}^{T-1} \boldsymbol{u}_t^{\top} \boldsymbol{R}_t \boldsymbol{u}_t, \tag{2.6}$$

subject to the constraint of the linear system defined in Eq. (2.1), with $\boldsymbol{Q}_t$ and $\boldsymbol{R}_t$ being positive semi-definite weight matrices that determine the tracking and control penalties for each time step. The linear constraint guarantees that the output of the method is a trajectory that has continuous derivatives up to order $n - 1$.

The combination of a linear system with this type of optimization objective is commonly used in process control and robotics applications, where it is known as discrete linear quadratic tracking (LQT) and corresponds to the quadratic cost case of model predictive control (MPC) [45]. This results in a standard optimization problem that can be solved iteratively or in batch form and produces an optimal controller or control command sequence. In typical control settings, the optimization is performed iteratively over a time horizon of observations and is thus commonly known as receding horizon control. However, for the intended use case of curve design, we can apply the optimization to the full duration of the desired trajectory. With the appropriate formulation of the reference, this results in a flexible curve generation method that can be used similarly to the more conventional ones.

### 2.3.3 Tracking Formulation

We formulate the reference state and weights for the optimization objective, by pairing each input Gaussian with an activation function:

$$h_i(t) = \frac{\phi_i(t)}{\sum_{j=1}^{m} \phi_j(t) + \epsilon}, \quad \text{with} \quad \phi_i(t) = \exp\left(-\frac{(t - \tau_i)^2}{2\sigma^2}\right), \tag{2.7}$$

where $\tau_i$ defines the *passage time* for the state, $\sigma$ is a global parameter which defines the *time interval* covered by each state, and $\epsilon$ is an arbitrarily small value that avoids divisions by 0.

With an assumption of local isochrony, we define the passage times for each state at equidistant time steps with: $\tau_{i+1} - \tau_i = T/(M-1)$, and with $\tau_1 = 1$ and $\tau_M = T$.

The reference states and weights are then generated by assigning to each time step the state for which $h_i(t) > 0.5$ (Fig. 2.3, second row) with:

$$\bar{\boldsymbol{x}}_i = \boldsymbol{C}^\top \boldsymbol{\mu}_i \quad \text{and} \quad \boldsymbol{Q}_t = \boldsymbol{C}^\top \boldsymbol{\Sigma}_i^{-1} \boldsymbol{C}. \tag{2.8}$$

With this formulation, the derivatives of the trajectory are fully determined by the optimization procedure, which is expressed by setting the corresponding precision terms $\boldsymbol{Q}_t$ to zero. Intuitively, a zero entry in $\boldsymbol{Q}_t$ means that the optimization has no precision requirements for the corresponding state entry and thus is free to enforce the smoothness requirement expressed in the second term of the cost function. In typical applications, the tracking weights $\boldsymbol{Q}_t$ are defined as diagonal matrices. This corresponds to a penalty in terms of the Euclidean distance to a reference state. In our stochastic formulation, the weights are expressed as full precision matrices, which correspond to a penalty in terms of the Mahalanobis distance to the reference state. When it is desirable to force the movement to a full stop, this can be done by setting $\boldsymbol{Q}_N = \boldsymbol{I}$ and all the derivative terms in $\bar{\boldsymbol{x}}_N$ to zero.

Increasing the value of $\sigma$ increases the time interval covered by a state, with $\sigma = T/(4(M-1))$ resulting in a stepwise reference that fully covers the time steps of the trajectory (Fig. 2.3c). This increases the influence of the GMM covariances on the resulting trajectory and allows a user to specify curvilinear trends and variability for longer segments of the trajectory. As the parameter $\sigma$ tends to zero, $\phi_i(t)$ will converge to a delta function (Fig. 2.3a), which will result in $\boldsymbol{Q}_t$ being non-zero

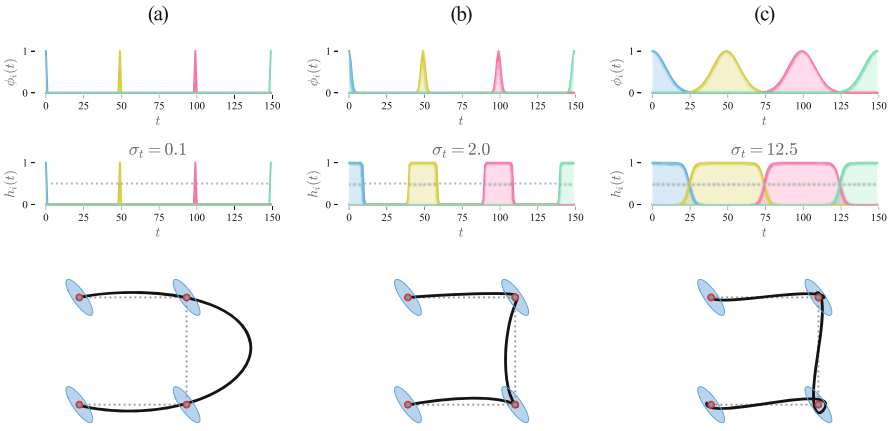



**Fig. 2.3** Effect of three different activation sequences with the same set of Gaussians

only in correspondence with each passage time $\tau_i$. This will result in smoother trajectories that interpolate the key-points. In general, a lower time interval will result in sparser tracking cost in the objective. This increases the influence of the control cost and potentially facilitates the addition of objectives and constraints to the optimization.

### 2.3.4 Stochastic Solution

The optimal trajectory can be retrieved iteratively using (1) dynamic programming [4, 7] or (2) in a batch form by solving a large ridge regression problem. Here we describe the latter, which results in a more compact solution and additional flexibility such as a straightforward probabilistic interpretation of the result. To compute the solution, we exploit the time invariance of the system and express all future states as a function of the initial state $\bar{\boldsymbol{x}}_1$ with:

$$\boldsymbol{x} = \boldsymbol{S}^{\boldsymbol{x}}\bar{\boldsymbol{x}}_1 + \boldsymbol{S}^{\boldsymbol{u}}\boldsymbol{u}\ , \tag{2.9}$$

where

$$\boldsymbol{S}^{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^N \end{bmatrix} \quad \text{and} \quad \boldsymbol{S}^{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{B} & \boldsymbol{B} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{N-1}\boldsymbol{B} & \boldsymbol{A}^{N-2}\boldsymbol{B} & \dots & \boldsymbol{B} \end{bmatrix}. \tag{2.10}$$

We then express the objective (2.6) in matrix form as:

$$J = (\bar{\boldsymbol{x}} - \boldsymbol{x})^\top\, \boldsymbol{Q}\, (\bar{\boldsymbol{x}} - \boldsymbol{x}) + \boldsymbol{u}^\top \boldsymbol{R}\boldsymbol{u}\ , \tag{2.11}$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ are large block matrices with $\boldsymbol{Q}_t$ and $\boldsymbol{R}_t$ along their block diagonals, while $\bar{\boldsymbol{x}}$, $\boldsymbol{x}$, and $\boldsymbol{u}$ are column vectors representing the reference, state, and control commands, this for each time step. Substituting (2.9) into (2.11), differentiating with respect to $\boldsymbol{u}$, and setting to zero result in a ridge regression solution of the form:

$$\boldsymbol{u} = \underbrace{\left(\left(\boldsymbol{S}^{\boldsymbol{u}}\right)^\top \boldsymbol{Q}\boldsymbol{S}^{\boldsymbol{u}} + \boldsymbol{R}\right)^{-1}}_{\boldsymbol{\Sigma}^{\boldsymbol{u}}} \left(\boldsymbol{S}^{\boldsymbol{u}}\right)^\top \boldsymbol{Q}\left(\bar{\boldsymbol{x}} - \boldsymbol{S}^{\boldsymbol{x}}\bar{\boldsymbol{x}}_1\right)\ , \tag{2.12}$$

which is then substituted back into (2.9) to generate a trajectory.

From Eq. (2.12), we can see that $\boldsymbol{R}$ effectively acts as a Tikhonov regularization term in the least squares solution, resulting in a global smoothing effect on the generated trajectory.

From a probabilistic perspective, $\boldsymbol{R}$ corresponds to a Gaussian prior on the deviations of the control commands from $\mathbf{0}$. The minimization of Eq. (2.11) can then be interpreted as the product of two Gaussians:

$$\mathcal{N}\big(\boldsymbol{u}, \boldsymbol{\Sigma}^{\boldsymbol{u}}\big) \propto \mathcal{N}\Big(\big(\boldsymbol{S}^{\boldsymbol{u}}\big)^{\dagger}\,(\bar{\boldsymbol{x}} - \boldsymbol{S}^{\boldsymbol{x}}\bar{\boldsymbol{x}}_1), \big(\boldsymbol{S}^{\boldsymbol{u}}\big)^{\top}\,\boldsymbol{Q}\boldsymbol{S}^{\boldsymbol{u}}\Big)\,\mathcal{N}(\mathbf{0}, \boldsymbol{R})\ , \tag{2.13}$$

describing a distribution of control commands with center $\boldsymbol{u}$ and covariance $\boldsymbol{\Sigma}^{\boldsymbol{u}}$. By using the linear relation (2.9), the distribution in control space can also be interpreted as a *trajectory distribution*:

$$\mathcal{N}\left(\boldsymbol{x}, \boldsymbol{\Sigma}^{\boldsymbol{x}}\right) \quad \text{with} \quad \boldsymbol{\Sigma}^{\boldsymbol{x}} = \boldsymbol{S}^{\boldsymbol{u}}\boldsymbol{\Sigma}^{\boldsymbol{u}}\left(\boldsymbol{S}^{\boldsymbol{u}}\right)^{\top}\ . \tag{2.14}$$

This formulation results in a generative model of trajectories, which can be used to generate variations that are similar to the ones that would be seen in multiple instances of human writing or drawing (Fig. 2.4). Because of its lower dimensionality, it is preferable to generate variations at the control level, which can be done by exploiting the eigendecomposition:

$$\boldsymbol{\Sigma}^{\boldsymbol{u}} = \boldsymbol{V}^{\boldsymbol{u}}\boldsymbol{D}^{\boldsymbol{u}}\left(\boldsymbol{V}^{\boldsymbol{u}}\right)^{\top}, \tag{2.15}$$

where $\boldsymbol{V}^{\boldsymbol{u}}$ denotes a matrix with all eigenvectors along the columns and $\boldsymbol{D}^{\boldsymbol{u}}$ denotes a matrix with the corresponding eigenvalues along its diagonal. We can then generate samples around the average commands sequence $u$ with:

$$\boldsymbol{u}' \ \sim\ \boldsymbol{u} + \boldsymbol{V}^{\boldsymbol{u}}\left(\boldsymbol{D}^{\boldsymbol{u}}\right)^{\frac{1}{2}}\mathcal{N}\left(\mathbf{0}, \sigma^{u}\boldsymbol{I}\right), \tag{2.16}$$

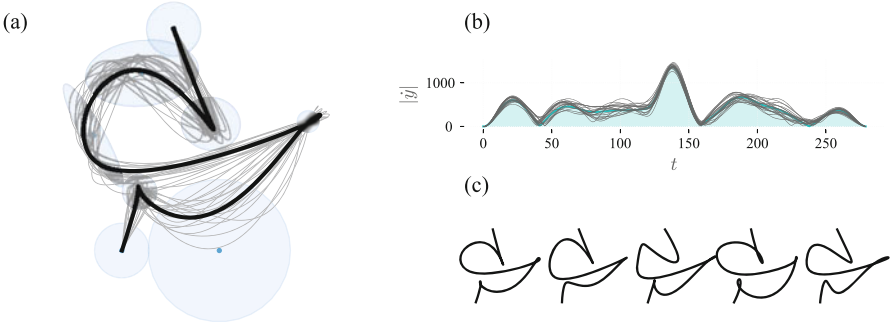



**Fig. 2.4** Stochastic sampling. (**a**) GMM with corresponding trajectory (dark thick race) overlaid together with samples from the trajectory distribution (light gray traces). (**b**) Corresponding sampled speed profiles. (**c**) A few samples selected from the trajectory distribution

where $\sigma^u$ is a user-defined parameter to select the desired sample variation. The resulting trajectories can then easily be retrieved by fitting in the selected samples $\boldsymbol{u}'$ back into Eq. (2.9), see Fig. 2.4 for the results.

### 2.3.5 Periodic Motions

In order to generate periodic motions (Fig. 2.5), we can reformulate the LQT objective with the addition of an equality constraint on the initial and final states of the trajectory. This can be formulated with the linear relation:

$$\boldsymbol{K}\boldsymbol{x} = \boldsymbol{K}\left(\boldsymbol{S}^{\boldsymbol{x}}\bar{\boldsymbol{x}}_1 + \boldsymbol{S}^{\boldsymbol{u}}\boldsymbol{u}\right) = \boldsymbol{0}, \tag{2.17}$$

with $\boldsymbol{K}$ a matrix with zero blocks for each time step apart for the ones corresponding to the states desired to be equal. Adding this constraint to Eq. (2.11) results in the Lagrangian:

$$\mathcal{L}(\boldsymbol{u}, \boldsymbol{\lambda}) = J + \boldsymbol{\lambda}^\top \boldsymbol{K}\boldsymbol{x}. \tag{2.18}$$

Differentiating for $\boldsymbol{u}$ and the Lagrange multipliers $\boldsymbol{\lambda}$ and then equating to $\boldsymbol{0}$ results in the following constrained solution, in matrix form:

$$\begin{bmatrix} \boldsymbol{u} \\ \hat{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{\Sigma}^{\boldsymbol{u}})^{-1} & (\boldsymbol{S}^{\boldsymbol{u}})^\top \boldsymbol{K}^\top \\ \boldsymbol{K}\boldsymbol{S}^{\boldsymbol{u}} & \boldsymbol{0} \end{bmatrix}^{-1} \begin{bmatrix} (\boldsymbol{S}^{\boldsymbol{u}})^\top \boldsymbol{Q}\,(\bar{\boldsymbol{x}} - \boldsymbol{S}^{\boldsymbol{x}}\bar{\boldsymbol{x}}_1) \\ \boldsymbol{0} \end{bmatrix}. \tag{2.19}$$

We observe that in order to generate periodic motions that are symmetric, it is convenient to utilize a *wrapped* version of the input components as a new input. To do so we repeat a subsequence of $w$ components at the start and end of the wrapped sequence with the indices of the original sequence organized as follows:

$$[M - w, \ldots, M, 1, \ldots, M, 1, \ldots, w], \tag{2.20}$$

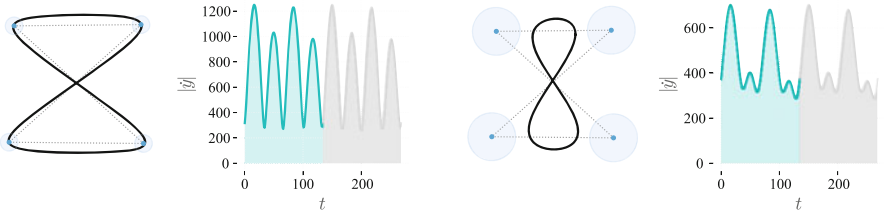



**Fig. 2.5** Periodic motions using Gaussians with different variances. The speed profiles are repeated (in light gray) to visualize the periodicity of the speed profile

where $M$ is the selected number of Gaussians. This produces a new sequence of $M^{\circ}$ Gaussians for the periodic motion, giving the following passage time sequence:

$$[\tau_{-w}, \tau_{-w+1}, \ldots, \tau_{M+1+w}]\,. \tag{2.21}$$

This results in a wrapped reference $\boldsymbol{Q}$ and $\bar{\boldsymbol{x}}$ that is constructed as described in Sect. 2.3.3. The linear constraint matrix is then given by:

$$\boldsymbol{K} = \Big[\mathbf{0}, \ldots, \underset{\tau_1}{\boldsymbol{I}}, \mathbf{0}, \ldots, \underset{\tau_{m+1}}{-\boldsymbol{I}}, \mathbf{0}, \ldots\Big]^{\top}. \tag{2.22}$$

The periodic trajectory is finally computed by plugging the command sequence $\boldsymbol{u}$ computed with Eq. (2.19) into Eq. (2.11), and then considering the subset of the trajectory defined between time steps $\tau_1$ and $\tau_{M+1}$.

## 2.4 User Interface

The proposed trajectory generation method is efficient and is well suited for interactive design applications. It is easy to drag the centers of the input Gaussians with a typical point-and-click procedure, and it is also easy to interactively manipulate the covariances. For example, this can be done by manipulating an ellipsoid, such that its center defines the mean $\boldsymbol{\mu}_i$, and the axes are used to manipulate the covariance $\boldsymbol{\Sigma}_i$ through its eigendecomposition. The latter can be described with:

$$\boldsymbol{\Sigma}_i = \boldsymbol{\Theta}_i \boldsymbol{S}_i^2 \boldsymbol{\Theta}_i^{\top}, \tag{2.23}$$

where $\boldsymbol{\Theta}_i$ corresponds to an orthogonal (rotation) matrix, and $\boldsymbol{S}_i$ is a scaling matrix. Here, we describe the 2D case in which the rotation and scaling matrices are given by:

$$\boldsymbol{\Theta}_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix}, \quad \theta_i = \tan^{-1}\frac{a_2}{a_1}, \quad \boldsymbol{S}_i = \begin{bmatrix} \frac{\|\boldsymbol{a}\|}{2} & 0 \\ 0 & \frac{\|\boldsymbol{b}\|}{2} \end{bmatrix}, \tag{2.24}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are the major and minor axes of an ellipse, which can be interactively dragged to manipulate the shape of the distribution (Fig. 2.6, left). While the examples given are two dimensional, an extension to three-dimensional ellipsoids is straightforward to implement with a so-called *arc-ball* interface [38].

The trajectories generated by our system are sequences of points, the resolution of which depends on the discretization time step $\Delta t$. The distance between consecutive points is not constant and reflects the smooth and physiologically plausible kinematics generated by the model. As a result, it is easy to generate natural looking stroke animations by incrementally sweeping a brush texture along

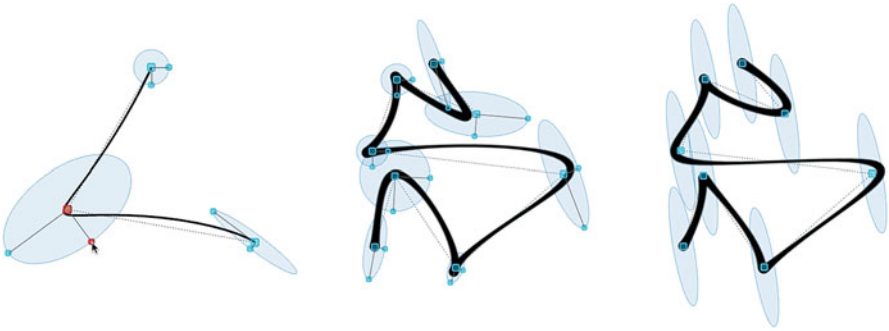


**Fig. 2.6** Examples of user interactions by the manipulation of ellipsoids representing GMMs, and resulting in various animated brush rendering effects

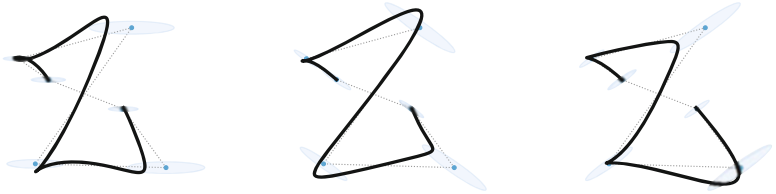


**Fig. 2.7** Three different stylizations of a letter "Z" using semi-tied covariances with different shared orientations

the points of the trajectory [5]. To increase the sense of dynamism, we slightly vary the brush size at a degree inversely proportional to the trajectory speed, which mimics the effect of more ink being deposited on a surface when the movement is slower (Fig. 2.6).

### *2.4.1 Semi-tied Structure*

In the previous paragraphs, we have seen that it is possible for a user to easily edit the shape and position of each Gaussian. For applications aimed at procedural content generation, it may be desirable to formulate a more parsimonious way of generating trajectories, in which different stylizations are generated without having to specify the covariance of each GMM component. We observe that one convenient way to provide the user this facility is to enforce a *shared orientation* for all covariance ellipsoids, which can easily be achieved with the formulation above by having the orientations $\mathbf{\Theta}_i$ set to the same value. This results in a *semi-tied* covariance structure of the input GMMs, in which all covariances share the same eigenvectors but not necessarily the same eigenvalues (Fig. 2.7).

From a motor control perspective, the semi-tied formalism can be interpreted as the alignment of different movement parts/primitives with a shared coordination

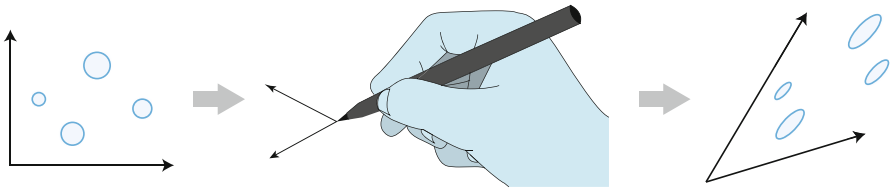


**Fig. 2.8** Illustrative example of how an oblique coordinate system could result from fine movements in handwriting, when using fingers and wrist only

pattern [39], which is in line with the hypothesis of postural-synergies at the motor planning level [9]. This implies a shared non-orthogonal (oblique) basis for all the covariances, which produces a shear transformation that in the 2D case transforms a circle into an oriented ellipse. Oblique coordinates have also been suggested to describe the coordination of handwriting movements made with the fingers and wrist only [11], which suggests another possible bio-physical interpretation of this result (Fig. 2.8).

With this simplified interface, it is possible to explore different stylizations of a key-point sequence with a reduced set of open parameters. The semi-tied covariances enforce a coupling between the coordinates of the trajectory, which results in an observable sense of coordination in the movement. At the same time, minimization of the control command amplitude produces smooth trajectories that evoke a natural drawing movement.

## 2.5 Conclusions

We have presented a method for the generation of smooth curves and motion trajectories with a stochastic formulation of optimal control. The output of our method is a trajectory distribution, which describes a family of motion paths that can mimic the appearance and the variability of human-made artistic traces. Each trajectory generated by our method reflects a movement with physiologically plausible kinematics. This can be exploited to produce rendering effects, realistic animations or also to drive the smooth motion of a robotic arm [3]. The input to the method is a sparse sequence of multivariate Gaussians that determine the overall shape of the output and explicitly define its variability. This results in a representation that is similar to the one used in conventional GCAD applications, and that can be edited interactively in a similar manner.

For our use case, we let the user explicitly define the GMM components. However, a similar representation can be learned from data with standard maximum-likelihood estimation methods [7]. Our choice of Gaussians as an input and output distribution is principally motivated by its effectiveness and simplicity of representation. From a user-interaction perspective, this allows users to intuitively

manipulate the input distributions by modifying the axes of each GMM component ellipsoid (Fig. 2.6). Furthermore, the straightforward relation of Gaussians to linear systems quadratic error terms allows us to solve the optimal control problem interactively and in closed form, while also offering a stochastic interpretation of the output. Extending the proposed method to non-linear dynamical systems and to distributions other than Gaussians is an interesting avenue of future research.

The curve generation method presented in this chapter is principally developed with creative computer graphics applications in mind, especially those that require mimicking the visual quality of traces observed in artistic applications of calligraphy and graffiti. There is no specific consensus on a metric that can be used to aesthetically evaluate the quality of visual traces or marks, and for a human this may depend on subjective factors such as cultural and educational background. However, there is growing psychological and neuro-science evidence suggesting that the observation of a static trace resulting from a human-made movement triggers a mental recovery of the movement underlying its production [20, 29, 35] and that such recovery influences its aesthetic appreciation [28]. As a result, we hypothesize that synthesizing curvilinear traces with kinematics similar to the ones made by a human may trigger similar responses in an observer. Hence, another promising line of future work would be to study the responses of human observers to traces generated with different parameters of the system, as well as evaluate how their artistic expertise or cultural background may influence aesthetic judgment.

Finally, we note that, while in this chapter we focused on the generation of 2D trajectories, the proposed method can naturally be generalized to higher dimensions. We envisage useful future applications, in particular in developing an interface for 3D trajectories, as well as for taking into consideration additional features such as a drawing tool orientation or the effects of varying the force applied along a trajectory.

## References

1. AlMeraj, Z., Wyvill, B., Isenberg, T., Gooch, A., Guy, R.: Automatically mimicking unique hand-drawn pencil lines. Comput. Graph. **33**(4), 496–508 (2009)
2. Baran, I., Lehtinen, J., Popović, J.: Sketching clothoid splines using shortest paths. In: Computer Graphics Forum, vol. 29, pp. 655–664. Wiley, London (2010)
3. Berio, D., Calinon, S., Fol Leymarie, F.: Learning dynamic graffiti strokes with a compliant robot. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3981–3986. IEEE, Piscataway (2016)
4. Berio, D., Calinon, S., Fol Leymarie, F.: Generating calligraphic trajectories with model predictive control. In: Proceedings of the 43rd Conference on Graphics Interface, pp. 132–139. Canadian Human-Computer Communications Society School of Computer Science, Waterloo (2017)
5. Berio, D., Leymarie, F.F., Plamondon, R.: Expressive curve editing with the sigma lognormal model. In: Diamanti, O., Vaxman, A. (eds.) EG 2018—Short Papers. The Eurographics Association (2018)
6. Bernstein, N.A., Latash, M.L., Turvey, M.: Dexterity and Its Development. Taylor & Francis, London (1996)

7. Calinon, S.: A tutorial on task-parameterized movement learning and retrieval. Intell. Serv. Robot. **9**(1), 1–29 (2016)
8. Cooper, M., Chalfant, H.: Subway Art. Rinehart and Winston, Holt (1984)
9. d'Avella, A., Saltiel, P., Bizzi, E.: Combinations of muscle synergies in the construction of a natural motor behavior. Nat. Neurosci. **6**(3), 300–308 (2003)
10. De Boor, C.: A Practical Guide to Splines, vol. 27. Springer, New York (1978)
11. Dooijes, E.: Analysis of handwriting movements. Acta Psychol. **54**(1), 99–114 (1983)
12. Edelman, S., Flash, T.: A model of handwriting. Biol. Cybern. **57**(1–2), 25–36 (1987)
13. Egerstedt, M., Martin, C.: Control Theoretic Splines: Optimal Control, Statistics, and Path Planning. Princeton University Press; Princeton Oxford, Princeton (2009)
14. Ferrer, M.A., Diaz, M., Carmona-Duarte, C., Morales, A.: A behavioral handwriting model for static and dynamic signature synthesis. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1041–1053 (2017)
15. Flash, T., Handzel, A.: Affine differential geometry analysis of human arm movements. Biol. Cybern. **96**(6), 577–601 (2007)
16. Flash, T., Henis, E.: Arm trajectory modifications during reaching towards visual targets. J. Cogn. Neurosci. **3**(3), 220–230 (1991)
17. Flash, T., Hochner, B.: Motor primitives in vertebrates and invertebrates. Curr. Opin. Neurobiol. **15**(6), 660–666 (2005)
18. Flash, T., Hogan, N.: The coordination of arm movements. J. Neurosci. **5**(7), 1688–1703 (1985)
19. Flash, T., Hogan, N.: Optimization principles in motor control. In: The Handbook of Brain Theory and Neural Networks, pp. 682–685. MIT Press, Cambridge, MA (1998)
20. Freedberg, D., Gallese, V.: Motion, emotion and empathy in esthetic experience. Trends Cogn. Sci. **11**(5), 197–203 (2007)
21. Freeman, F.: Experimental analysis of the writing movement. Psychol. Monogr. Gen. Appl. **17**(4), 1–57 (1914)
22. Fujioka, H., Kano, H., Nakata, H., Shinoda, H.: Constructing and reconstructing characters, words, and sentences by synthesizing writing motions. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **36**(4), 661–670 (2006)
23. Haeberli, P.: Dynadraw: A dynamic drawing technique. www.graficaobscura.com/dyna/ (1989)
24. House, D., Singh, M.: Line drawing as a dynamic process. In: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications, pp. 351–60. IEEE, Piscataway (2007)
25. Jordan, M., Wolpert, D.: Computational motor control. In: Gazzaniga, M. (ed.) The Cognitive Neurosciences, 2nd edn. MIT Press, Cambridge, MA (1999)
26. Kyprianidis, J., Collomosse, J., Wang, T., Isenberg, T.: State of the "art": A taxonomy of artistic stylization techniques for images and video. IEEE Trans. Vis. Comput. Graph. **19**(5), 866–885 (2013)
27. Lacquaniti, F., Terzuolo, C., Viviani, P.: The law relating the kinematic and figural aspects of drawing movements. Acta Psychol. **54**(1), 115–130 (1983)
28. Leder, H., Bär, S., Topolinski, S.: Covert painting simulations influence aesthetic appreciation of artworks. Psychol. Sci. **23**(12), 1479–1481 (2012)
29. Longcamp, M., Anton, J.L., Roth, M., Velay, J.L.: Visual presentation of single letters activates a premotor area involved in writing. NeuroImage **19**(4), 1492–1500 (2003)
30. Lu, J., Yu, F., Finkelstein, A., DiVerdi, S.: Helpinghand: Example-based stroke stylization. ACM Trans. Graph. **31**(4), 46 (2012)
31. McCrae, J., Singh, K.: Sketching piecewise clothoid curves. Comput. Graph. **33**(4), 452–461 (2009)
32. Mediavilla, C., Marshall, A., van Stone, M., Xuriguera, G., Jackson, D.: Calligraphy: from calligraphy to abstract painting. Scirpus (1996)
33. Meirovitch, Y., Bennequin, D., Flash, T.: Geometrical invariance and smoothness maximization for task-space movement generation. IEEE Trans. Robot. **32**(4), 837–853 (2016)
34. Morasso, P.: Spatial control of arm movements. Exp. Brain Res. **42**(2), 223–7 (1981)
35. Pignocchi, A.: How the intentions of the draftsman shape perception of a drawing. Conscious. Cogn. **19**(4), 887–898 (2010)

36. Plamondon, R.: A kinematic theory of rapid human movements. Part I. Biol. Cybern. **72**(4), 295–307 (1995)
37. Plamondon, R., O'Reilly, C., Galbally, J., Almaksour, A., Anquetil, É.: Recent developments in the study of rapid human movements with the kinematic theory. Pattern Recogn. Lett. **35**, 225–35 (2014)
38. Shoemake, K.: Arcball: A user interface for specifying three-dimensional orientation using a mouse. In: Graphics Interface, vol. 92, pp. 151–156 (1992)
39. Tanwani, A., Calinon, S.: Learning robot manipulation tasks with task-parameterized semitied hidden semi-Markov model. IEEE Robot. Autom. Lett. **1**(1), 235–242 (2016)
40. Thiel, Y., Singh, K., Balakrishnan, R.: Elasticurves: Exploiting stroke dynamics and inertia for the real-time neatening of sketched 2D curves. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, pp. 383–392. ACM, New York (2011)
41. Todorov, E., Jordan, M.: Optimal feedback control as a theory of motor coordination. Nat. Neurosci. **5**(11), 1226–1235 (2002)
42. Todorov, E., Jordan, M.I.: Smoothness maximization along a predefined path accurately predicts the speed profiles of complex arm movements. J. Neurophysiol. **80**(2), 696–714 (1998)
43. Uno, Y., Kawato, M., Suzuki, R.: Formation and control of optimal trajectory in human multijoint arm movement. Biol. Cybern. **61**(2), 89–101 (1989)
44. Viviani, P., Schneider, R.: A developmental study of the relationship between geometry and kinematics in drawing movements. J. Exp. Psychol. Hum. Percept. Perform. **17**(1), 198–218 (1991)
45. Zeestraten, M., Calinon, S., Caldwell, D.G.: Variable duration movement encoding with minimal intervention control. In: Proceedings of the International Conference on Robotics and Automation (ICRA), pp. 497–503. IEEE, Stockholm, Sweden (2016)

# Chapter 3
# Mixture Models for the Analysis, Edition, and Synthesis of Continuous Time Series


**Sylvain Calinon**

**Abstract** This chapter presents an overview of techniques used for the analysis, edition, and synthesis of continuous time series, with a particular emphasis on motion data. The use of mixture models allows the decomposition of time signals as a superposition of basis functions. It provides a compact representation that aims at keeping the essential characteristics of the signals. Various types of basis functions have been proposed, with developments originating from different fields of research, including computer graphics, human motion science, robotics, control, and neuroscience. Examples of applications with radial, Bernstein, and Fourier basis functions are presented, with associated source codes to get familiar with these techniques.

## 3.1 Introduction

The development of techniques to process continuous time series is required in various domains of application, including computer graphics, human motion science, robotics, control, and neuroscience. These techniques need to cover various purposes, including the encoding, modeling, analysis, edition, and synthesis of time series (sometimes needed simultaneously). The development of these techniques is also often governed by additional important constraints such as interpretability and reproducibility. These heavy requirements motivate the use of mixture models, effectively leveraging the formalism and ubiquity of these models.

The first part of this chapter reviews decomposition techniques based on radial basis functions (RBFs) and locally weighted regression (LWR). The connections between LWR and Gaussian mixture regression (GMR) are discussed, based on the encoding of time series as Gaussian mixture models (GMMs). I will show how this mixture modeling principle can be extended to a weighted superposition of

S. Calinon (✉)
Idiap Research Institute, Martigny, Switzerland
e-mail: sylvain.calinon@idiap.ch



N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6_3

Bernstein basis functions, often known as Bézier curves. The aim is to examine the connections with mixture models and to highlight the generative aspects of these techniques. In particular, this link exposes the possibility of representing Bézier curves with higher order Bernstein polynomials. I then discuss the decomposition of time signals as Fourier basis functions, by showing how a mixture of Gaussians can leverage the multivariate Gaussian properties in the spatial and frequency domains. Finally, I show that these different decomposition techniques can be represented as time series distributions through a probabilistic movement primitives representation.

Pointers to various practical applications are provided for further readings, including the analysis of biological signals in the form of multivariate continuous time series, the development of computer graphics interfaces to edit trajectories and motion paths for manufacturing robots, the analysis and synthesis of periodic human gait data, or the generation of exploratory movements in mobile platforms with ergodic control.

The techniques presented in this chapter are described with a uniform notation that does not necessarily follow the original notation. The goal is to tie links between these different techniques, which are often presented in isolation of the more general context of mixture models. MATLAB codes accompany the chapter [25], with full compatibility with GNU Octave.

## 3.2 Movement Primitives

The term *movement primitives* refers to an organization of continuous motion signals in the form of a superposition in parallel and in series of simpler signals, which can be viewed as "building blocks" to create more complex movements, see Fig. 3.1. This principle, coined in the context of motor control [23], remains valid for a wide range of continuous time signals (for both analysis and synthesis). Next, I present three popular families of basis functions that can be employed for time series decomposition.

### *3.2.1 Radial Basis Functions (RBFs)*

Radial basis functions (RBFs) are ubiquitous in continuous time series encoding [28], notably due to their simplicity and ease of implementation. Most algorithms exploiting this representation rely on some form of regression, often related to locally weighted regression (LWR), which was introduced by Cleveland [8] in statistics and popularized by Atkeson [3] in robotics. By representing, respectively, $N$ input and output datapoints as $\boldsymbol{X}^I = [\boldsymbol{x}_1^I, \boldsymbol{x}_2^I, \dots, \boldsymbol{x}_N^I]^\top$ and $\boldsymbol{X}^O = [\boldsymbol{x}_1^O, \boldsymbol{x}_2^O, \dots, \boldsymbol{x}_N^O]^\top$, we are interested in the problem of finding a matrix $\boldsymbol{A}$ so that $\boldsymbol{A}\boldsymbol{X}^I$ would match $\boldsymbol{X}^O$ by considering different weights on the input–output datapoints $\{\boldsymbol{X}^I, \boldsymbol{X}^O\}$ (namely some datapoints are more informative than others for

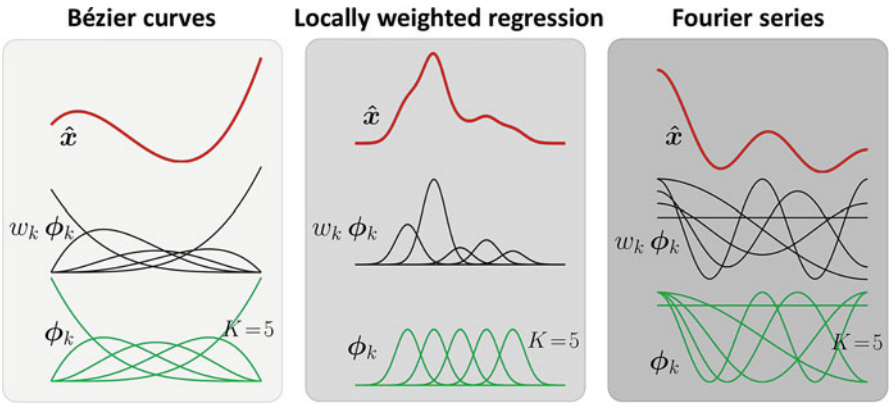



**Fig. 3.1** Motion primitives with different basis functions $\phi_k$, where a unidimensional time series $\hat{\boldsymbol{x}} = \sum_{k=1}^{K} w_k \phi_k$ is constructed as a weighted superposition of $K$ signals $\phi_k$

the estimation of $\boldsymbol{A}$). A weighted least squares estimate $\hat{\boldsymbol{A}}$ can be found by solving the objective

$$\begin{aligned}\hat{\boldsymbol{A}} &= \arg\min_{\boldsymbol{A}} \operatorname{tr}\Big((\boldsymbol{X}^O - \boldsymbol{X}^I\boldsymbol{A})^\top \boldsymbol{W}(\boldsymbol{X}^O - \boldsymbol{X}^I\boldsymbol{A})\Big) \\ &= (\boldsymbol{X}^{I\top}\boldsymbol{W}\boldsymbol{X}^I)^{-1}\boldsymbol{X}^{I\top}\boldsymbol{W}\,\boldsymbol{X}^O, \end{aligned} \tag{3.1}$$

where $\boldsymbol{W} \in \mathbb{R}^{N\times N}$ is a weighting matrix. Locally weighted regression (LWR) is a direct extension of the weighted least squares formulation in which $K$ weighted regressions are performed on the same dataset $\{\boldsymbol{X}^I, \boldsymbol{X}^O\}$. It aims at splitting a nonlinear problem so that it can be solved locally by linear regression. LWR computes $K$ estimates $\hat{\boldsymbol{A}}_k$, each with a different function $\phi_k(\boldsymbol{x}_n^I)$, classically defined as the radial basis functions

$$\tilde{\phi}_k(\boldsymbol{x}_n^I) = \exp\Big(-\frac{1}{2}(\boldsymbol{x}_n^I - \boldsymbol{\mu}_k^I)^\top \boldsymbol{\Sigma}_k^{I\,-1}(\boldsymbol{x}_n^I - \boldsymbol{\mu}_k^I)\Big), \tag{3.2}$$

where $\boldsymbol{\mu}_k^I$ and $\boldsymbol{\Sigma}_k^I$ are the parameters of the $k$-th RBF, or in its rescaled form[1]

$$\phi_k(\boldsymbol{x}_n^I) = \frac{\tilde{\phi}_k(\boldsymbol{x}_n^I)}{\sum_{i=1}^{K}\tilde{\phi}_i(\boldsymbol{x}_n^I)}. \tag{3.3}$$

An associated diagonal matrix

[1]We will see later that the rescaled form is required for some techniques, but for locally weighted regression, it can be omitted to enforce the independence of the local function approximators.

$$\boldsymbol{W}_k = \text{diag}\Big(\phi_k(\boldsymbol{x}_1^I), \phi_k(\boldsymbol{x}_2^I), \ldots, \phi_k(\boldsymbol{x}_N^I)\Big) \tag{3.4}$$

can be used with (3.1) to evaluate $\hat{\boldsymbol{A}}_k$. The result can then be employed to compute

$$\hat{\boldsymbol{X}}^O = \sum_{k=1}^{K} \boldsymbol{W}_k \boldsymbol{X}^I \hat{\boldsymbol{A}}_k. \tag{3.5}$$

The centroids $\boldsymbol{\mu}_k^I$ in (3.2) are usually set to uniformly cover the input space, and $\boldsymbol{\Sigma}_k^I = \boldsymbol{I}\sigma^2$ is used as a common bandwidth shared by all basis functions. Figure 3.2 shows an example of LWR to encode planar trajectories.

LWR can be directly extended to local least squares polynomial fitting by changing the definition of the inputs. Multiple variants of the above formulation exist, including online estimation with a recursive formulation [27], Bayesian treatments

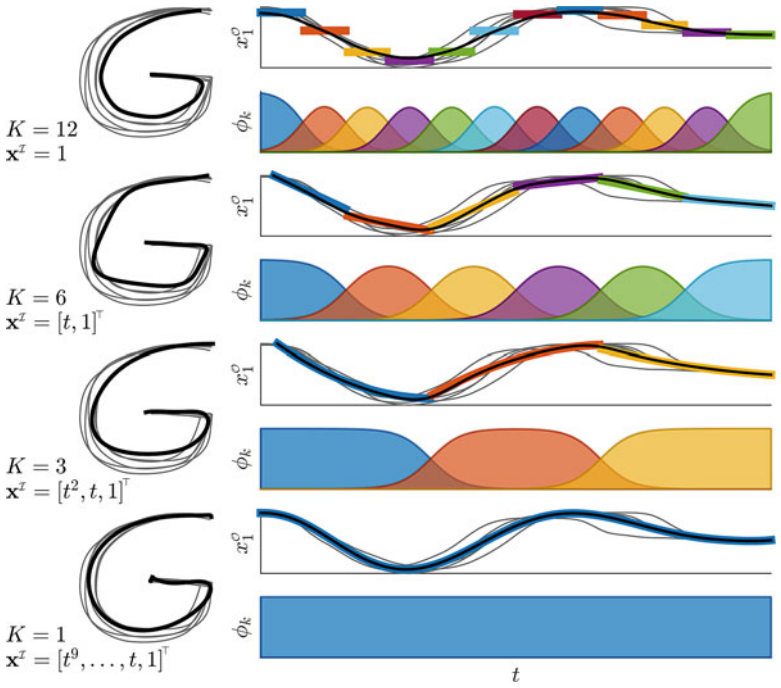



**Fig. 3.2** Polynomial fitting with locally weighted regression (LWR), by considering different degrees of the polynomial and by adapting the number of basis functions accordingly. The top row shows a very localized encoding of the movement with constant values used in (3.1), thus requiring the use of many basis functions to represent the trajectory. The next rows show that a reduction of this number of basis functions typically needs to be compensated with more complex basis functions (polynomial of higher degrees). The bottom row depicts the limit case in which a global encoding of the movement would require a polynomial of high degree

of LWR [31], or extensions such as locally weighted projection regression (LWPR) that exploit partial least squares to cope with redundant or irrelevant inputs [33].

Examples of application range from inverse dynamics modeling [33] to the skillful control of a devil-stick juggling robot [4]. A MATLAB code example `demo_LWR01.m` can be found in [25].

#### 3.2.1.1 Gaussian Mixture Regression (GMR)

Gaussian mixture regression (GMR) is a another popular technique for time series and motion representations [7, 12]. It relies on linear transformation and conditioning properties of multivariate Gaussian distributions. GMR provides a synthesis mechanism to compute output distributions with a computation time independent of the number of datapoints used to train the model. A characteristic of GMR is that it does not model the regression function directly. Instead, it first models the joint probability density of the data in the form of a Gaussian mixture model (GMM). It can then compute the regression function from the learned joint density model, resulting in very fast computation of a conditional distribution.

In GMR, both input and output variables can be multidimensional. Any subset of input–output dimensions can be selected, which can change, if required, at each time step. Thus, any combination of input–output mappings can be considered, where expectations on the remaining dimensions are computed as a multivariate distribution. In the following, we will denote the block decomposition of a datapoint $\boldsymbol{x}_t \in \mathbb{R}^D$ at time step $t$, and the center $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$ of the $k$-th Gaussian in the GMM as

$$\boldsymbol{x}_t = \begin{bmatrix} \boldsymbol{x}_t^I \\ \boldsymbol{x}_t^O \end{bmatrix}, \quad \boldsymbol{\mu}_k = \begin{bmatrix} \boldsymbol{\mu}_k^I \\ \boldsymbol{\mu}_k^O \end{bmatrix}, \quad \boldsymbol{\Sigma}_k = \begin{bmatrix} \boldsymbol{\Sigma}_k^I & \boldsymbol{\Sigma}_k^{IO} \\ \boldsymbol{\Sigma}_k^{OI} & \boldsymbol{\Sigma}_k^O \end{bmatrix}. \tag{3.6}$$

We first consider the example of time-based trajectories by using $\boldsymbol{x}_t^I$ as a time variables. At each time step $t$, $\mathcal{P}(\boldsymbol{x}_t^O|\boldsymbol{x}_t^I)$ can be computed as the multimodal conditional distribution

$$\mathcal{P}(\boldsymbol{x}_t^O|\boldsymbol{x}_t^I) = \sum_{k=1}^{K} h_k(\boldsymbol{x}_t^I)\mathcal{N}\left(\hat{\boldsymbol{\mu}}_k^O(\boldsymbol{x}_t^I), \hat{\boldsymbol{\Sigma}}_k^O\right), \tag{3.7}$$

$$\text{with} \quad \hat{\boldsymbol{\mu}}_k^O(\boldsymbol{x}_t^I) = \boldsymbol{\mu}_k^O + \boldsymbol{\Sigma}_k^{OI}{\boldsymbol{\Sigma}_k^I}^{-1}(\boldsymbol{x}_t^I - \boldsymbol{\mu}_k^I),$$

$$\hat{\boldsymbol{\Sigma}}_k^O = \boldsymbol{\Sigma}_k^O - \boldsymbol{\Sigma}_k^{OI}{\boldsymbol{\Sigma}_k^I}^{-1}\boldsymbol{\Sigma}_k^{IO},$$

$$\text{and} \quad h_k(\boldsymbol{x}_t^I) = \frac{\pi_k\, \mathcal{N}(\boldsymbol{x}_t^I|\, \boldsymbol{\mu}_k^I, \boldsymbol{\Sigma}_k^I)}{\sum_{i=1}^{K} \pi_i\, \mathcal{N}(\boldsymbol{x}_t^I|\, \boldsymbol{\mu}_i^I, \boldsymbol{\Sigma}_i^I)},$$

computed with

$$\mathcal{N}\left(\boldsymbol{x}_t^I|\,\boldsymbol{\mu}_k^I,\boldsymbol{\Sigma}_k^I\right)=(2\pi)^{-\frac{D}{2}}|\boldsymbol{\Sigma}_k^I|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\left(\boldsymbol{x}_t^I-\boldsymbol{\mu}_k^I\right)^\top\boldsymbol{\Sigma}_k^{I\,-1}\left(\boldsymbol{x}_t^I-\boldsymbol{\mu}_k^I\right)\right).$$

When a unimodal output distribution is required, the law of total mean and variance (see Fig. 3.3, *right*) can be used to approximate the distribution with the Gaussian

$$\mathcal{P}(\boldsymbol{x}_t^O|\boldsymbol{x}_t^I)=\mathcal{N}\left(\boldsymbol{x}_t^O|\,\hat{\boldsymbol{\mu}}^O(\boldsymbol{x}_t^I),\hat{\boldsymbol{\Sigma}}^O(\boldsymbol{x}_t^I)\right), \tag{3.8}$$

$$\text{with}\quad \hat{\boldsymbol{\mu}}^O(\boldsymbol{x}_t^I)=\sum_{k=1}^K h_k(\boldsymbol{x}_t^I)\;\hat{\boldsymbol{\mu}}_k^O(\boldsymbol{x}_t^I),$$

$$\text{and}\quad \hat{\boldsymbol{\Sigma}}^O(\boldsymbol{x}_t^I)=\sum_{k=1}^K h_k(\boldsymbol{x}_t^I)\left(\hat{\boldsymbol{\Sigma}}_k^O+\hat{\boldsymbol{\mu}}_k^O(\boldsymbol{x}_t^I)\;\hat{\boldsymbol{\mu}}_k^O(\boldsymbol{x}_t^I)^\top\right)-\hat{\boldsymbol{\mu}}^O(\boldsymbol{x}_t^I)\,\hat{\boldsymbol{\mu}}^O(\boldsymbol{x}_t^I)^\top.$$

Figure 3.3 presents an example of GMR with 1D input and 1D output. With the GMR representation, LWR corresponds to a GMM with diagonal covariances. Expressing LWR in the more general form of GMR has several advantages: (1) it allows the encoding of local correlations between the motion variables by extending the diagonal covariances to full covariances; (2) it provides a principled approach to estimate the parameters of the RBFs, similar to a GMM parameters fitting problem; (3) it often allows a significant reduction of the number of RBFs, because the position and spread of each RBF are also estimated; and (4) the (online) estimation of the mixture model parameters and the model selection problem (automatically estimating the number of basis functions) can readily exploit techniques compatible

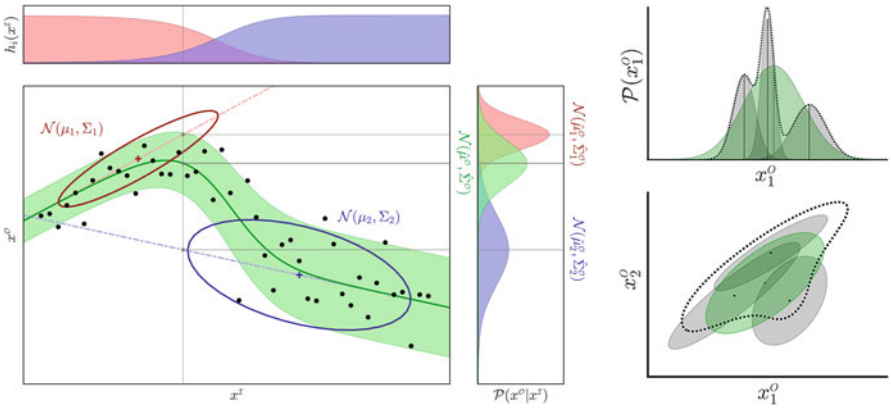



**Fig. 3.3** *Left:* Gaussian mixture regression (GMR) for 1D input $x^I$ and 1D output $x^O$. *Right:* Gaussian that best approximates a mixture of Gaussians. The multimodal distributions in dashed line depict the probability density functions for the mixtures of three Gaussians in gray color (examples in 1D and 2D are depicted). The Gaussians in green color approximate these multimodal distributions

with GMM (Bayesian nonparametrics with Dirichlet processes, spectral clustering, small variance asymptotics, expectation-maximization procedures, etc.).

Another approach to encode and synthesize a movement is to rely on time-invariant autonomous systems. GMR can also be employed in this context to retrieve an autonomous system $\mathcal{P}(\dot{\boldsymbol{x}}|\boldsymbol{x})$ from the joint distribution $\mathcal{P}(\boldsymbol{x}, \dot{\boldsymbol{x}})$ encoded in a GMM, where $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ are position and velocity, respectively (see [13] for details). Similarly, it can be used in an autoregressive context by retrieving $\mathcal{P}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \ldots, \boldsymbol{x}_{t-T})$ at each time step $t$, from the joint encoding of the positions on a time window of size $T$.

Practical applications of GMR include the analysis of speech signals [15, 32], electromyography signals [17], vision and MoCap data [30], and cancer prognosis [10]. A MATLAB code example `demo_GMR01.m` can be found in [25].

### 3.2.2 Bernstein Basis Functions

Bézier curves are well-known representations of trajectories [11]. Their underlying representation is a superposition of basis functions, which is overlooked in many applications. For $0 \leqslant t \leqslant 1$, a linear Bézier curve is the line traced by the function $\boldsymbol{x}_{p_0,p_1}(t)$, from $\boldsymbol{p}_0$ to $\boldsymbol{p}_1$,

$$\boldsymbol{x}_{p_0,p_1}(t) = (1-t)\boldsymbol{p}_0 + t\,\boldsymbol{p}_1. \tag{3.9}$$

For $0 \leqslant t \leqslant 1$, a quadratic Bézier curve is the path traced by the function

$$\begin{aligned}\boldsymbol{x}_{p_0,p_1,p_2}(t) &= (1-t)\,\boldsymbol{x}_{p_0,p_1}(t) + t\,\boldsymbol{x}_{p_1,p_2}(t)\\ &= (1-t)\Big((1-t)\boldsymbol{p}_0 + t\,\boldsymbol{p}_1\Big) + t\Big((1-t)\boldsymbol{p}_1 + t\,\boldsymbol{p}_2\Big)\\ &= (1-t)^2\boldsymbol{p}_0 + 2(1-t)t\,\boldsymbol{p}_1 + t^2\boldsymbol{p}_2.\end{aligned} \tag{3.10}$$

For $0 \leqslant t \leqslant 1$, a cubic Bézier curve is the path traced by the function

$$\begin{aligned}\boldsymbol{x}_{p_0,p_1,p_2,p_3}(t) &= (1-t)\,\boldsymbol{x}_{p_0,p_1,p_2}(t) + t\,\boldsymbol{x}_{p_1,p_2,p_3}(t)\\ &= (1-t)^3\boldsymbol{p}_0 + 3(1-t)^2t\,\boldsymbol{p}_1 + 3(1-t)t^2\boldsymbol{p}_2 + t^3\boldsymbol{p}_3.\end{aligned} \tag{3.11}$$

For $0 \leqslant t \leqslant 1$, a recursive definition for a Bézier curve of degree $n$ can be expressed as a linear interpolation of a pair of corresponding points in two Bézier curves of degree $n-1$, namely

$$\boldsymbol{x}(t) = \sum_{i=0}^{n} b_{i,n}(t)\,\boldsymbol{p}_i, \quad \text{with} \quad b_{i,n}(t) = \binom{n}{i}(1-t)^{n-i}\,t^i, \tag{3.12}$$

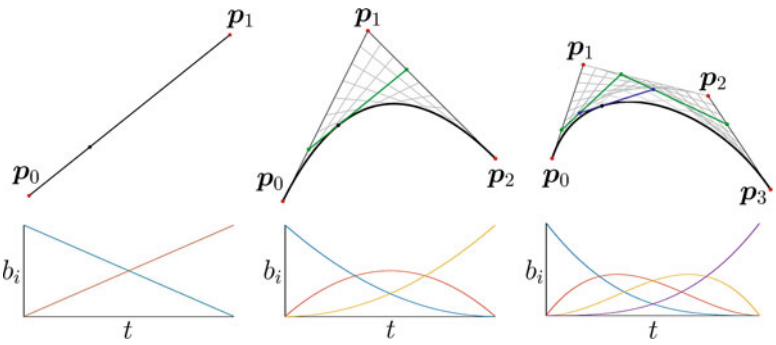



**Fig. 3.4** Linear (*left*), quadratic (*center*) and cubic (*right*) Bézier curves constructed as a weighted superposition of Bernstein basis functions

with $b_{i,n}(t)$ the Bernstein basis polynomials of degree n, where $\binom{n}{i} = \frac{n!}{i!(n-i)!}$ are binomial coefficients.

Figure 3.4 illustrates the construction of Bézier curves of different orders. Practical applications are diverse but include most notably trajectories in computer graphics [11] and path planning [9]. A MATLAB code example `demo_Bezier01.m` can be found in [25].

### 3.2.3 Fourier Basis Functions

In time series encoding, the use of Fourier basis functions provides useful connections between the spatial domain and the frequency domain. In the context of Gaussian mixture models, several Fourier series properties can be exploited, notably regarding zero-centered Gaussians, shift, symmetry, and linear combination. For the 1D case, these properties are:

- If $\phi(x) = \mathcal{N}\left(x \mid 0, \sigma^2\right) = \left(2\pi\sigma^2\right)^{-\frac{1}{2}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$ is used to create a periodic function with period $L \gg \sigma$, the corresponding Fourier series coefficients are of the form $\phi_k = \exp\left(-\frac{2\pi^2 k^2 \sigma^2}{L^2}\right)$;
- If $\phi_k$ are the Fourier series coefficients of a function $\phi(x)$, $\exp\left(-i\frac{2\pi k\mu}{L}\right)\phi_k$ are the Fourier coefficients of $\phi(x-\mu)$, with $i$ the imaginary unit ($i^2 = -1$);
- If $\phi_{k,1}$ (resp. $\phi_{k,2}$) are the Fourier series coefficients of a function $\phi_1(x)$ (resp. $\phi_2(x)$), then $\alpha_1\phi_{k,1} + \alpha_2\phi_{k,2}$ are the Fourier coefficients of $\alpha_1\phi_1(x) + \alpha_2\phi_2(x)$.

Well-known applications of Fourier basis functions in the context of time series include speech processing [15, 32] and the analysis of periodic motions such as gaits [2]. Such decompositions also have a wider scope of applications, as illustrated next with ergodic control.

### 3.2.4 Ergodic Control

In ergodic control, the aim is to find a series of control commands $\boldsymbol{u}(t)$ so that the retrieved trajectory $\boldsymbol{x}(t) \in \mathbb{R}^D$ covers a bounded space $\mathcal{X}$ in proportion of a given spatial distribution $\phi(\boldsymbol{x})$. As proposed in [21], this can be achieved by defining a metric in the spectral domain, by decomposing in Fourier series coefficients both the spatial distribution $\phi(\boldsymbol{x})$ and the (partially) retrieved trajectory $\boldsymbol{x}(t)$.[2] The goal of ergodic control is to minimize

$$\epsilon\big(\boldsymbol{x}(t)\big) = \frac{1}{2} \sum_{\boldsymbol{k} \in \mathcal{K}} \Lambda_{\boldsymbol{k}} \Big( c_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big) - \phi_{\boldsymbol{k}} \Big)^2 \tag{3.13}$$

$$= \frac{1}{2} \Big( \boldsymbol{c}\big(\boldsymbol{x}(t)\big) - \boldsymbol{\phi} \Big)^{\top} \boldsymbol{\Lambda} \Big( \boldsymbol{c}\big(\boldsymbol{x}(t)\big) - \boldsymbol{\phi} \Big), \tag{3.14}$$

where $\Lambda_{\boldsymbol{k}}$ are weights, $\phi_{\boldsymbol{k}}$ are the Fourier series coefficients of $\phi(\boldsymbol{x})$, and $c_{\boldsymbol{k}}$ are the Fourier series coefficients along the trajectory $\boldsymbol{x}(t)$. $\mathcal{K}$ is a set of index vectors in $\mathbb{N}^D$ covering the $D$-dimensional array $\boldsymbol{k} = \boldsymbol{r} \times \boldsymbol{r} \times \cdots \times \boldsymbol{r}$, with $\boldsymbol{r} = [0, 1, \ldots, K]$ and $K$ the resolution of the array. $\boldsymbol{c} \in \mathbb{R}^{K^D}$ and $\boldsymbol{\phi} \in \mathbb{R}^{K^D}$ are vectors composed of elements $c_{\boldsymbol{k}}$ and $\phi_{\boldsymbol{k}}$, respectively. $\boldsymbol{\Lambda} \in \mathbb{R}^{K^D \times K^D}$ is a diagonal weighting matrix with elements $\Lambda_{\boldsymbol{k}}$. In (3.13), the weights

$$\Lambda_{\boldsymbol{k}} = \Big( 1 + \|\boldsymbol{k}\|^2 \Big)^{-\frac{D+1}{2}} \tag{3.15}$$

assign more importance on matching low frequency components (related to a metric for Sobolev spaces of negative order). The Fourier series coefficients $c_{\boldsymbol{k}}$ along a trajectory $\boldsymbol{x}(t)$ of duration $t$ are defined as

$$c_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big) = \frac{1}{t} \int_{s=0}^{t} f_{\boldsymbol{k}}\big(\boldsymbol{x}(s)\big)\, ds, \tag{3.16}$$

whose discretized version can be computed recursively at each time step $t$ to build

$$c_{\boldsymbol{k}}(\boldsymbol{x}_t) = \frac{1}{t} \sum_{s=1}^{t} f_{\boldsymbol{k}}(\boldsymbol{x}_s), \tag{3.17}$$

or equivalently in vector form $\boldsymbol{c}(\boldsymbol{x}_t) = \frac{1}{t} \sum_{s=1}^{t} \boldsymbol{f}(\boldsymbol{x}_s)$.

[2] In [21], cosine basis functions are employed but the approach can be extended to other basis functions.

For a spatial signal $\boldsymbol{x} \in \mathbb{R}^D$, where $x_d$ is on the interval $[-\frac{L}{2}, \frac{L}{2}]$ of period $L$, $\forall d \in \{1, \dots, D\}$, the basis functions of the Fourier series with complex exponential functions are defined as

$$f_{\boldsymbol{k}}(\boldsymbol{x}) = \frac{1}{L^D} \prod_{d=1}^{D} \exp\left(-i \frac{2\pi k_d x_d}{L}\right)$$
$$= \frac{1}{L^D} \prod_{d=1}^{D} \cos\left(\frac{2\pi k_d x_d}{L}\right) - i \sin\left(\frac{2\pi k_d x_d}{L}\right), \quad \forall \boldsymbol{k} \in \mathcal{K}. \tag{3.18}$$

#### 3.2.4.1 Computation of Fourier Series Coefficients $\phi_{\boldsymbol{k}}$ for a Spatial Distribution Represented as a Gaussian Mixture Model

We consider a desired spatial distribution $\phi_0(\boldsymbol{x})$ represented as a mixture of $J$ Gaussians with centers $\boldsymbol{\mu}_j$, covariance matrices $\boldsymbol{\Sigma}_j$, and mixing coefficients $\alpha_j$ (with $\sum_{j=1}^{J} \alpha_j = 1$ and $\alpha_j \geqslant 0$),

$$\phi_0(\boldsymbol{x}) = \sum_{j=1}^{J} \alpha_j \, \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big) \tag{3.19}$$
$$= \sum_{j=1}^{J} \alpha_j \, (2\pi)^{-\frac{D}{2}} \, |\boldsymbol{\Sigma}_j|^{-\frac{1}{2}} \, \exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_j)\Big),$$

with each dimension on the interval $\left[0, \frac{L}{2}\right]$. $\phi_0(\boldsymbol{x})$ is extended to a periodized function by constructing an even function on the interval $\mathcal{X}$, where each dimension $x_d$ is on the interval $\mathcal{X} = \left[-\frac{L}{2}, \frac{L}{2}\right]$ of period $L$. This is achieved with mirror symmetries of the Gaussians around all zero axes, see Fig. 3.5d. The resulting spatial distribution can be expressed as a mixture of $2^D J$ Gaussians

$$\phi(\boldsymbol{x}) = \sum_{j=1}^{J} \sum_{m=1}^{2^D} \frac{\alpha_j}{2^D} \, \mathcal{N}\big(\boldsymbol{x} \,\big|\, \boldsymbol{A}_m \boldsymbol{\mu}_j, \boldsymbol{A}_m \boldsymbol{\Sigma}_j \boldsymbol{A}_m^\top\big), \tag{3.20}$$

with linear transformation matrices $\boldsymbol{A}_m$.[3] By exploiting the symmetries and Gaussian distribution properties presented in Sect. 3.2.3, the Fourier series coefficients $\phi_{\boldsymbol{k}}$ can be analytically computed as

[3] $\boldsymbol{A}_m = \text{diag}(\boldsymbol{H}_{2^D-D+1:2^D,m})$, where $\boldsymbol{H}_{2^D-D+1:2^D,m}$ is a vector composed of the last $D$ elements in the column $m$ of the Hadamard matrix $\boldsymbol{H}$ of size $2^D$. Alternatively, $\boldsymbol{A}_m = \text{diag}\big(\text{vec}(\boldsymbol{\ell}_m)\big)$ can be constructed with the array $\boldsymbol{\ell}_m$, with $m$ indexing the first dimension of the array $\boldsymbol{\ell} = \boldsymbol{s} \times \boldsymbol{s} \times \cdots \times \boldsymbol{s} \in \mathbb{Z}^{2\times 2\times \cdots \times 2}$ with $\boldsymbol{s} = [-1, 1]$. In 2D, we have $\boldsymbol{A}_1 = \left[\begin{smallmatrix} -1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$, $\boldsymbol{A}_2 = \left[\begin{smallmatrix} -1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $\boldsymbol{A}_3 = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$ and $\boldsymbol{A}_2 = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, see Fig. 3.5d.

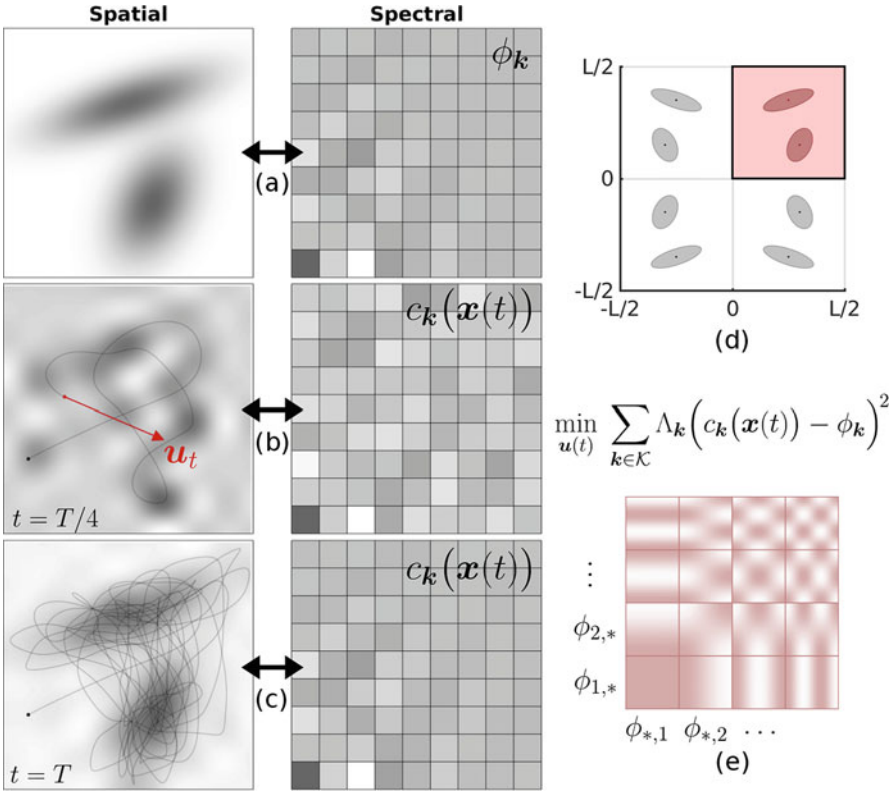



**Fig. 3.5** 2D ergodic control problem. In (**a**)–(**c**), the left graphs show the spatial distribution (gray colormap) that the agent has to explore, encoded here as a mixture of two Gaussians. The right graphs show the corresponding Fourier series coefficients $\phi_{\boldsymbol{k}}$ in the frequency domain ($K = 9$ coefficients per dimension), which can be computed analytically by exploiting the shift, symmetry and linear combination properties of Gaussians. (**b**) Shows the evolution of the reconstructed spatial distribution (left graph) and the computation of the next control command $\boldsymbol{u}$ (red arrow) after one fourth of the movement. The corresponding Fourier series coefficients $c_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big)$ are shown in the right graph. (**c**) Shows that after $T$ iterations, the agent covers the space in proportion to the desired spatial distribution, with a good match of coefficients in the frequency domain ($\phi_{\boldsymbol{k}}$ in (**a**) and $c_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big)$ in (**c**) are nearly the same). (**d**) Shows how a periodic signal $\phi(\boldsymbol{x})$ (with range $[-L/2, L/2]$ for each dimension) can be constructed from the original mixture of two Gaussians $\phi_0(\boldsymbol{x})$ (red area). The constructed signal $\phi(\boldsymbol{x})$ is composed of eight Gaussians in this 2D example (mirroring the Gaussians along horizontal and vertical axes to construct an even signal of period $L$). (**e**) Depicts the spatial reconstruction of each Fourier series coefficient (for the first four coefficients in each dimension), corresponding to periodic signals at different frequencies along the two axes

$$\begin{aligned}\phi_{\boldsymbol{k}} &= \int_{\boldsymbol{x}\in\mathcal{X}} \phi(\boldsymbol{x})\ f_{\boldsymbol{k}}(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x} \\ &= \frac{1}{L^D}\sum_{j=1}^{J}\sum_{m=1}^{2^D}\frac{\alpha_j}{2^D}\exp\left(-i\frac{2\pi\boldsymbol{k}^\top\boldsymbol{A}_m\boldsymbol{\mu}_j}{L}\right)\exp\left(-\frac{2\pi^2\boldsymbol{k}^\top\boldsymbol{A}_m\boldsymbol{\Sigma}_j\boldsymbol{A}_m^\top\boldsymbol{k}}{L^2}\right) \\ &= \frac{1}{L^D}\sum_{j=1}^{J}\sum_{m=1}^{2^{D-1}}\frac{\alpha_j}{2^{D-1}}\cos\left(\frac{2\pi\boldsymbol{k}^\top\boldsymbol{A}_m\boldsymbol{\mu}_j}{L}\right)\exp\left(-\frac{2\pi^2\boldsymbol{k}^\top\boldsymbol{A}_m\boldsymbol{\Sigma}_j\boldsymbol{A}_m^\top\boldsymbol{k}}{L^2}\right).\end{aligned} \tag{3.21}$$

With this mirroring, we can see that $\phi_{\boldsymbol{k}}$ are real and even, where an evaluation over $\boldsymbol{k}\in\mathcal{K}$, $j\in\{1,2,\ldots,J\}$ and $m\in\{1,2,\ldots,2^{D-1}\}$ in (3.21) is sufficient to fully characterize the signal.

#### 3.2.4.2 Controller for a Spatial Distribution Represented as a Gaussian Mixture Model

In [21], ergodic control is set as the constrained problem of computing a control command $\hat{\boldsymbol{u}}(t)$ at each time step $t$ with

$$\hat{\boldsymbol{u}}(t) = \arg\min_{\boldsymbol{u}(t)}\ \epsilon\big(\boldsymbol{x}(t)+\Delta t\big), \quad \text{s.t.} \quad \dot{\boldsymbol{x}}(t) = f\big(\boldsymbol{x}(t),\boldsymbol{u}(t)\big), \quad \|\boldsymbol{u}(t)\| \leqslant u^{\max}, \tag{3.22}$$

where the simple system $\dot{\boldsymbol{x}}(t) = \boldsymbol{u}(t)$ is considered (control with velocity commands), and where the error term is approximated with the Taylor series

$$\epsilon\big(\boldsymbol{x}(t)+\Delta t\big) \approx \epsilon\big(\boldsymbol{x}(t)\big) + \dot{\epsilon}\big(\boldsymbol{x}(t)\big)\Delta t + \frac{1}{2}\ddot{\epsilon}\big(\boldsymbol{x}(t)\big)\Delta t^2. \tag{3.23}$$

By using (3.13), (3.16), (3.18) and the chain rule $\frac{\partial f}{\partial t} = \frac{\partial f}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial t}$, the Taylor series is composed of the control term $\boldsymbol{u}(t)$ and $\nabla_{\boldsymbol{x}} f_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big)\in\mathbb{R}^{1\times D}$, the gradient of $f_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big)$ with respect to $\boldsymbol{x}(t)$. Solving the constrained objective in (3.22) then results in the analytical solution (see [21] for the complete derivation)

$$\begin{aligned}\boldsymbol{u} = \tilde{\boldsymbol{u}}(t)\frac{u^{\max}}{\|\tilde{\boldsymbol{u}}(t)\|}, \quad \text{with} \quad \tilde{\boldsymbol{u}} &= -\sum_{\boldsymbol{k}\in\mathcal{K}}\Lambda_{\boldsymbol{k}}\Big(c_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big)-\phi_{\boldsymbol{k}}\Big)\nabla_{\boldsymbol{x}} f_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big)^\top \\ &= -\nabla_{\boldsymbol{x}}\boldsymbol{f}\big(\boldsymbol{x}(t)\big)\,\boldsymbol{\Lambda}\Big(\boldsymbol{c}\big(\boldsymbol{x}(t)\big)-\boldsymbol{\phi}\Big),\end{aligned} \tag{3.24}$$

where $\nabla_{\boldsymbol{x}}\boldsymbol{f}\big(\boldsymbol{x}(t)\big)\in\mathbb{R}^{D\times K^D}$ is a concatenation of the vectors $\nabla_{\boldsymbol{x}} f_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big)$. Figure 3.5 shows a 2D example of ergodic control to create a motion approximating the distribution given by a mixture of two Gaussians. A remarkable characteristic of

such approach is that the controller produces natural exploration behaviors (see Fig. 3.5c) without relying on stochastic noise in the formulation. In the limit case, if the distribution $\phi(\boldsymbol{x})$ is a single Gaussian with a very small isotropic covariance, the controller results in a standard tracking behavior.

Examples of application include surveillance with multi-agent systems [21], active shape estimation [1], and localization for fish-like robots [22]. A MATLAB code example `demo_ergodicControl01.m` can be found in [25].

## 3.3 Probabilistic Movement Primitives

The representation of time series as a superposition of basis functions can also be exploited to construct trajectory distributions. Representing a collection of trajectories in the form of a multivariate distribution has several advantages. First, new trajectories can be stochastically generated. Then, the conditional probability property (see (3.7)) can be exploited to generate trajectories passing through via-points (including starting and/or ending points). This is simply achieved by specifying as inputs $\boldsymbol{x}^I$ in (3.7) the datapoints that the system needs to pass through (with corresponding dimensions in the hyperdimensional vector) and by retrieving as output $\boldsymbol{x}^O$ the remaining parts of the trajectory.

A naive approach to represent a collection of $M$ trajectories in a probabilistic form is to reorganize each trajectory as a hyperdimensional datapoint $\boldsymbol{x}_m = [\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, \dots, \boldsymbol{x}_T^\top]^\top \in \mathbb{R}^{DT}$, and fitting a Gaussian $\mathcal{N}(\boldsymbol{\mu}^{\boldsymbol{x}}, \boldsymbol{\Sigma}^{\boldsymbol{x}})$ to these datapoints, see Fig. 3.6, *left*. Since the dimension $DT$ might be much larger than the number of datapoints $M$, a potential solution to this issue could be to consider an eigendecomposition of the covariance (ordered by decreasing eigenvalues)

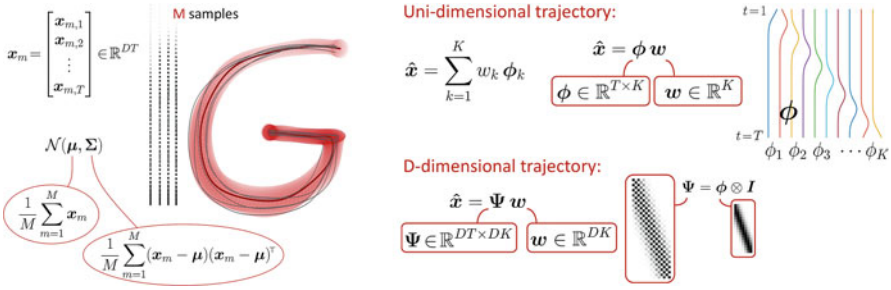



**Fig. 3.6** *Left:* Raw trajectory distribution as a Gaussian of size $DT$ by organizing each of the $M$ samples as a trajectory vector, where each trajectory has $T$ time steps and each point has $D$ dimensions ($T = 100$ and $D = 2$ in this example). *Right:* Trajectory distribution encoded with probabilistic movement primitives (superposition of $K$ basis functions). The right part of the figure depicts the linear mapping functions $\boldsymbol{\phi}$ and $\boldsymbol{\Psi}$ created by a decomposition with radial basis functions

$$\boldsymbol{\Sigma}^{\boldsymbol{x}} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^{\top} = \sum_{j=1}^{DT} \lambda_j \boldsymbol{v}_j \boldsymbol{v}_j^{\top}, \tag{3.25}$$

with $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{DT}]$ and $\boldsymbol{D} = \text{diag}(\lambda_1^2, \lambda_2^2, \ldots, \lambda_{DT}^2)$. This can be exploited to project the data in a subspace of reduced dimensionality through principal component analysis. By keeping the first $KT$ components, such approach provides a Gaussian distribution of the trajectories with the structure $\mathcal{N}(\boldsymbol{\Psi}\boldsymbol{\mu}^{\boldsymbol{w}}, \boldsymbol{\Psi}\boldsymbol{\Psi}^{\top})$, where $\boldsymbol{\Psi} = [\boldsymbol{v}_1\lambda_1, \boldsymbol{v}_2\lambda_2, \ldots, \boldsymbol{v}_{DK}\lambda_{DK}]$.

The ProMP (probabilistic movement primitive) model proposed in [24] also encodes the trajectory distribution in a subspace of reduced dimensionality, but provides a RBF structure to this decomposition instead of the eigendecomposition as in the above. It assumes that each sample trajectory $m \in \{1, \ldots, M\}$ can be approximated by a weighted sum of $K$ normalized RBFs with

$$\boldsymbol{x}_m = \boldsymbol{\Psi}\boldsymbol{w}_m + \boldsymbol{\epsilon}, \quad \text{where} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \lambda \boldsymbol{I}), \tag{3.26}$$

and basis functions organized as

$$\boldsymbol{\Psi} = \boldsymbol{\phi} \otimes \boldsymbol{I} = \begin{bmatrix} \boldsymbol{I}\phi_1(t_1) & \boldsymbol{I}\phi_2(t_1) & \cdots & \boldsymbol{I}\phi_K(t_1) \\ \boldsymbol{I}\phi_1(t_2) & \boldsymbol{I}\phi_2(t_2) & \cdots & \boldsymbol{I}\phi_K(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{I}\phi_1(t_T) & \boldsymbol{I}\phi_2(t_T) & \cdots & \boldsymbol{I}\phi_K(t_T) \end{bmatrix}, \tag{3.27}$$

with $\boldsymbol{\Psi} \in \mathbb{R}^{DT \times DK}$, identity matrix $\boldsymbol{I} \in \mathbb{R}^{D \times D}$, and $\otimes$ the Kronecker product. A vector $\boldsymbol{w}_m \in \mathbb{R}^{DK}$ can be estimated for each of the $M$ sample trajectories by the least squares estimate

$$\boldsymbol{w}_m = (\boldsymbol{\Psi}^{\top}\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^{\top}\boldsymbol{x}_m. \tag{3.28}$$

By assuming that $\{\boldsymbol{w}_m\}_{m=1}^{M}$ can be represented with a Gaussian $\mathcal{N}(\boldsymbol{\mu}^{\boldsymbol{w}}, \boldsymbol{\Sigma}^{\boldsymbol{w}})$ characterized by a center $\boldsymbol{\mu}^{\boldsymbol{w}} \in \mathbb{R}^{DK}$ and a covariance $\boldsymbol{\Sigma}^{\boldsymbol{w}} \in \mathbb{R}^{DK \times DK}$, a trajectory distribution $\mathcal{P}(\boldsymbol{x})$ can then be computed as

$$\boldsymbol{x} \sim \mathcal{N}\Big(\boldsymbol{\Psi}\boldsymbol{\mu}^{\boldsymbol{w}}, \boldsymbol{\Psi}\boldsymbol{\Sigma}^{\boldsymbol{w}}\boldsymbol{\Psi}^{\top} + \sigma^2\boldsymbol{I}\Big), \tag{3.29}$$

with $\boldsymbol{x} \in \mathbb{R}^{DT}$ a trajectory of $T$ datapoints of $D$ dimensions organized in a vector form and $\boldsymbol{I} \in \mathbb{R}^{DT \times DT}$, see Figs. 3.6 and 3.7.

The parameters of the ProMP model are $\sigma^2$, $\mu_k^I$, $\Sigma_k^I$, $\boldsymbol{\mu}^{\boldsymbol{w}}$, and $\boldsymbol{\Sigma}^{\boldsymbol{w}}$. A Gaussian of $DK$ dimensions is estimated, providing a compact representation of the movement, separating the temporal components $\boldsymbol{\Psi}$ and spatial components $\mathcal{N}(\boldsymbol{\mu}^{\boldsymbol{w}}, \boldsymbol{\Sigma}^{\boldsymbol{w}})$. Similarly to LWR, ProMP can be coupled with GMM/GMR to automatically estimate the location and bandwidth of the basis functions as a joint distribution problem, instead of specifying them manually. A mixture of ProMPs can be efficiently estimated by fitting a GMM to the datapoints $\boldsymbol{w}_m$, and using the linear transformation property of

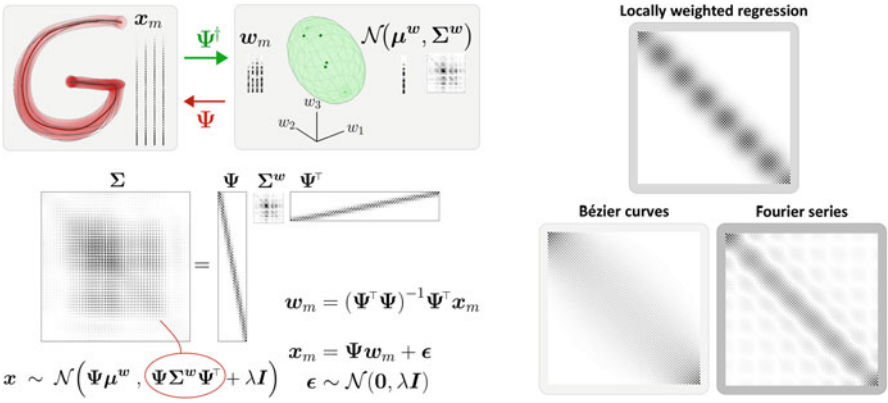



**Fig. 3.7** *Left:* Illustration of probabilistic movement primitives as a linear mapping between the original space of trajectories and a subspace of reduced dimensionality. After projecting each trajectory sample in this subspace (with linear map $\boldsymbol{\Psi}^\dagger$ computed as the pseudoinverse of $\boldsymbol{\Psi}$), a Gaussian is evaluated, which is then projected back to the original trajectory space by exploiting the linear transformation property of multivariate Gaussians (with linear map $\boldsymbol{\Psi}$). Such decomposition results in a low rank structure of the covariance matrix, which is depicted in the bottom part of the figure. *Right:* Representation of the covariance matrix $\boldsymbol{\Psi}\boldsymbol{\Psi}^\top$ for various basis functions, all showing some form of sparsity

Gaussians to convert this mixture into a mixture at the trajectory level. Moreover, such representation can be extended to other basis functions, including Bernstein and Fourier basis functions, see Fig. 3.7, *right*.

ProMP has been demonstrated in various robotic tasks requiring human-like motion capabilities such as playing the maracas and using a hockey stick [24], or for collaborative object handover and assistance in box assembly [20]. A MATLAB code example `demo_proMP01.m` can be found in [25].

## 3.4 Further Challenges and Conclusion

This chapter presented various forms of superposition for time signals analysis and synthesis, by emphasizing the connections to Gaussian mixture models. The connections between these decomposition techniques are often underexploited, mainly due to the fact that these techniques were developed separately in various fields of research. The framework of mixture models provides a unified view that is inspirational to make links between these models. Such links also stimulate future developments and extensions.

Future challenges include a better exploitation of the joint roles that mixture of experts (MoE) and product of experts (PoE) can offer in the treatment of time series and control policies [26]. While MoE can decompose a complex signal by superposing a set of simpler signals, PoE can fuse information by considering more

elaborated forms of superposition (with full precision matrices instead of scalar weights). Often, either one or the other approach is considered in practice, but many applications would leverage the joint use of these two techniques.

There are also many further challenges specific to each basis function categories presented in this chapter. For Gaussian mixture regression (GMR), a relevant extension is to include a Bayesian perspective to the approach. This can take the form of a model selection problem, such as an automatic estimation of the number of Gaussians and rank of the covariance matrices [29]. This can also take the form of a more general Bayesian modeling perspective by considering the variations of the mixture model parameters (including means and covariances) [26]. Such extension brings new perspectives to GMR, by providing a representation that allows uncertainty quantification and multimodal conditional estimates to be considered. Other techniques like Gaussian processes also provide uncertainty quantification, but they are typically much slower. A Bayesian treatment of mixture model conditioning offers new perspectives for an efficient and robust treatment of wide-ranging data. Namely, models that can be trained with only few datapoints but that are rich enough to scale when more training data are available.

Another important challenge in GMR is to extend the techniques to more diverse forms of data. Such regression problem can be investigated from a geometrical perspective (e.g., by considering data lying on Riemannian manifolds [17]) or from a topological perspective (e.g., by considering relative distance space representations [16]). It can also be investigated from a structural perspective by exploiting tensor methods [19]. When data are organized in matrices or arrays of higher dimensions (tensors), classical regression methods first transform these data into vectors, therefore ignoring the underlying structure of the data and increasing the dimensionality of the problem. This flattening operation typically leads to overfitting when only few training data are available. Tensor representations instead exploit the intrinsic structure of multidimensional arrays. Mixtures of experts can be extended to tensorial representations for regression of tensor-valued data [18], which could potentially be employed to extend GMR representations to arrays of higher dimensions.

Regarding Bézier curves, even if the technique is well established, there is still room for further perspectives, in particular with the links to other techniques that such approach has to offer. For example, Bézier curves can be reframed as a model predictive control (MPC) problem [5, 9], a widespread optimal control technique used to generate movements with the capability of anticipating future events. Formulating Bézier curves as a superposition of Bernstein polynomials also leaves space for probabilistic interpretations, including Bayesian treatments.

The consideration of Fourier series for the superposition of basis functions might be the approach with the widest range of possible developments. Indeed, the representation of continuous time signals in the frequency domain is omnipresent in many fields of research, and, as exemplified with ergodic control, there are many opportunities to exploit the Gaussian properties in mixture models by taking into account their dual representation in spatial and frequency domains.

With the specific application of ergodic control, the dimensionality issue requires further consideration. In the basic formulation, by keeping $K$ basis functions to encode time series composed of datapoints of dimension $D$, $K^D$ Fourier series components are required. Such formulation has the advantage of taking into account all possible correlations across dimensions, but it slows down the process when $D$ is large. A potential direction to cope with such scaling issue would be to rely on Gaussian mixture models (GMMs) with low-rank structures on the covariances [29], such as in mixtures of factor analyzers (MFA) or mixtures of probabilistic principal component analyzers (MPPCA) [6]. Such subspaces of reduced dimensionality could potentially be exploited to reduce the number of Fourier basis coefficients to be computed.

Finally, the probabilistic representation of movements primitives in the form of trajectory distributions also offers a wide range of new perspectives. Such models classically employ radial basis functions, but can be extended to a richer family of basis functions (including a combination of those). This was exemplified in the chapter with the use of Bernstein and Fourier bases to build probabilistic movement primitives, see Fig. 3.7, *right*. More generally, links to kernel methods can be created by extension of this representation [14]. Other extensions include the use of mixture models and associated Bayesian methods to encode the weights $\boldsymbol{w}_m$ in the subspace of reduced dimensionality.

**Acknowledgements** I would like to thank Prof. Michael Liebling for his help in the development of the ergodic control formulation applied to Gaussian mixture models and for his recommendations on the preliminary version of this chapter.

The research leading to these results has received funding from the European Commission's Horizon 2020 Programme (H2020/2018-20) under the MEMMO Project (Memory of Motion, http://www.memmo-project.eu/), grant agreement 780684.

## References

1. Abraham, I., Prabhakar, A., Hartmann, M.J., Murphey, T.D.: Ergodic exploration using binary sensing for nonparametric shape estimation. IEEE Robot. Autom. Lett. **2**(2), 827–834 (2017)
2. Antonsson, E. K., Mann, R.W.: The frequency content of gait. J. Biomech. **18**(1), 39–47 (1985)
3. Atkeson, C. G.: Using local models to control movement. In: Advances in Neural Information Processing Systems (NIPS), vol. 2, pp 316–323 (1989)
4. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally weighted learning for control. Artif. Intell. Rev. **11**(1–5), 75–113 (1997)
5. Berio, D., Calinon, S., Leymarie, F.F.: Generating calligraphic trajectories with model predictive control. In: Proceedings of the 43rd Conference on Graphics Interface, pp 132–139. Canadian Human-Computer Communications Society School of Computer Science, University of Waterloo, Waterloo (2017)
6. Bouveyron, C., Brunet, C.: Model-based clustering of high-dimensional data: A review. Comput. Stat. Data Anal. **71**, 52–78 (2014)
7. Calinon, S., Lee, D.: Learning control. In: Vadakkepat, P., Goswami, A. (eds.) Humanoid Robotics: A Reference, pp. 1261–1312. Springer, Berlin (2019)

8. Cleveland, W.S.: Robust locally weighted regression and smoothing scatterplots. Am. Stat. Assoc. **74**(368), 829–836 (1979)
9. Egerstedt, M., Martin, C.: Control Theoretic Splines: Optimal Control, Statistics, and Path Planning. Princeton University Press, Princeton (2010)
10. Falk, T.H., Shatkay, H., Chan, W.Y.: Breast cancer prognosis via Gaussian mixture regression. In: Conference on Electrical and Computer Engineering, pp. 987–990. IEEE, Piscataway (2006)
11. Farouki, R.T.: The Bernstein polynomial basis: A centennial retrospective. Comput. Aided Geom. Des. **29**(6), 379–419 (2012)
12. Ghahramani, Z., Jordan, M.I.: Supervised learning from incomplete data via an EM approach. In: Cowan, J.D., Tesauro, G., Alspector, J. (eds.) Advances in Neural Information Processing Systems (NIPS), vol 6, pp 120–127. Morgan Kaufmann Publishers, San Francisco (1994)
13. Hersch, M., Guenter, F., Calinon, S., Billard, A.: Dynamical system modulation for robot learning via kinesthetic demonstrations. IEEE Trans. Robot. **24**(6), 1463–1467 (2008)
14. Huang, Y., Rozo, L., Silvério, J., Caldwell, D.G.: Kernelized movement primitives. Int. J. Robot. Res. **38**(7), 833–852 (2019)
15. Hueber, T., Bailly, G.: Statistical conversion of silent articulation into audible speech using full-covariance HMM. Comput. Speech Lang. **36**(C), 274–293 (2016)
16. Ivan, V., Zarubin, D., Toussaint, M., Komura, T., Vijayakumar, S.: Topology-based representations for motion planning and generalization in dynamic environments with interactions. Int. J. Robot. Res. **32**(9–10), 1151–1163 (2013)
17. Jaquier, N., Calinon, S.: Gaussian mixture regression on symmetric positive definite matrices manifolds: Application to wrist motion estimation with sEMG. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 59–64. IEEE, Piscataway (2017)
18. Jaquier, N., Haschke, R., Calinon, S.: Tensor-variate mixture of experts. arXiv:190211104 pp 1–11 (2019)
19. Kolda, T., Bader, B.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
20. Maeda, G.J., Neumann, G., Ewerton, M., Lioutikov, R., Kroemer, O., Peters, J.: Probabilistic movement primitives for coordination of multiple human-robot collaborative tasks. Auton. Robot. **41**(3), 593–612 (2017)
21. Mathew, G., Mezic, I.: Metrics for ergodicity and design of ergodic dynamics for multi-agent systems. Phys. D Nonlinear Phenom. **240**(4), 432–442 (2011)
22. Miller, L.M., Silverman, Y., MacIver, M.A., Murphey, T.D.: Ergodic exploration of distributed information. IEEE Trans. Robot. **32**(1), 36–52 (2016)
23. Mussa-Ivaldi, F.A., Giszter, S.F., Bizzi, E.: Linear combinations of primitives in vertebrate motor control. Proc Natl. Acad. Sci. **91**, 7534–7538 (1994)
24. Paraschos, A., Daniel, C., Peters, J.R., Neumann, G.: Probabilistic movement primitives. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems (NIPS), pp 2616–2624. Curran Associates, Red Hook (2013)
25. PbDlib robot programming by demonstration software library. http://www.idiap.ch/software/pbdlib/. Accessed 18 April 2019
26. Pignat, E., Calinon, S.: Bayesian Gaussian mixture model for robotic policy imitation. arXiv:190410716, pp. 1–7 (2019)
27. Schaal, S., Atkeson, C.G.: Constructive incremental learning from only local information. Neural Comput. **10**(8), 2047–2084 (1998)
28. Stulp, F., Sigaud, O.: Many regression algorithms, one unified model—a review. Neural Netw. **69**, 60–79 (2015)
29. Tanwani, A.K., Calinon, S.: Small variance asymptotics for non-parametric online robot learning. Int. J. Rob. Res. **38**(1), 3–22 (2019)
30. Tian, Y., Sigal, L., De la Torre, F., Jia, Y.: Canonical locality preserving latent variable model for discriminative pose inference. Image Vis. Comput. **31**(3), 223–230 (2013)

31. Ting, J.A., Kalakrishnan, M., Vijayakumar, S., Schaal, S.: Bayesian kernel shaping for learning control. In: Advances in Neural Information Processing Systems (NIPS), pp 1673–1680 (2008)
32. Toda, T., Black, A.W., Tokuda, K.: Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. IEEE Trans. Audio Speech Lang. Process. **15**(8), 2222–2235 (2007)
33. Vijayakumar, S., D'souza, A., Schaal, S.: Incremental online learning in high dimensions. Neural Comput. **17**(12), 2602–2634 (2005)

# Part II
# Generalized Gaussian-Based Models

# Chapter 4
# Multivariate Bounded Asymmetric Gaussian Mixture Model


**Muhammad Azam, Basim Alghabashi, and Nizar Bouguila**

**Abstract** In this chapter, 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 and 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 abovementioned 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.

## 4.1 Introduction

Modeling data based on probability distributions has become an important research field in recent years. The use of data has increased over the years and representing them in an efficient way is of prime importance. Hence, we represent the data in terms of probability distributions and use it for various applications. This helps us learn the rudimental patterns within the data and can be used in various

M. Azam (✉)
Department of Electrical and Computer Engineering (ECE), Concordia University, Montreal, QC, Canada
e-mail: mu_azam@encs.concordia.ca

B. Alghabashi · N. Bouguila
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: b_algh@encs.concordia.ca; nizar.bouguila@concordia.ca



N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6_4

pattern recognition tasks such as multimedia categorization, storage, retrieval, etc. [1, 32, 38]. Clustering techniques help a great deal for these type of tasks as the learning is unsupervised. The idea of clustering is to learn the inherent characteristic of specific categories within the data and represent them based on this characteristic. When it comes to clustering, the role of mixture models is inevitable. Various applications nowadays use mixture models as the core method for clustering tasks [4, 5]. Mixture models consider the data to be a combination of multiple components drawn from a probability distribution. The data is modeled based on this assumption. Given a set of data, there exist a certain number of components that describe the properties of the data. Due to this reason using finite mixture models assuming the data has finite number of clusters is a good option. An important part of mixture model design is the choice of distribution for modeling the data. The use of Gaussian distribution for this purpose has been predominant in the industry in recent years. Furthermore, Gaussian mixture models (GMM) have been used in a lot of industrial applications like speech recognition, multimedia categorization, industrial automation, fault tolerant systems, etc. [39, 40, 59, 60].

In the case of GMMs, the 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 [11, 12, 27, 56]. Asymmetric Gaussian distribution has two standard deviation parameters on the left and right side of distribution, which make it possible to model asymmetric data [11].

All abovementioned distributions are unbounded having a support range of $(-\infty, +\infty)$. In many real applications, the observed data always fall in bounded support regions [15, 22, 34, 48]; hence, it is more appropriate to model the data with bounded support distribution. Based on the fact that data in many real applications fall in bounded support, the idea of bounded support mixture models was presented in [22, 34]. Motivated by observations in [34], 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 and 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

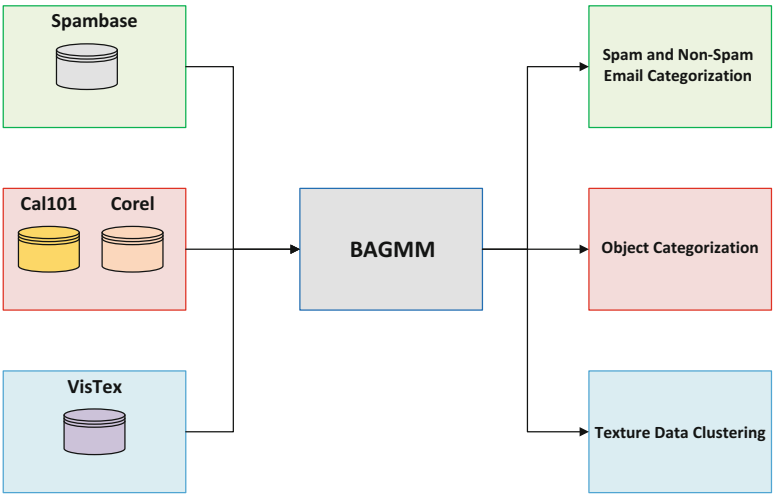



**Fig. 4.1** Graphical abstract

proposed algorithm is examined via performance measures and a comparison with AGMM. In Fig. 4.1, graphical abstract is presented which also provides more clear understanding of the contributions of this research work.

The rest of the paper is organized as follows: Sect. 4.2 describes the proposed BAGMM in detail. Section 4.3 is devoted to spam and non-spam email categorization. Section 4.4 presents the object categorization performed by our algorithm. In Sect. 4.5, texture image clustering is provided and Sect. 4.6 is dedicated to conclusion.

## 4.2 Bounded Asymmetric Gaussian Mixture 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 our model, we introduce finite mixture model with EM, asymmetric Gaussian mixture model, and bounded support mixtures.

### *4.2.1 Finite Mixture Model and EM Algorithm*

Finite mixture models are formed by taking a linear combination of distributions which are called components of mixture model. If complete likelihood of data $\mathcal{X}$ is represented as $p(\mathcal{X}|\Theta)$, where $\Theta$ is complete set of parameters of mixture model,

then parameters of mixture model are estimated via maximum likelihood (ML) estimate as follows:

$$\hat{\Theta}_{\mathrm{ML}} = \arg\max_{\Theta}\{p(\mathcal{X}|\Theta)\} \tag{4.1}$$

The details of complete likelihood of data in mixture model and its parameters are provided in the upcoming subsections [11, 17, 42, 44, 52]. In order to compute the parameters of mixture model, ML estimate cannot be found analytically and usual option for optimizing the parameter estimation is EM algorithm [17, 43, 44, 57], which is an iterative approach to determine the local maxima of likelihood. In EM algorithm, parameter estimation is performed in two steps, namely expectation (E-step) and maximization (M-step). In E-step, conditional expectation of complete likelihood is computed and in M-Step, new values of all the parameters of mixture model are estimated [17, 43, 47].

### 4.2.2 *Mixture of Asymmetric Gaussian Distributions*

Asymmetric Gaussian mixture model was proposed to handle the asymmetric properties present in different kind of data [11, 12, 31]. For a univariate data, if one data sample is represented by $X$, then asymmetric Gaussian distribution is represented as follows:

$$f(X|\mu,\sigma_l,\sigma_r) = \frac{2}{\sqrt{2\pi}(\sigma_l+\sigma_r)} \times \begin{cases} \exp\left[-\frac{(\mathrm{X}-\mu)^2}{2\sigma_l^2}\right] & \text{if} \quad \mathrm{X} < \mu \\ \exp\left[-\frac{(\mathrm{X}-\mu)^2}{2\sigma_r^2}\right] & \text{if} \quad \mathrm{X} \geq \mu \end{cases} \tag{4.2}$$

where parameters of distribution $\mu$, $\sigma_l$, and $\sigma_r$ 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 [11, 12, 31]. In Fig. 4.2, graphical representation of AGMM is displayed, where $X_i$ is a data point

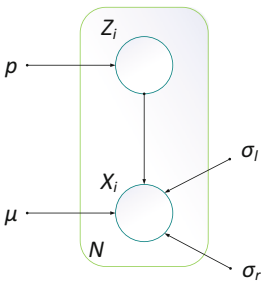



**Fig. 4.2** Graphical representation of an asymmetric Gaussian mixture model

with $i = 1, \ldots, N$, $\mu$, $\sigma_l$, and $\sigma_r$, 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 Sect. 4.2.3.

### 4.2.3 Mixture of Bounded Asymmetric Gaussian Distribution for Multidimensional Data

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

$$p(\mathbf{X}|\Theta) = \sum_{j=1}^{K} p(\mathbf{X}|\xi_j)p_j \tag{4.3}$$

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

$$p(\mathbf{X}|\xi_j) = \frac{f(\mathbf{X}|\xi_j)\mathrm{H}(\mathbf{X}|\Omega_j)}{\int_{\partial_j} f(\mathrm{u}|\xi_j)d\mathrm{u}} \tag{4.4}$$

$$\text{where } \mathrm{H}(\mathbf{X}|\Omega_j) = \begin{cases} 1 & \text{if } \mathbf{X} \in \partial_j \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

$$f(\mathbf{X}|\xi_j) = \prod_{d=1}^{D} \frac{2}{\sqrt{2\pi}(\sigma_{l_{jd}} + \sigma_{rjd})} \times \begin{cases} \exp\left[-\frac{(\mathrm{X}_d - \mu_{jd})^2}{2\sigma_{l_{jd}}^2}\right] & \text{if } \mathrm{X}_d < \mu_{jd} \\ \exp\left[-\frac{(\mathrm{X}_d - \mu_{jd})^2}{2\sigma_{r_{jd}}^2}\right] & \text{if } \mathrm{X}_d \geq \mu_{jd} \end{cases} \tag{4.6}$$

where $\boldsymbol{\mu}_j = (\mu_{j1}, \ldots, \mu_{jD})$, $\boldsymbol{\sigma}_{\mathbf{l_j}} = (\sigma_{l_{j1}}, \ldots, \sigma_{l_{jD}})$, and $\boldsymbol{\sigma}_{\mathbf{r_j}} = (\sigma_{r_{j1}}, \ldots, \sigma_{r_{jD}})$ are the mean, left standard deviation, and right standard deviation of the $D$-dimensional BAGD, respectively. The term $\int_{\partial_j} f(\mathrm{u}|\xi_j)d\mathrm{u}$ in Eq. (4.4) is the normalization constant that indicates the share of $f(\mathbf{X}|\xi_j)$ which belongs to the support region $\partial$. The AGD $f(\mathbf{X}|\xi_j)$ can also be defined as:

$$f(\mathbf{X}|\xi_j) = \begin{cases} \mathrm{g}_1(\mathbf{X}|\xi_j) & \text{if } \mathrm{X}_d < \mu_{jd} \\ \mathrm{g}_2(\mathbf{X}|\xi_j) & \text{if } \mathrm{X}_d \geq \mu_{jd} \end{cases} \tag{4.7}$$

where

$$g_1(\mathbf{X}|\xi_j) = \prod_{d=1}^{D} \frac{2}{\sqrt{2\pi}(\sigma_{l_{jd}} + \sigma_{r\,jd})} \exp\left[-\frac{(X_d - \mu_{jd})^2}{2\sigma_{l_{jd}}^2}\right] \tag{4.8}$$

$$g_2(\mathbf{X}|\xi_j) = \prod_{d=1}^{D} \frac{2}{\sqrt{2\pi}(\sigma_{l_{jd}} + \sigma_{r\,jd})} \exp\left[-\frac{(X_d - \mu_{jd})^2}{2\sigma_{r_{jd}}^2}\right] \tag{4.9}$$

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

$$p(\mathcal{X}|\Theta) = \prod_{i=1}^{N} \sum_{j=1}^{K} p(\mathbf{X}_i|\xi_j) p_j \tag{4.10}$$

provided $p_j \geq 0$ and $\sum_{j=1}^{K} p_j = 1$. In Eq. (4.10), $\Theta$ represents the parameters of mixture model having $K$ classes as $\Theta = (\xi_1, \xi_2, \xi_3, \xi_4)$, where $\xi_1 = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$, $\xi_2 = (\boldsymbol{\sigma}_{l_1}, \ldots, \boldsymbol{\sigma}_{l_K})$, $\xi_3 = (\boldsymbol{\sigma}_{r_1}, \ldots, \boldsymbol{\sigma}_{r_K})$, and $\xi_4 = (p_1, \ldots, p_K)$.

Stochastic indicator vectors $\mathbf{Z}_i = (Z_{i1}, \ldots, Z_{iK})$, 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 $\mathbf{X}_i$ belongs to class $j$ and 0, otherwise. The complete-data likelihood is given below:

$$p(\mathcal{X}, \mathcal{Z}|\Theta) = \prod_{i=1}^{N} \prod_{j=1}^{K} \left(p(\mathbf{X}_i|\xi_j) p_j\right)^{Z_{ij}} \tag{4.11}$$

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

$$Z_{ij} = p(j|\mathbf{X}_i) = \frac{p(\mathbf{X}_i|\xi_j) p_j}{\sum_{j=1}^{K} p(\mathbf{X}_i|\xi_j) p_j} \tag{4.12}$$

and $\mathcal{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_N\}$.

### *4.2.4 Parameters Learning*

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

$$\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{K} Z_{ij} \log\left(p(\mathbf{X}_i|\xi_j)p_j\right) \tag{4.13}$$

$$\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{K} Z_{ij} \left\{ \log p_j + \log f(\mathbf{X_i}|\xi_j) + \log \mathrm{H}(\mathbf{X_i}|\Omega_j) - \log \int_{\partial_j} f(\mathrm{u}|\xi_j)d\mathrm{u} \right\} \tag{4.14}$$

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 $p_j$, $\mu_j$, $\sigma_{l_j}$, and $\sigma_{r_j}$. The parameter estimation for bounded support asymmetric Gaussian mixture model is explained below.

#### 4.2.4.1 Mixing Parameter Estimation

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:

$$\Phi(\mathcal{X}, \mathcal{Z}, \Theta, \Lambda) = \sum_{i=1}^{N}\sum_{j=1}^{K} Z_{ij} \log\left(p(\mathbf{X}_i|\xi_j)p_j\right) + \Lambda\left(1 - \sum_{j=1}^{K} p_j\right) \tag{4.15}$$

where $\Lambda$ is the Lagrange multiplier. Differentiating the augmented function with respect to $p_j$ we get:

$$\hat{p}_j = \frac{1}{N}\sum_{i=1}^{N} p(j|\mathbf{X}_i) \tag{4.16}$$

#### 4.2.4.2 Mean Parameter Estimation

The new value of mean $\mu_{jd}$ can be estimated by maximizing the log-likelihood function given in Eq. (4.14) with respect to $\boldsymbol{\mu}_j$.

$$\hat{\mu}_{jd} = \frac{\sum_{i=1}^{N} Z_{ij}\left\{\mathrm{X}_{id} - \frac{\int_{\partial_j} f(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})d\mathrm{x}}{\int_{\partial_j} f(\mathrm{u}|\xi_j)d\mathrm{u}}\right\}}{\sum_{i=1}^{N} Z_{ij}} \tag{4.17}$$

Note that, in Eq. (4.17), 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:

$$\int_{\partial_j} f(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})dx \approx \frac{1}{M}\sum_{m=1}^{M}(s_{m_{jd}}-\mu_{jd})\mathrm{H}(s_{m_{jd}}|\Omega_j) \tag{4.18}$$

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. (4.17) can be approximated as:

$$\int_{\partial_j} f(\mathrm{u}|\xi_j)dx \approx \frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(s_{m_{jd}}|\Omega_j) \tag{4.19}$$

$$\hat{\mu}_{jd} = \frac{\sum_{i=1}^{N} Z_{ij}\left\{\mathrm{X}_{id} - \frac{\sum_{m=1}^{M}(s_{m_{jd}}-\mu_{jd})\mathrm{H}(s_{m_{jd}}|\Omega_j)}{\sum_{m=1}^{M}\mathrm{H}(s_{m_{jd}}|\Omega_j)}\right\}}{\sum_{i=1}^{N} Z_{ij}} \tag{4.20}$$

#### 4.2.4.3 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. (4.14) with respect to $\boldsymbol{\sigma}_{l_j}$.

$$\frac{\partial \log[p(\mathcal{X},\mathcal{Z}|\Theta)]}{\partial \sigma_{l_{jd}}} = 0 \tag{4.21}$$

$$\frac{\partial \mathcal{L}(\mathcal{X},\mathcal{Z}|\Theta)}{\partial \sigma_{l_{jd}}} = \sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma_{l_{jd}}^3}\right) \tag{4.22}$$

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

$$\sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma_{l_{jd}}^3}\right) - \sum_{i=1,\mathrm{u}<\mu_{jd}}^{N}\frac{Z_{ij}}{\sigma_{l_{jd}}^3}\left\{\frac{\int_{\partial_j}\mathrm{g}_1(\mathrm{u}|\xi_j)dx(\mathrm{u}-\mu_{jd})^2dx}{\int_{\partial_j}\mathrm{g}_1(\mathrm{u}|\xi_j)dx}\right\} = 0 \tag{4.23}$$

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

$$\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^2 dx \approx \frac{1}{M}\sum_{m=1}^{M}(\mathrm{l}_{m_{jd}}-\mu_{jd})^2 \mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j) \tag{4.24}$$

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. (4.17) can be approximated as:

$$\int_{\partial_j} \mathrm{g}_1(\mathrm{u}|\xi_j)dx \approx \frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j) \tag{4.25}$$

$$\sum_{i=1,\mathrm{X}_{id}<\mu_{jd}}^{N} Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma^3_{l_{jd}}}\right) - \sum_{i=1}^{N}\frac{Z_{ij}}{\sigma^3_{l_{jd}}}\left\{\frac{\frac{1}{M}\sum_{m=1}^{M}(\mathrm{l}_{m_{jd}}-\mu_{jd})^2\mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j)}{\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{l}_{m_{jd}}|\Omega_j)}\right\} = 0 \tag{4.26}$$

It is noticed that Eq. (4.26) is non-linear, 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. (4.22) and (4.26).

$$\hat{\sigma}_{l_{jd}} \simeq \sigma_{l_{jd}} - \left[\left(\frac{\partial^2 \log[p(\mathcal{X},\mathcal{Z}|\Theta)]}{\partial {\sigma^2}_{l_{jd}}}\right)^{-1}\left(\frac{\partial \log[p(\mathcal{X},\mathcal{Z}|\Theta)]}{\partial \sigma_{l_{jd}}}\right)\right] \tag{4.27}$$

#### 4.2.4.4 Right Standard Deviation Estimation

Right standard deviation $\sigma_{r_{jd}}$ can be estimated by maximizing the log-likelihood function given in Eq. (4.14) with respect to $\boldsymbol{\sigma}_{r_j}$.

$$\frac{\partial \log[p(\mathcal{X},\mathcal{Z}|\Theta)]}{\partial \sigma_{r_{jd}}} = 0 \tag{4.28}$$

$$\frac{\partial \mathcal{L}(\mathcal{X},\mathcal{Z}|\Theta)}{\partial \sigma_{r_{jd}}} = \sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma^3_{r_{jd}}}\right) \tag{4.29}$$

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

$$\sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma^3_{r_{jd}}}\right)-\sum_{i=1,\mathrm{u}\geq\mu_{jd}}^{N}\frac{Z_{ij}}{\sigma^3_{r_{jd}}}\left\{\frac{\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)dx(\mathrm{u}-\mu_{jd})^2dx}{\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)dx}\right\}=0 \tag{4.30}$$

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

$$\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)(\mathrm{u}-\mu_{jd})^2dx\approx\frac{1}{M}\sum_{m=1}^{M}(\mathrm{r}_{m_{jd}}-\mu_{jd})^2\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j) \tag{4.31}$$

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. (4.17) can be approximated as:

$$\int_{\partial_j}\mathrm{g}_2(\mathrm{u}|\xi_j)dx\approx\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j) \tag{4.32}$$

$$\sum_{i=1,\mathrm{X}_{id}\geq\mu_{jd}}^{N} Z_{ij}\left(\frac{(\mathrm{X}_{id}-\mu_{jd})^2}{\sigma^3_{r_{jd}}}\right)-\sum_{i=1}^{N}\frac{Z_{ij}}{\sigma^3_{r_{jd}}}\left\{\frac{\frac{1}{M}\sum_{m=1}^{M}(\mathrm{r}_{m_{jd}}-\mu_{jd})^2\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j)}{\frac{1}{M}\sum_{m=1}^{M}\mathrm{H}(\mathrm{r}_{m_{jd}}|\Omega_j)}\right\}=0 \tag{4.33}$$

It is noticed that Eq. (4.33) 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. (4.29) and (4.33).

$$\hat{\sigma}_{r_{jd}}\simeq\sigma_{r_{jd}}-\left[\left(\frac{\partial^2\log[p(\mathcal{X},\mathcal{Z}|\Theta)]}{\partial{\sigma^2}_{r_{jd}}}\right)^{-1}\left(\frac{\partial\log[p(\mathcal{X},\mathcal{Z}|\Theta)]}{\partial\sigma_{r_{jd}}}\right)\right] \tag{4.34}$$

The complete learning of BAGMM is given in Algorithm 1, 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.

## 4.3 Textual Spam Detection

Email has become the prominent choice of communication, particularly for professional purposes [58]. 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

**Algorithm 1** Model learning for BAGMM

**Input**:Dataset $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$, $t_{\min}$.
**Output**: $\Theta$, $\mathcal{Z}$.
**{Initialization}**:
  $K$-Means Algorithm (Computation of $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ & cluster assignment)
    **for all** $1 \leq j \leq K$ do
      Computation of $p_j$
      Computation of $\{(\boldsymbol{\sigma}_{l_j}$ & $\boldsymbol{\sigma}_{r_j})$=$\boldsymbol{\sigma}_j\}$
    **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 $p(j|\mathbf{X}_i)$ for $i = 1, \ldots, N$. using Eq. (4.12).
    **end for**
  **{[M step]}**:
    **for all** $1 \leq j \leq K$ do
      Estimation of mixing parameter $p_j$ using Eq. (4.16).
      Estimation of mean $\boldsymbol{\mu}_j$ using Eq. (4.20).
      Estimation of left standard deviation $\boldsymbol{\sigma}_{l_j}$ using Eq. (4.27).
      Estimation of right standard deviation $\boldsymbol{\sigma}_{r_j}$ using Eq. (4.34).
    **end for**
**end while**

as phishing [25]. 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 [49]. Therefore, it is crucial that spam instances be efficiently and accurately detected and removed to avoid wasting additional efforts.

Recent works applying Gaussian mixture models on spam detection have shown their efficiency and modeling capabilities [18, 55]. 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 performance of clustering tasks is usually examined based on the accuracy computed as: $\left(\frac{TP+TN}{TP+TN+FP+FN}\right)$, which is the ratio of correctly predicted instances to all the instances. Here the term $TP$ stands for true positives, $TN$ for true negatives, $FP$ for false positives, and $FN$ stands for false negatives. However, for spam detection, accuracy alone is not sufficient to conclude the effectiveness of clustering approach. In other words, we also need to consider other essential metrics, namely precision $\left(\frac{TP}{TP+FP}\right)$ meaning the ratio of accurately returned spams to all the returned ones, sensitivity $\left(\frac{TP}{TP+FN}\right)$ which is the ratio of the

correctly predicted spams to the total actual spams, specificity $\left(\frac{TN}{TN+FP}\right)$ describing the proportion of the correctly predicted not-spam to all the actual not-spam, and false positive rate $\left(\frac{FP}{FP+TN}\right)$, the ratio of inaccurate predicted spams to all actual non-spams. In addition, particularly in case of imbalance in clusters' weights, we must also examine the F1-Score $\left(\left(\frac{(\beta^2+1)\times Sens\times Prec}{Sens+\beta\times Prec}\right), \beta > 0\right)$, which is the harmonic mean of precision and sensitivity; G-mean 1 $(\sqrt{Prec \times Sens})$, the geometric mean of precision and sensitivity; G-mean 2 $(\sqrt{Specs \times Sens})$, the geometric mean of specificity and sensitivity; Mathew's correlation coefficient (MCC) $\left(\frac{TP\cdot TN - FP\cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}\right)$: a performance metric to measure the quality of classification [13, 19, 29, 30].

The spambase dataset [10] 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. 4.3 and results in Table 4.1 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.

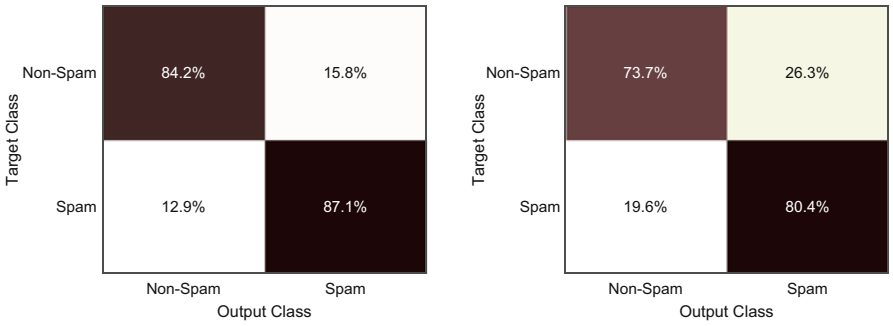



**Fig. 4.3** Confusion matrix of spambase dataset with BAGMM and AGMM, respectively

**Table 4.1** Performance of spambase data clustering based on different metrics

| | Performance metrics (%) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Models | 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 |

## 4.4 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 [53]. 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 misclassification with the noises. Recent clustering analyses using mixture models have shown good results on numerous categorization problems, namely scenes [26], sport activities [14], medial related images [61], and 3D objects [2]. Thus, the prospective progress has motivated the authors to apply the proposed model on this challenging task with two widely used datasets: Caltech 101 [16] and Corel [35, 36].

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) [37]. 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.

### *4.4.1 Experiments and Results*

#### 4.4.1.1 Experimental Framework and Results: Caltech 101 Dataset

In this subsection, we used the Caltech 101 dataset for object clustering. This dataset is popular [16, 20, 23, 24] which has demonstrated its effectiveness for object categorization using different algorithms [46, 62], techniques [3], and feature extraction methods [33, 41, 50, 54] 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 good quality, the categories are well annotated, selected, and have pose variation controlled. For the experimentation, we have used 5 classes, namely "brain," "bonsai," "airplane," "faces," and "motorbikes" where these classes contain 98, 128, 800, 435, and 798 images, respectively. Some examples of images from these classes are given in Fig. 4.4. After several experiments, we examined that optimal vocabulary size is 50 and hence, BOVW gives a matrix having a size of $2259 \times 50$, 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 Sect. 4.3. For comparison, we

**Fig. 4.4** Sample images of each class of Caltech 101 dataset

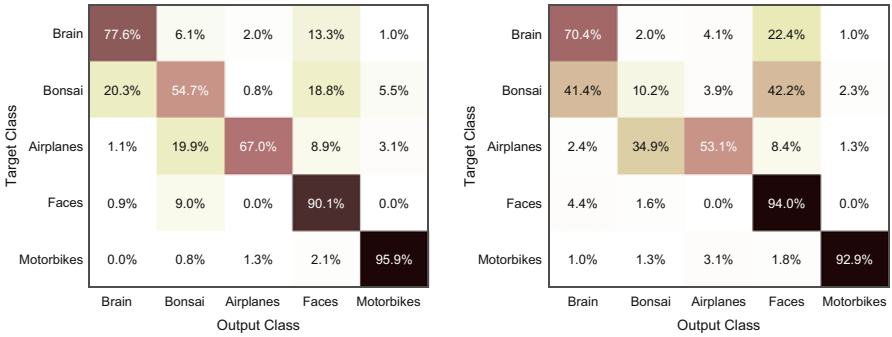



**Fig. 4.5** Confusion matrix of Caltech 101 dataset with BAGMM and AGMM, respectively

**Table 4.2** Performance of object data clustering (Caltech 101) based on different metrics

| | Performance metrics (%) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Models | 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 |

have implemented the same framework with AGMM. In this data clustering task, the distribution of classes is not balanced which makes it difficult to differentiate between different classes and it is depicted from the confusion matrix provided for AGMM as shown in Fig. 4.5. 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 4.2.

#### 4.4.1.2 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, 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. 4.6. After feature extraction, BOVW is a matrix of dimension $500 \times 50$, where columns represent the frequency of visual words and

**Fig. 4.6** Sample images of each class of Corel dataset

**Table 4.3** Performance of object data clustering (Corel dataset) based on different metrics

| | Performance metrics (%) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Models | 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 |

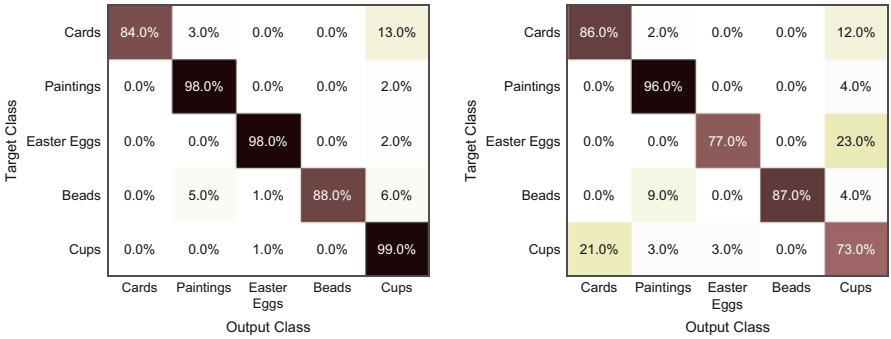



**Fig. 4.7** Confusion matrix of Corel dataset with BAGMM and AGMM, respectively

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 Sect. 4.3. 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 4.3 and confusion matrix in Fig. 4.7, 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%.

## 4.5 Texture Image Clustering

Texture is a fundamental element of human visual impression towards the world [28]. Indeed, understanding different textures is very beneficial for further complicated object classification, segmentation analyses, which includes various objects and surface types [6]. In order to counter issues, namely noise, complexity, slow convergence, and over-fitting, feature extraction is required. Various types of feature

extraction methods exist [7]. But, the co-occurrence matrix is a popular feature extraction technique when it comes to texture data [8, 9, 51]. Thus, co-occurrence matrix is used to extract the texture characteristics [21]. The co-occurrences are calculated with respect to their neighbors: (1;0), $\left(1; \frac{\pi}{4}\right)$, $\left(1; \frac{\pi}{2}\right)$, and $\left(3; \frac{\pi}{4}\right)$. 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.

## *4.5.1 Experiments and Results*

### 4.5.1.1 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 [45]. 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 contain 192, 320, 448, and 128 sub-images. Some examples of images from VisTex dataset are given in Fig. 4.8. The data matrix after feature extraction is provided to BAGMM for data clustering. In order to validate our proposed algorithm, we have used several performance metrics as described in Sect. 4.3. In order to have a comparison, we have implemented the same clustering framework with AGMM. From the results provided in Table 4.4, 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 makes the clustering task very difficult and it is obvious from the confusion matrix for AGMM in Fig. 4.9. By applying BAGMM, the clustering accuracy is improved tremendously and FPR is reduced from 10.10% to 7.20%.

**Fig. 4.8** Sample images of each class of VisTex dataset

**Table 4.4** 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 |

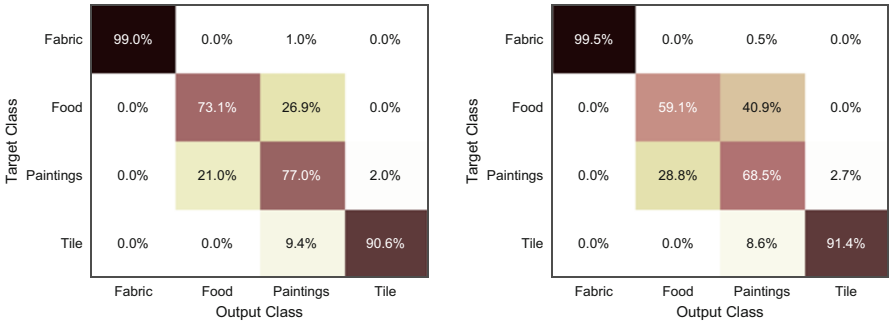



**Fig. 4.9** Confusion matrix of VisTex dataset with BAGMM and AGMM, respectively

## 4.6 Conclusion

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.

**Acknowledgement** The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC).

## References


1. Ahmad, T., Jameel, A., Ahmad, B.: Pattern recognition using statistical and neural techniques. In: International Conference on Computer Networks and Information Technology, pp. 87–91 (2011). https://doi.org/10.1109/ICCNIT.2011.6020913
2. Bdiri, T., Bouguila, N., Ziou, D.: Object clustering and recognition using multi-finite mixtures for semantic classes and hierarchy modeling. Expert Syst. Appl. **41**(4), 1218–1235 (2014)
3. Berg, A.C., Berg, T.L., Malik, J.: Shape matching and object recognition using low distortion correspondences. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, pp. 26–33. IEEE, Piscataway (2005)
4. Bouguila, N., ElGuebaly, W.: On discrete data clustering. In: Proceedings of the 12th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, PAKDD'08, pp. 503–510. Springer, Berlin (2008). http://dl.acm.org/citation.cfm?id=1786574.1786622
5. Bouguila, N., Ziou, D.: A nonparametric Bayesian learning model: Application to text and image categorization. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 463–474. Springer, Berlin (2009)
6. Braunl, T., Stefan Feyrer, D.I., Wolfgang Rapf, D.I., Michael Reinhardt, D.I.: Texture recognition. In: Parallel Image Processing, pp. 121–130. Springer, Berlin (2001)
7. Chadha, A., Mallik, S., Johar, R.: Comparative study and optimization of feature-extraction techniques for content based image retrieval. arXiv preprint arXiv:1208.6335 (2012)
8. Clausi, D.A., Jernigan, M.E.: A fast method to determine co-occurrence texture features. IEEE Trans. Geosci. Remote Sens. **36**(1), 298–300 (1998). https://doi.org/10.1109/36.655338
9. De Siqueira, F.R., Schwartz, W.R., Pedrini, H.: Multi-scale gray level co-occurrence matrices for texture description. Neurocomputing **120**, 336–345 (2013)
10. Dua, D., Graff, C.: UCI machine learning repository (2017)
11. Elguebaly, T., Bouguila, N.: Background subtraction using finite mixtures of asymmetric Gaussian distributions and shadow detection. Mach. Vis. Appl. **25**(5), 1145–1162 (2014)
12. Elguebaly, T., Bouguila, N.: Simultaneous high-dimensional clustering and feature selection using asymmetric Gaussian mixture models. Image Vis. Comput. **34**, 27–41 (2015)
13. Espindola, R., Ebecken, N.: On extending f-measure and G-mean metrics to multi-class problems. WIT Trans. Inf. Commun. Technol. **35**, 10 (2005)
14. Fan, W., Bouguila, N., Ziou, D.: Variational learning for finite Dirichlet mixture models and applications. IEEE Trans. Neural Netw. Learn. Syst. **23**(5), 762–774 (2012)
15. Farag, A., El-Baz, A., Gimel'farb, G.: Precise segmentation of multimodal images. IEEE Trans. Image Process. **15**(4), 952–968 (2006). https://doi.org/10.1109/TIP.2005.863949
16. Fei-Fei, L., Fergus, R., Perona, P.: 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, pp. 178–178 (2004)
17. Figueiredo, M.A., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **24**(3), 381–396 (2002)
18. Fu, S., Bouguila, N.: Asymmetric Gaussian mixtures with reversible jump MCMC. In: 2018 IEEE Canadian Conference on Electrical Computer Engineering (CCECE), pp. 1–4. IEEE, Piscataway (2018). https://doi.org/10.1109/CCECE.2018.8447816
19. Gorodkin, J.: Comparing two $K$-category assignments by a $K$-category correlation coefficient. Comput. Biol. Chem. **28**(5–6), 367–374 (2004)
20. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: Tenth IEEE International Conference on Computer Vision, ICCV, pp. 1458–1465. IEEE Computer Society, Silver Spring (2005)
21. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Trans. Syst. Man Cybern. **SMC-3**(6), 610–621 (1973)
22. Hedelin, P., Skoglund, J.: Vector quantization based on Gaussian mixture models. IEEE Trans. Speech Audio Process. **8**(4), 385–401 (2000). https://doi.org/10.1109/89.848220

23. Holub, A., Welling, M., Perona, P.: Exploiting unlabelled data for hybrid object classification. In: Proceedings of the Neural Information Processing Systems. Workshop Inter-Class Transfer, vol. 7, p. 2 (2005)
24. Holub, A.D., Welling, M., Perona, P.: Combining generative models and fisher kernels for object recognition. In: Tenth IEEE International Conference on Computer Vision (ICCV'05), vol. 1, pp. 136–143. IEEE, Piscataway (2005)
25. Hong, J.: The state of phishing attacks. Commun. ACM **55**(1), 74–81 (2012)
26. Ihou, K.E., Bouguila, N.: Variational-based latent generalized Dirichlet allocation model in the collapsed space and applications. Neurocomputing **332**, 372–395 (2019)
27. Ji, Z., Huang, Y., Sun, Q., Cao, G.: A spatially constrained generative asymmetric Gaussian mixture model for image segmentation. J. Vis. Commun. Image Represent. **40**, 611–626 (2016)
28. Jian, M., Liu, L., Guo, F.: Texture image classification using perceptual texture features and Gabor wavelet features. In: 2009 Asia-Pacific Conference on Information Processing, vol. 2, pp. 55–58 (2009). https://doi.org/10.1109/APCIP.2009.150
29. Jurman, G., Furlanello, C.: A unifying view for performance measures in multi-class prediction. arXiv preprint arXiv:1008.2908 (2010)
30. Jurman, G., Riccadonna, S., Furlanello, C.: A comparison of MCC and CEN error measures in multi-class prediction. PLoS One **7**(8), e41882 (2012)
31. Kato, T., Omachi, S., Aso, H.: 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), pp. 405–413. Springer, Berlin (2002)
32. Khodaskar, A.A., Ladhake, S.A.: Pattern recognition: Advanced development, techniques and application for image retrieval. In: 2014 International Conference on Communication and Network Technologies, pp. 74–78 (2014). https://doi.org/10.1109/CNT.2014.7062728
33. Lazebnik, S., Schmid, C., Ponce, J.: 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, pp. 2169–2178. IEEE, Piscataway (2006)
34. Lindblom, J., Samuelsson, J.: Bounded support Gaussian mixture modeling of speech spectra. IEEE Trans. Speech Audio Process. **11**(1), 88–99 (2003). https://doi.org/10.1109/TSA.2002.805639
35. Liu, G.H., Li, Z.Y., Zhang, L., Xu, Y.: Image retrieval based on micro-structure descriptor. Pattern Recogn. **44**(9), 2123–2133 (2011)
36. Liu, G.H., Yang, J.Y., Li, Z.: Content-based image retrieval using computational visual attention model. Pattern Recogn. **48**(8), 2554–2566 (2015)
37. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
38. Ma, H., Chan, J.C., Saha, T.K., Ekanayake, C.: Pattern recognition techniques and their applications for automatic classification of artificial partial discharge sources. IEEE Trans. Dielectr. Electr. Insul. **20**(2), 468–478 (2013). https://doi.org/10.1109/TDEI.2013.6508749
39. Ma, J., Jiang, J., Liu, C., Li, Y.: Feature guided Gaussian mixture model with semi-supervised EM and local geometric constraint for retinal image registration. Inf. Sci. **417**, 128–142 (2017)
40. Malsiner-Walli, G., Frühwirth-Schnatter, S., Grün, B.: Model-based clustering based on sparse finite Gaussian mixtures. Stat. Comput. **26**(1–2), 303–324 (2016)
41. Marin-Jimenez, M.J., De La Blanca, N.P.: Empirical study of multi-scale filter banks for object categorization. In: International Conference on Pattern Recognition, ICPR, pp. 578–581. IEEE, Piscataway (2006)
42. Mclachlan, G., Basford, K.: Mixture models: Inference and applications to clustering. J. R. Stat. Soc. Ser. C **38**(2), 384–385 (1989). https://doi.org/10.2307/2348072
43. McLachlan, G., Krishnan, T.: The EM Algorithm and Extensions, vol. 382. Wiley, London (2007)
44. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley, London (2004)
45. MIT Media Lab.: Vistex texture database (1995). https://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html

46. Mutch, J., Lowe, D.G.: Multiclass object recognition with sparse, localized features. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 1, pp. 11–18. IEEE, Piscataway (2006)
47. Neal, R.M., Hinton, G.E.: A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Learning in graphical models, pp. 355–368. Springer, Berlin (1998)
48. Nguyen, T.M., Wu, Q.J., Zhang, H.: Bounded generalized Gaussian mixture model. Pattern Recogn. **47**(9) (2014)
49. Park, I., Sharman, R., Rao, H.R., Upadhyaya, S.: The effect of spam and privacy concerns on e-mail users' behavior. J. Trans. Inf. Syst. Secur. **3**(1), 39–62 (2016)
50. Serre, T., Wolf, L., Poggio, T.: Object recognition with features inspired by visual cortex. Tech. rep., Massachusetts Institute of Technology Cambridge Department of Brain and Cognitive Sciences (2006)
51. Tang, X.: Texture information in run-length matrices. IEEE Trans. Image Process. **7**(11), 1602–1609 (1998)
52. Titterington, D., Smith, A., Makov, U.: Statistical Analysis of Finite Mixture Distributions. Wiley, New York (1985)
53. Viitaniemi, V., Laaksonen, J.: Techniques for still image scene classification and object detection. In: Kollias, S., Stafylopatis, A., Duch, W., Oja, E. (eds.) Artificial Neural Networks – ICANN 2006, pp. 35–44. Springer, Berlin (2006)
54. Wang, G., Zhang, Y., Fei-Fei, L.: 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, pp. 1597–1604. IEEE, Piscataway (2006)
55. Wang, M., Zhang, W., Zhang, Y., Ji, X.: Detecting image spam based on cross entropy. In: 2011 Eighth Web Information Systems and Applications Conference, pp. 19–22 (2011). https://doi.org/10.1109/WISA.2011.11
56. Wang, G., Wang, Z., Chen, Y., Zhao, W.: A robust non-rigid point set registration method based on asymmetric Gaussian representation. Comput. Vis. Image Underst. **141**, 67–80 (2015)
57. Xu, L., Jordan, M.I.: On convergence properties of the EM algorithm for Gaussian mixtures. Neural Comput. **8**(1), 129–151 (1996)
58. Xu, H., Yu, B.: Automatic thesaurus construction for spam filtering using revised back propagation neural network. Expert Syst. Appl. **37**(1), 18 – 23 (2010)
59. Yang, M.H., Ahuja, N.: Gaussian mixture model for human skin color and its applications in image and video databases. In: Storage and retrieval for image and video databases VII, vol. 3656, pp. 458–467. International Society for Optics and Photonics (1998)
60. Yang, J., Liao, X., Yuan, X., Llull, P., Brady, D.J., Sapiro, G., Carin, L.: Compressive sensing by learning a Gaussian mixture model from measurements. IEEE Trans. Image Process. **24**(1), 106–119 (2015)
61. Yin, D., Pan, J., Chen, P., Zhang, R.: Medical image categorization based on Gaussian mixture model. In: 2008 International Conference on BioMedical Engineering and Informatics, vol. 2, pp. 128–131 (2008)
62. Zhang, H., Berg, A.C., Maire, M., Malik, J.: 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, pp. 2126–2136. IEEE, Piscataway (2006)

# Chapter 5
# Online Recognition via a Finite Mixture of Multivariate Generalized Gaussian Distributions


**Fatma Najar, Sami Bourouis, Rula Al-Azawi, and Ali Al-Badi**

**Abstract** The huge amount of data expanding day by day entail creating powerful real-time algorithms. Such algorithms allow a reactive processing between the input multimedia data and the system. In particular, we are mainly concerned with active learning and clustering images and videos for the purpose of pattern recognition. In this paper, we propose a novel online recognition algorithm based on multivariate generalized Gaussian distributions. We estimate at first the generative model's parameters within a discriminative framework (fixed-point, Riemannian averaged fixed-point, and Fisher scoring). Then, we propose an online recognition algorithm in accordance with those algorithms. Finally, we applied our proposed framework on three challenging problems, namely: human action recognition, facial expression recognition, and pedestrian detection from infrared images. Experiments demonstrate the robustness of our approach by comparing with the state-of-the art algorithms and offline learning techniques.

## 5.1 Introduction

Online, real-time sequential arrival of data has increased the computer science community efforts to analyze, understand, and extract information. Despite the

F. Najar (✉)
Laboratoire RISC Robotique Informatique et Systèmes Complexes, Université de Tunis El Manar, ENIT, Tunis, Tunisie
e-mail: fatma.najjar@enit.utm.tn

S. Bourouis
Taif University, Taif, Saudi Arabia

Université de Tunis El Manar, LR-SITI Laboratoire Signal, Image et Technologies de l'Information, Tunis, Tunisie
e-mail: s.bourouis@tu.edu.sa

R. Al-Azawi · A. Al-Badi
Gulf College, Al Maabelah, Muscat, Oman
e-mail: aalbadi@gulfcollege.edu.om



N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6_5

fact that information is continuously changing in real time and cannot be available at once, the traditional learning approach remains constant. In fact, when data is generated in a function of time, we need to incrementally assemble data as long as they arrive in a time sequence. Besides, when the size of data is out of the memory limits, it will be computationally infeasible to train over the entire dataset. In order to meet these necessities, online learning has been emerged to deal with data in an incremental process, react to new data, and predict the future coming inputs. As the notation suggests, online learning is an online method that processes information at a time. The core idea of this learning algorithm is to generate a model from training on a stored dataset and then using an iterative algorithm like stochastic gradient descent and recursive least squares to learn new data introduced dynamically to the model.

Researchers have made interesting progress in developing online approaches in several research fields including machine learning, pattern recognition, computer vision, game theory, and information theory. Online learning is important for various applications such as faster clustering, forecasting times, catastrophic interference, spam filtering, pattern recognition, and online tracking. Among the extensive related work in this field, we cite the most interesting approaches proposed in literature.

Recognizing human activity is an active research topic where the need to identify real-time moves and actions continuously over the time remains a challenging problem. Authors in [46] propose an online method to recognize human gestures through discriminative key poses and speed-aware action graphs. In [50], a hidden Markov model with modified short-time Viterbi algorithm was proposed for online recognizing human daily activity. In order to deal with the problem of clustering parallel data streams, the authors in [5] develop an online version of the classical K-means clustering algorithm. The idea of this method is based on an incremental computation of distances between streams of data using a DFT approximation. A probabilistic model was proposed for online clustering in [48] to detect the novel objects from sequences of data. They used a non-parametric Dirichlet process for modeling documents in an online fashion and an empirical Bayes method to estimate model hyperparameters. When features are expensive, authors in [39] proposed a novel online feature selection allowing the feature to be only available one at a time. This online framework was based on grafting approach that combines the speed of filters with the accuracy of wrappers. Applied to spam filtering, an online model has been presented to filter a sequence of emails using distance-based kernels and string kernels in [2]. In this paper, we consider particularly the problem of online recognition which is one of the most important problems that arises in computer vision, image analysis, information retrieval, data compression, and pattern recognition. Online cluster analysis is the task of grouping data into homogeneous clusters as long as they arrive in a temporal sequence. Finite mixture model is among the most applied approaches in the context of machine learning applications [11–13, 19, 33], especially for online clustering. In [41], an online approach was introduced based on a stochastic approximation of the Expectation-Maximization algorithm for the normalized Gaussian network. Experiments results showed that this online EM-algorithm for the NGnet is able to manage dynamic

environments and to deal efficiently with the robot dynamics problem. In video surveillance application, an adaptive Gaussian mixture model [29] has been used to model real video data with an incremental EM for the learning update. In [51], Gaussian mixture models have been proposed in an online fashion based on description length reducing prior and a MAP estimation procedure for an up-to-date description of the data. Despite the adoption of this model to various online clustering because of its simplicity, real-world applications cannot be considered by the Gaussian assumption which fails to fit the shape of the data. For instance, recent works have shown that other non-Gaussian models such as the Dirichlet, the generalized Gaussian, and the Beta Liouville mixtures provide better clustering results in several applications. In [8], a finite mixture of Dirichlet and a stochastic approach was proposed in the light of online clustering application, namely the dynamic summarization of image databases. A more general distribution has been applied to this type of non-Gaussian data is the generalized Dirichlet. The authors in [18] have proposed an online variational learning of generalized Dirichlet mixture models with feature selection to challenging problems, namely text clustering and image clustering using the bag-of-visual-words representation. Another approach that can control how the system should perceive new coming data over time is based on the generalized inverted Dirichlet [4]. For recognizing human facial expression, an online variational learning based on Beta-Liouville mixture model was proposed in [17]. A novel approach based on spherical mixtures has been proposed in [3] to tackle the problem of tracking and detecting news topic trend. The model in [1], besides, proposes a flexible online clustering algorithm in order to accurately approximate the non-Gaussian data. This online technique, based on finite mixture of generalized Gaussian distribution, has been applied to video foreground segmentation. In fact, generalized Gaussian mixture models have been the subject of wide applications [16, 26, 40]. However, in many multivariate statistical processes, generalized Gaussian distribution fails to be as accurate as the multivariate generalized Gaussian mixture as shown in previous works [32, 34, 35]. In fact, authors have proved that this multivariate mixture model is able to efficiently recognize human activity. Based on these studies, it is concluded that it is interesting to build our online framework based on the multivariate generalized Gaussian mixture model. One of the fundamental tasks of finite mixture model is parameter estimation, usually related to optimization problem.

The question to ask then is this: how to recursively estimate the parameters of the mixture of multivariate generalized Gaussian distributions and how to simultaneously select the number of components? In this paper, we seek to answer this question by improving our previous deterministic approaches proposed in [34, 35] and presenting a novel online recognition algorithm based on multivariate generalized Gaussian mixture model suitable for various applications. We are mainly interested by recognizing the human actions, facial expression from videos and detecting pedestrian from infrared images.

This paper is organized as follows: Sect. 5.2 proposes the deterministic framework based on multivariate generalized Gaussian mixture model. In Sect. 5.3, we introduce our novel online learning algorithm. Next, we applied the proposed algo-

rithms to the problem of human action recognition, facial expression recognition, and pedestrian detection from infrared images. Finally, we conclude this paper with a summary and potential future works in Sect. 5.4.

## 5.2 Multivariate Generalized Gaussian Mixture Model

### 5.2.1 Multivariate Generalized Gaussian Distribution

Multivariate generalized Gaussian distribution (MGGD) is defined by the probability density functions [25] as follows:

$$p(\mathbf{X}|\Sigma;\beta;\boldsymbol{\mu}) = \frac{\Gamma(\frac{d}{2})}{\pi^{\frac{d}{2}}\Gamma(\frac{d}{2\beta})2^{\frac{d}{2\beta}}}\frac{\beta}{|\Sigma|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}((\mathbf{X}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{X}-\boldsymbol{\mu}))^{\beta}\right] \quad (5.1)$$

where $\mathbf{X} \in R^d$, $\Sigma = mM$ is a $d \times d$ symmetric positive definite matrix, called the dispersion matrix, $\boldsymbol{\mu}$ is a d-dimensional mean vector, and $\beta > 0$ is the shape parameter that we assumed to be the same for all the dimensions of the data. Noting now that if $\beta = 1$, the MGGD is equivalent to the multivariate Gaussian distribution. The shape parameter $\beta$ controls the peakedness and the spread of the distribution. The smaller the beta, the more peaked for the probability distribution function (pdf), and the larger the beta, the flatter will be the pdf just as exposed in Fig. 5.1b. Positive shape parameter values produce skewed distributions to the left and bounded to the right. In contrast, negative shape parameter values produce skewed distributions to the right and bounded to the left.

### 5.2.2 Finite Mixture and Deterministic Learning

The finite mixture of K multivariate generalized Gaussian distributions is given by:

$$p(\mathbf{X}|\Theta) = \sum_{j=1}^{K} p_j\, p(\mathbf{X}|\Theta_j), \quad (5.2)$$

where $p(\mathbf{X}|\Theta_j)$ is known as the $j$th component of the mixture defined with its parameters $\Theta_j = (\boldsymbol{\mu}_j, \Sigma_j, \beta_j)$. The parameter $p_j$ is called a mixing weight parameter and must satisfy $0 \leq p_j \leq 1$ together with $\sum_{j=1}^{K} p_j = 1$.

The main purpose of deterministic techniques is maximizing the likelihood function with respect to model's parameters. One of the standard inferential methods and the powerful tool used to fit Gaussian based-mixture model to an observed data is the Expectation-Maximization (EM) algorithm [15]. Its aim is to optimize the

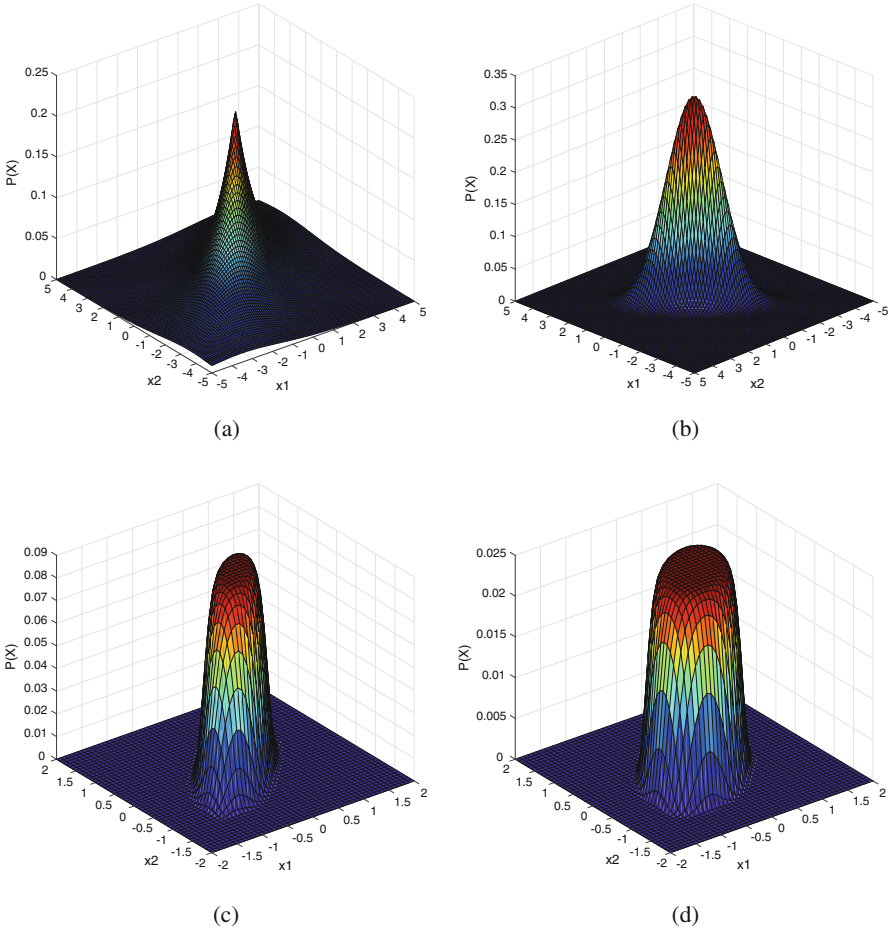



**Fig. 5.1** Multivariate generalized Gaussian distributions with different shape parameters. (**a**) $\beta = 0.5$. (**b**) $\beta = 1$. (**c**) $\beta = 3$. (**d**) $\beta = 5$

likelihood function in regard to the model's parameters. The EM algorithm starts with initializing parameters $\Theta_0$. Then, it iterates between two steps: the expectation and the maximization and converges to the maximum. In the expectation step (E-step), the expected likelihood is estimated given the current estimated parameters. For that purpose, the following posterior probability, named also responsibilities, for the j-th component of the mixture is computed:

$$p(j|\mathbf{X}) = \frac{p_j\, p(\mathbf{X}|\Theta_j)}{\sum_{m=1}^{K} p_m\, p(\mathbf{X}|\Theta_m)} \tag{5.3}$$

During the maximization step (M-step), the model's parameters are updated using the current responsibilities. In order to maximize the likelihood function, the log-likelihood function is maximized instead with respect to parameters as it is a monotone function. Then applying the logarithm to the likelihood function, it follows for $\mathcal{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$ that :

$$\mathcal{L}(\mathcal{X}|\Theta) = \sum_{i=1}^{N} \log p(\mathbf{X}_i|\Theta) = \sum_{i=1}^{N} \log \left( \sum_{j=1}^{K} p_j \, p(\mathbf{X}_i|\Theta_j) \right) \tag{5.4}$$

The M-step can be formally described as solving directly the following equation:

$$\frac{\partial \mathcal{L}(\mathcal{X}|\Theta)}{\partial \Theta_j} = 0 \tag{5.5}$$

Given the multivariate generalized Gaussian distribution $p(\mathbf{X}|\boldsymbol{\mu}_j, \Sigma_j, \beta_j)$, we obtain the following estimates :

- Mixing parameter

$$\hat{p}_j = \frac{1}{N} \sum_{i=1}^{N} p(j|\mathbf{X}_i) \tag{5.6}$$

- Mean parameter

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_{i=1}^{N} p(j|X_i)|X_i - \boldsymbol{\mu}_j|^{\beta_j - 1} X_i}{\sum_{i=1}^{N} p(j|X_i)|X_i - \boldsymbol{\mu}_j|^{\beta_j - 1}} \tag{5.7}$$

As there is no closed form for the covariance matrix and the likelihood estimation for this parameter is indistinct, the authors in [9, 10, 38] have proven the existence of the covariance matrix estimator up to certain conditions. We present those mentioned works in the following section.

#### 5.2.2.1 Fixed-Point Estimation Method

One of the above-mentioned parameters estimation techniques of the MGGD is the so-called fixed-point method [38]. Indeed, this method guarantees the existence and uniqueness of the MLE of the covariance matrix for each shape parameter belonging to [0,1]. The existence was proved by showing that the profile likelihood is positive, bounded in the set of symmetric positive definite matrices and equals to zero on the boundary of this set. Regarding the uniqueness, it was proved that for any initial symmetric positive definite matrix, the sequence of matrices satisfying a fixed point equation converges to the unique maximum of this profile likelihood.

Let $(X_1, X_2, \ldots, X_N)$ be a random sample of $N$ observation vectors of dimension $d$, drawn from a zero-mean MGGD with scatter matrix $C = m\Sigma$; m is the scale parameter, and $\beta$ is the shape parameter. The MLE of $m$, $\beta$, and $\Sigma$ are found by solving the maximum likelihood equations defined as follows:

$$\hat{m} = \left[ \frac{1}{N} \sum_{i=1}^{N} (u_i)^{\beta} \right]^{\frac{1}{\beta}}, \tag{5.8}$$

where $u_i = X_i^T \Sigma^{-1} X_i$.

Assuming first that $\beta$ is known. By differentiating the likelihood function with respect to $\Sigma$, the MLE of the covariance matrix satisfies the following fixed point (FP) equation:

$$f(\Sigma) = \sum_{i=1}^{N} \frac{d}{u_i + u_i^{1-\beta} \sum_{i \neq j} u_j^{\beta}} X_i X_i^T, \tag{5.9}$$

In other words, the fixed point equation can be written as:

$$\hat{\Sigma}_{k+1} = f(\Sigma_k) \tag{5.10}$$

Indeed, mathematically the solution of fixed point equation is settled using an iterative proceeding until $\Sigma$ stabilize (i.e., there is no sensible difference between $\Sigma_k$ and $\Sigma_{k+1}$).

Afterwards, an iterative algorithm based on a Newton–Raphson technique is then applied to compute the maximum likelihood estimation of the shape parameter.

$$\hat{\beta}_{k+1} = \hat{\beta}_k - \frac{\alpha(\hat{\beta}_k)}{\alpha'(\hat{\beta}_k)} \tag{5.11}$$

where

$$\begin{aligned} \alpha(\beta) = {} & \frac{dN}{2\sum_{i=1}^{N} u_i^{\beta}} \sum_{i=1}^{N} \left[ u_i^{\beta} \log(u_i) \right] - \frac{dN}{2\beta} \left[ \psi\left( \frac{d}{2\beta} \right) + \log(2) \right] \\ & - N - \frac{dN}{2\beta} \log\left( \frac{\beta}{dN} \sum_{i=1}^{N} u_i^{\beta} \right) \end{aligned} \tag{5.12}$$

where $\psi$ is the digamma function.

#### 5.2.2.2 Riemannian Averaged Fixed-Point Estimation Algorithm

The second developed algorithm is the named "Riemannian Averaged Fixed Point" method (RA-FP) [10]. The latter estimator is proposed as a generalization form of the previous proposed technique fixed-point (FP) estimator [38]. The basic idea of RA-FP algorithm is to implement successive Riemannian average of fixed point iterates in order to estimate the covariance matrix for any positive value of the shape parameter. This process is different from the fixed-point algorithm which estimates the covariance matrix for only the shape parameter belonging to [0, 1].

The RA-FP uses the Riemannian geometry for estimating the covariance matrix. The RA-FP based estimation of $\Sigma$ is determined as follows:

For $t \in [0, 1]$, the Riemannian average of $\hat{\Sigma}_{k+1}$ is defined as:

$$\begin{aligned} \hat{\Sigma}_{k+1} &= \Sigma_k \#_{t_k} f(\Sigma_k) \\ &= \Sigma_k^{1/2} (\Sigma_k^{-1/2} f(\Sigma_k) \Sigma_k^{-1/2})^{t_k} \Sigma_k^{1/2} \end{aligned} \tag{5.13}$$

where

$$t_k = \frac{1}{k+1}, t_k \in [0, 1], \tag{5.14}$$

and $f(\Sigma)$ is defined as (5.9).

If $t_k = 1$, the RA-FP estimator is reduced to the fixed-point estimator, and Eq. (5.13) yields to (5.10).

For computing the maximum likelihood of the shape parameter, an iterative algorithm based on a Newton–Raphson technique [10] is applied as in the fixed-point algorithm.

#### 5.2.2.3 Fisher Scoring Algorithm

The Fisher scoring algorithm [9] is a maximum likelihood estimator based on the fixed-point technique also and followed by an optimization through the Fisher scoring method. The estimators of $m, \beta$ are given by Eqs. (5.8) and (5.11) as proposed in [38]. Hence, in this work, the main purpose of the Fisher scoring algorithm is to optimize the likelihood function based on fixed-point technique and followed by an optimization iteration through the Fisher information matrix.

The likelihood function of vectors $\mathcal{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N)$ is given by:

$$\mathcal{L}(\mathcal{X}|\Theta) = \prod_{i=1}^{N} \sum_{j=1}^{K} p_j \, p(\mathbf{X}_i | \Sigma_j; \beta_j; \boldsymbol{\mu}_j) \tag{5.15}$$

The gradient of likelihood function with regard to covariance matrix is defined as:

$$\nabla \mathcal{L}(\Sigma; \Theta) = [F(\Sigma)]^{\frac{N-2}{2}} \nabla F(\Sigma), \tag{5.16}$$

with :

$$\begin{aligned} F : S_{++}^{K} &\rightarrow R^{+}\{0\} \\ \Sigma \quad &\rightarrow |\Sigma|^{-1} \Big( \sum_{i=1}^{N} u_i^{\beta} \Big)^{-\frac{d}{\beta}} \end{aligned} \tag{5.17}$$

and the gradient of F at a point $\Sigma$ is given by:

$$\nabla F(\Sigma) = F(\Sigma)\Sigma^{-1}[f(\Sigma) - \Sigma]\Sigma^{-1} \tag{5.18}$$

where $f(\Sigma)$ is defined by fixed-point algorithm in [38].

To numerically maximize the likelihood function, the Fisher-scoring iteration is given by:

$$\Sigma_{k+1} = \Sigma_k + G^{-1} \nabla L(\Sigma; \Theta) \tag{5.19}$$

where the entries of the Fisher information matrix are defined by [45]:

$$G_{ii}(\beta) = \frac{1}{4}\Big( \frac{3d + 6\beta}{d+2} - 1 \Big), \tag{5.20}$$

$$G_{ij}(\beta) = \frac{1}{4}\Big( \frac{d + 2\beta}{d+2} - 1 \Big), i \neq j, \tag{5.21}$$

for $i, j = 1, \ldots, K$.

Afterwards, an iterative algorithm based on Newton–Raphson technique is applied to compute the maximum likelihood estimation of the shape parameter as the two previous estimator algorithms.

We summarize the EM-algorithm for the multivariate generalized Gaussian mixture model in the following algorithm:

## 5.3 Online Learning Algorithm

The deterministic framework presented in the above section was based on batch learning; the parameters are updated on the entire dataset at once. In this section, we introduce an online EM learning approach. We suppose the dataset was represented by M multivariate generalized Gaussian distributions with parameters $\Theta_N$. Assume now at time $t + 1$, a new data $\mathbf{X}_{N+1}$ is inserted to the database, thus, we should

**Algorithm 1** MGGMM learning algorithm

**Require:** $\mathcal{X}$, $K$
**Ensure:** $\Theta^*$
  **Initialization**
  Apply the K-Means to obtain the parameters of each component.
  Then, apply the Method-of-Moment for each component j
  **repeat**
    **for** j:=1 **to** K **do**
      *E-step*: Compute posterior probabilities using Eq. (5.3)
      *M-step*:
      Update $\boldsymbol{\mu}_j$ using Eq. (5.7)
      Update $\beta_j$ using Eq. (5.11)
      Update $\Sigma_j$ using FP algorithm ((5.10), (5.9)), RA-FP algorithm (5.13) or FS algorithm (5.19).
    **end for**
  **until** Convergence of Likelihood
  Return the model's parameters $\Theta^*$.

update the different mixture model parameters with the new input vector. For which reason, a stochastic approximation for obtaining the maximum likelihood of mixture parameters was considered. Namely, we have used the stochastic ascent gradient parameter updating proposed in [47] where the updating parameters are given by:

$$\Theta_{N+1}^{(t+1)} = \Theta_N^{(t)} + \delta_N \frac{\partial \log(p(\mathbf{X}_{N+1}|\Theta_N^{(t)}, p_j^{(t)}))}{\partial \Theta_N^{(t)}} \tag{5.22}$$

In terms of updating the mixing weight, the above updating equation does not ensure the constraints: $0 \leq p_j \leq 1, \sum_{j=1}^{M} p_j = 1$. For this aim, a logit parameterization was presented to overcome this problem.

$$\pi_j^{(t)} = \log \frac{p_j}{p_M}, \; j = 1, \ldots, M-1 \tag{5.23}$$

$$\pi_j^{(t+1)} = \pi_j^{(t)} + \delta_N \left( Z_{N+1,j}^{(t+1)} - p_j^{(t)} \right), \; j = 1, \ldots, M-1 \tag{5.24}$$

So that, for $j = 1 \ldots, M-1$,the updating mixing weight is given by:

$$p_j^{(t+1)} = \frac{\exp(\pi_j^{(t+1)})}{1 + \sum_{j=1}^{M-1} \exp\left(\pi_j^{(t+1)}\right)}, \tag{5.25}$$

$$p_M^{(t+1)} = \frac{1}{1 + \sum_{j=1}^{M-1} \exp\left(\pi_j^{(t+1)}\right)} \tag{5.26}$$

The updating mean parameter, shape parameter, and covariance matrix are as follows:

$$\boldsymbol{\mu}_j^{(t+1)} = \boldsymbol{\mu}_j^{(t)} + \delta_N * Z_{N+1,j}^{(t+1)} * \left[ (\mathbf{X}_{N+1} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\mathbf{X}_{N+1} - \boldsymbol{\mu}_j) \right]^{\beta_j - 1} \tag{5.27}$$

$$\begin{aligned} \beta_j^{(t+1)} = \beta_j^{(t)} + \delta_N * Z_{N+1,j}^{(t+1)} * \Bigg[ & \left( \frac{1}{\beta_j} + \frac{d\psi(d/2\beta_j)}{2\beta_j^2} + \frac{d\log(2)}{2\beta_j^2} \right) \\ & -((\mathbf{X}_{N+1} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\mathbf{X}_{N+1} - \boldsymbol{\mu}_j)^{\beta_j}) \\ & \log((\mathbf{X}_{N+1} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\mathbf{X}_{N+1} - \boldsymbol{\mu}_j)) \Bigg] \end{aligned} \tag{5.28}$$

$$\begin{aligned} \Sigma_j^{(t+1)} = \Sigma_j^{(t)} + \delta_N * Z_{N+1,j}^{(t+1)} * \Bigg[ & \left( -\frac{1}{2} tr(\Sigma_j^{-1} d\Sigma) \right) \\ & + \frac{\beta_j}{2} (\mathbf{X}_{N+1} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} d\Sigma_j \Sigma_j^{-1} \\ & ((\mathbf{X}_{N+1} - \boldsymbol{\mu}_j)(\mathbf{X}_{N+1} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\mathbf{X}_{N+1} - \boldsymbol{\mu}_j))^{(\beta_j - 1)} \Bigg] \end{aligned} \tag{5.29}$$

Thus, the complete online updating MGGMM algorithm was resumed as follows.

**Algorithm 2** Online MGGMM learning algorithm

**Require:** $\mathcal{X} = \{X_1, \ldots, X_N\}, \Theta_N^{(t)}, K$
**Ensure:** $\Theta_{N+1}^{(t+1)}$
At $t+1$, new data vector $\mathbf{X}_{N+1}$
**repeat**
**for** j:=1 **to** K **do**
Compute posterior probability

$$z_{N+1j} = \frac{p_j p(\mathbf{X}_{N+1}|\Theta_j)}{\sum_{m=1}^{K} p_m p(\mathbf{X}_{N+1}|\Theta_m)} \tag{5.30}$$

Affect $\mathbf{X}_{N+1}$ to a cluster using the Bayes rule: $\mathbf{X}_{N+1}$ is affected to cluster $j_1$ if $z_{N+1j_1} > z_{N+1j}, \forall j \neq j_1$
Update the weights using Eqs. (5.25) and (5.26)
Update $\boldsymbol{\mu}_j$ using Eq. (5.27)
Update $\beta_j$ using Eq. (5.28)
Update $\Sigma_j$ using Eq. (5.29).
**end for**
**until** Convergence of Likelihood
Return the model's parameters $\Theta^{N+1}$.

## 5.4 Experiments Results

### *5.4.1 Datasets*

In our experiments, we are using three different datasets to evaluate the performance of the proposed online mixture model. For human action recognition, we use the well-known KTH dataset [42]. This human action dataset presents to date the most tremendous at handset of video sequences for human actions. It contains 2391 sequences categorized in six different human actions: walking, jogging, running, boxing, hand waving, and hand clapping. Each action is performed in four diverse scenarios: outdoors, outdoors with scale variations, outdoors with different clothes, and indoors. We present some examples of frames from video sequences in each category in Fig. 5.2. For recognizing facial expression, the set of data that we have used is the Cohn–Kanade dataset [22]. It contains 486 sequences where each sequence starts with a neutral expression and proceeds to a target expression of anger, surprise, joy, fear, sadness, or disgust. The sequences are collected from 97 university students ranging in age from 18 to 30 years. Sixty-five percent were female, 15% were African-American, and 3% were Asian or Latino with one to six emotions per subject. Sample images from this database with different facial expressions are shown in Fig. 5.3. In regard to the detecting pedestrian from infrared images, we make use of a challenging dataset of thermal imagery, namely the OSU thermal dataset. It is composed of 10 test collections with the total of 284 thermal images. Those images contain 984 pedestrians captured from Ohio State University campus using a Raytheon 300D thermal sensor core with 75 mm lens mounted on an 8-story building. We display an exemplary of different number of pedestrians in Fig. 5.4.

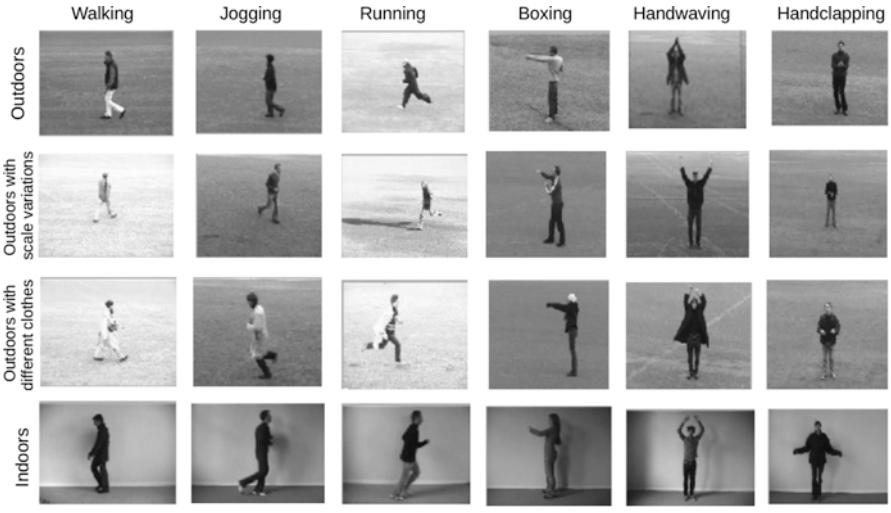



**Fig. 5.2** Examples of frames from the KTH dataset of different human actions within different scenarios

**Fig. 5.3** Sample face expression images from the Cohn–Kanade database


**Fig. 5.4** Example of pedestrian images from the OSU-thermal database

### *5.4.2 Database Preprocessing Approach*

The methodology that we have adopted for each application can be summarized as follows (Fig. 5.5). Basically, we have adopted the bag-of-words approach to represent our images and video sequences. In this model, each image or video of the dataset is depicted as a set of features. First, we extracted local spatio-temporal features from each video sequence from KTH database using space-time interest point detector [28] and SIFT3D descriptor [43]. From Cohn–Kanade videos dataset, we extracted dynamic textures features using LBP-TOP descriptor within $9 \times 8$ blocks [49]. Besides, for the infrared images, we used dense SIFT descriptors [30]

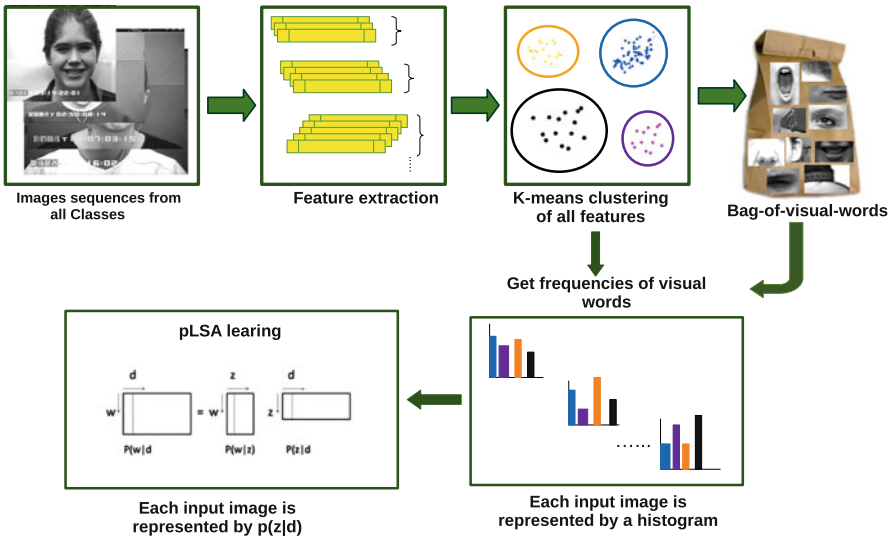



**Fig. 5.5** Database preprocessing approach

of 16 × 16 pixel patches computed over a grid with a spacing of 8 pixels. Second, we have quantized the extracted features into visual words using K-means algorithm [14], and each image and video is then represented as a frequency histogram over the visual words. Finally, we have applied a probabilistic latent semantic analysis (pLSA) to the obtained histograms in order to represent each image by a $d$-dimensional vector where $d$ is the number of latent aspects [6].

### *5.4.3 Online Human Action Recognition*

Recognizing human from videos is a widely studied problem in many applications, both offline and online. The interest of online processing is motivated by the promise of many applications. If we take the example of video surveillance systems for airports, online human recognition plays a key role in protecting against acts of terrorism and in providing real-time surveillance in various airport departments. Another application of online human action processing is in smart environments such as health care and assisted living geared to provide housing facilities for elderly population and people with disabilities. Accordingly, many recent researchers have concentrated on online human action recognition. An approach based on motion data and location information has been adapted in [50] to indoor human daily activity recognition. A combination between neural network

and HMM has been proposed to model motion data and location information. A more recent approach [46] based on semi-supervised learning was proposed to robustly recognize moves online from unsegmented data. In [24], an online activity recognition on smart phones using the built-in accelerometers was proposed to classify the basic movement of the user. This method was performed using the KNN classifier and evaluated by applying Naive Bayes classification method. To deal with the problem of continuous activities and personalized learning, an online multitask learning method for large-scale personalized activity recognition has been introduced in [44]. Using a dataset obtained from real-home settings, the authors in [36] have proposed an evolving fuzzy systems to recognize activities of the daily living (ADLs) from sensor streams.

### *5.4.4 Online Human Facial Expression Recognition*

Recognizing human facial expression is an active research problem in the recent years. Various works have focused on online facial expression recognition that have been used for different applications such as smart environment, video surveillance systems, e-education, and many other interesting utilization. The interest in online human facial expression recognition is motivated by the promise of automatically categorizing the different types of human face expression used in computer interaction, medicine, e-learning, access control, monitoring, and marketing. For example, knowing the client's emotional state, computer can become a more effective interface to detect patient feeling about medical treatment. For instance, interpreting autism's expressions could help in developing a therapy system. In tutoring system, detecting the state of the learner may enhance the presentation style of e-learning program. Another interesting application is to detect drivers' state, helping the driver monitor their stress level and alert other cars. If we take the example of video surveillance systems for ATM, facial expression recognition plays a key role in protecting against acts of terrorism and theft as it doesn't dispense money when someone is scared. Many researches have focused on facial expression recognition but few of them were interested how could we understand the facial expression in an interactive way. A system built on elastic graph matching [21] was proposed to track and detect the face of a person in a live video sequence. In [7], a study on understanding how babies learn to recognize facial expressions is presented. They have used a cognitive system algebra combined with a neural network model to online recognize facial expressions. A method for collecting and analyzing facial responses over the web was introduced in [31]. The proposed framework was utilized to crowdsource over three million face videos in response to thousands of media clips ranging from advertisements to movie trailers to TV shows and political debates.

### 5.4.5 Online Pedestrian Detection in Infrared Images

Infrared (IR) thermography is an imaging method for visualizing radiation not observable by human eye. Analyzing thermal images has occurred growing interest both in research and in industry with a wide area of applications. In military surveillance uses, they have the need to mount infrared detection system on vehicles or towers for border surveillance. Fire-fighters use infrared imaging as a mechanism to find missing people in buildings on fire. As well, car-pedestrians accidents which occur at night acquire the use of far-infrared camera in order to discern the thermal energy. Online detection pedestrians from infrared images has not been much explored yet. In [27], a real-time online learning was proposed to track pedestrians using boosted random ferns and using Weber–Fechner's law to detect pedestrians according to the season and the weather. Another interesting work on online pedestrians detection based on particle filters and combination of a local intensity distribution (LID) with oriented center symmetric local binary patterns was introduced in [23]. The proposed algorithm was applied to various thermal videos to detect the most likely target position in the subsequent frame.

### 5.4.6 Results

In this section, we evaluate our proposed framework in different experiments and we compare recognition rates with methods from literature and offline methods. After preprocessing our databases, we used 15 subjects from each activity from KTH dataset to construct the visual vocabulary and the remaining 10 subjects to test. For facial expression and pedestrian detection experiments, we selected 70% of the data to the training and the remaining for the test.

We start by studying the impact of the visual vocabulary sizes on the recognition accuracy for our online methods onFP-MGGMM, onRA-FP-MGGMM, onFS-MGGMM, and the other approaches (GMM,GGMM), as depicted in Fig. 5.6a for KTH dataset, Fig. 5.7a for Cohn–Kanade and in Fig. 5.8a for OSU thermal dataset. According to these results, the maximum accuracy value is obtained with visual vocabulary sizes of 200 for KTH, 20 for Cohn–Kanade, and 50 for OSU thermal dataset. Moreover, we have studied the impact of the number of aspects on the recognition accuracy as shown in Figs. 5.6b, 5.7b, and 5.8b and we found that the optimal accuracy was obtained when the number of aspects was set to 20 for KTH and 6 for Cohn–Kanade and OSU thermal dataset.

We achieved the best performance with human action recognition, facial expression recognition, and infrared pedestrian detection in different proposed online learning multivariate generalized Gaussian methods. For instance, for human action recognition, the online Riemannian averaged fixed-point multivariate generalized Gaussian mixture achieves the best recognition rates (99.37%) as shown in Table 5.1. In recognizing facial expression, the online fixed-point MGGMM

(96.84%) outperforms the other related works, Gaussian-based models, and also the two proposed online mixture models as indicated in Table 5.2. With respect to infrared pedestrian detection, experiments results on OSU thermal dataset shown in Table 5.3 that the Fisher-scoring MGGMM provides the best performance (96.64%) as compared to other online Gaussian-based models. We notice from those tables (Tables 5.1–5.3) that our three proposed discriminative online learning methods reached superior performance where the accuracy increases approximately by 20% comparing to the basic Gaussian mixture model and the univariate generalized Gaussian mixture. We display also the confusion matrix for the proposed online learning methods for KTH database in Tables 5.4, 5.5, 5.6, for Cohn–Kanade in Tables 5.7, 5.8, 5.9, and for OSU-thermal database in Tables 5.10, 5.11, and 5.12.

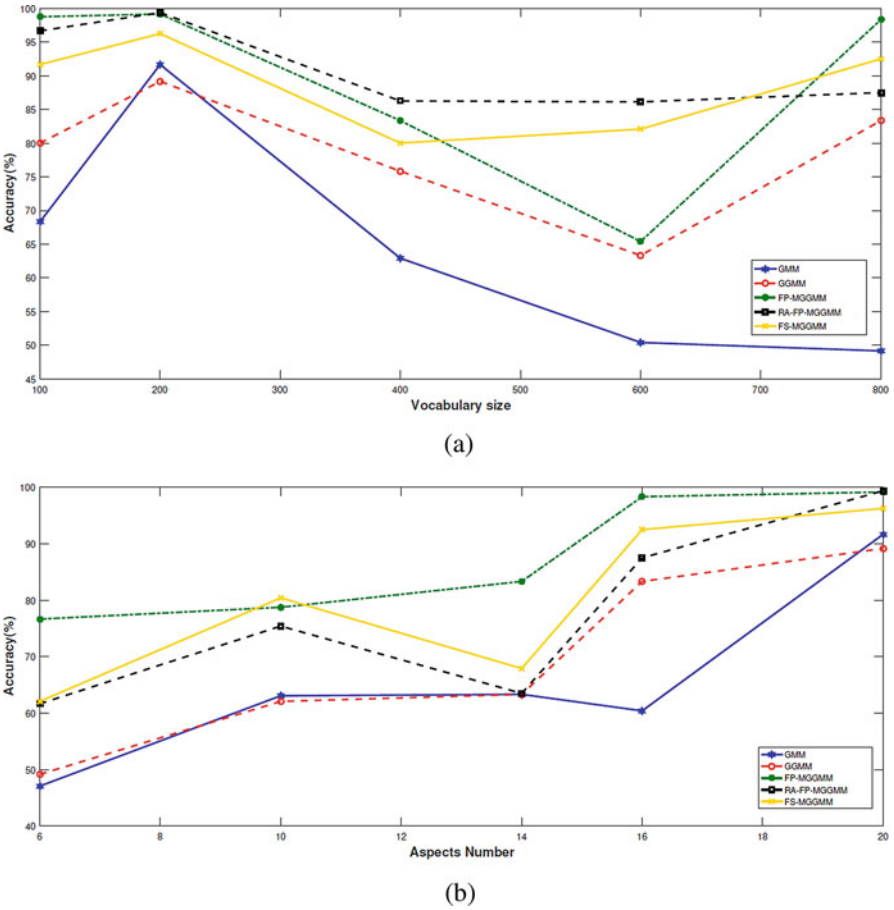
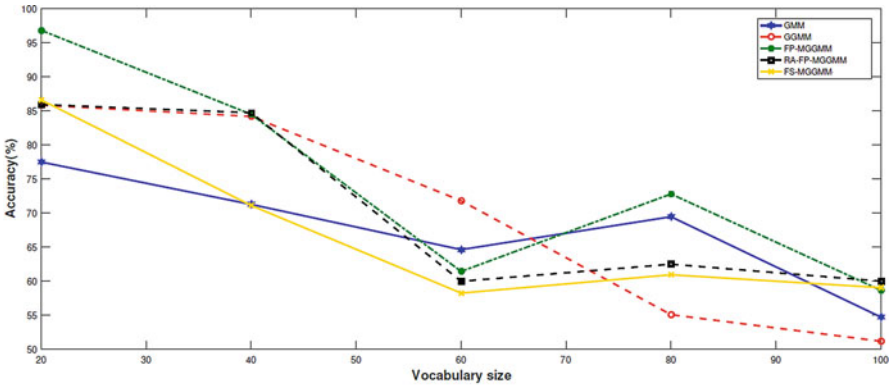
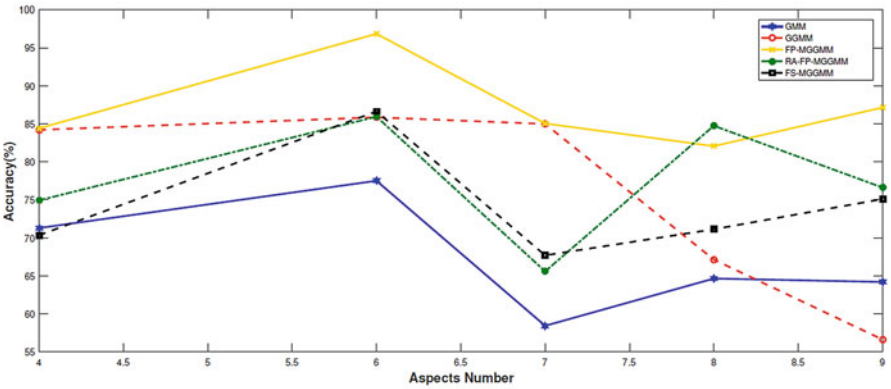



**Fig. 5.6** (**a**) Recognition accuracy vs. vocabulary size for the KTH dataset; (**b**) recognition accuracy vs. the number of aspects for the KTH dataset



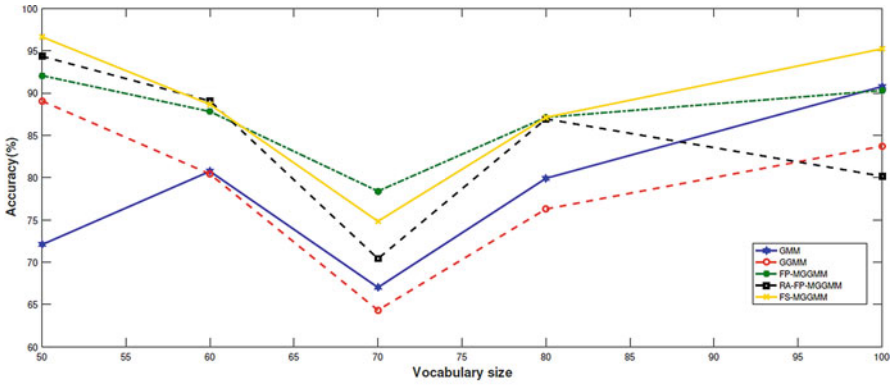

(a)



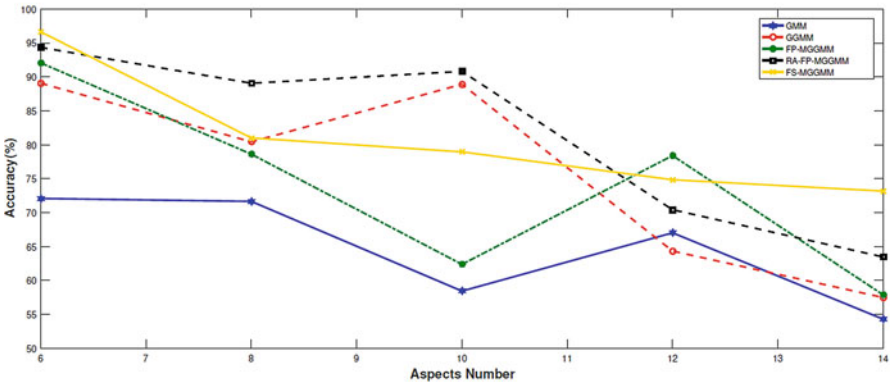

(b)

**Fig. 5.7** (**a**) Recognition accuracy vs. vocabulary size for the Cohn–Kanade dataset. (**b**) Recognition accuracy vs. the number of aspects for the Cohn–Kanade dataset

We compare the performance of the offline learning and online learning proposed in Sect. 5.2.2 on all the three datasets. Tables 5.13, 5.14, and 5.15 illustrate the accuracy and the running time for each proposed model and for the Gaussian mixture model and the generalized Gaussian mixture model. According to those tables, we notice that online learning has improved the quality of the clusters and decreased the time of running compared to the offline learning.




(a)




(b)

**Fig. 5.8** (**a**) Recognition accuracy vs. vocabulary size for the OSU-thermal dataset. (**b**) Recognition accuracy vs. the number of aspects for the OSU-thermal dataset

**Table 5.1** The average recognition rates using different online algorithms for KTH dataset

| Approach | Recognition rates |
|---|---|
| sig min-Hash [20] | 91.2 |
| Baseline [20] | 44.3 |
| OHAC [37] | 82.2 |
| onGMM | 91.66 |
| onGGMM | 89.16 |
| onFP-MGGMM | 99.16 |
| onRA-FP-MGGMM | 99.37 |
| onFS-MGGMM | 96.25 |

**Table 5.2** The average recognition rates using different online algorithms for Cohn–Kanade dataset

| Approach | Recognition rates |
|---|---|
| onBM [17] | 84.57 |
| onDM | 79.69 |
| onGDM | 83.08 |
| onGMM | 77.50 |
| onGGMM | 85.85 |
| onFP-MGGMM | 96.84 |
| onRA-FP-MGGMM | 85.94 |
| onFS-MGGMM | 86.62 |

**Table 5.3** The average recognition rates using different algorithms for OSU-thermal dataset

| Approach | Recognition rates |
|---|---|
| onGMM | 90.76 |
| onGGMM | 89.06 |
| onFP-MGGMM | 92.06 |
| onRA-FP-MGGMM | 94.34 |
| onFS-MGGMM | 96.64 |

**Table 5.4** Confusion matrix for KTH dataset using onFP-MGGMM

| | C1 | C2 | C3 | C4 | C5 | C6 |
|---|---|---|---|---|---|---|
| C1 | 97.5% | 0% | 2.5% | 0% | 0% | 0% |
| C2 | 0% | 100% | 0% | 0% | 0% | 0% |
| C3 | 0% | 0% | 100% | 0% | 0% | 0% |
| C4 | 0% | 0% | 0% | 100% | 0% | 0% |
| C5 | 0% | 0% | 0% | 0% | 100% | 0% |
| C6 | 0% | 0% | 0% | 0% | 2.5% | 97.5% |

**Table 5.5** Confusion matrix for KTH dataset using onRA-FP-MGGMM

| | C1 | C2 | C3 | C4 | C5 | C6 |
|---|---|---|---|---|---|---|
| C1 | 92.5% | 5% | 0% | 0% | 2.5% | 0% |
| C2 | 0% | 100% | 0% | 0% | 0% | 0% |
| C3 | 0% | 0% | 100% | 0% | 0% | 0% |
| C4 | 0% | 0% | 0% | 100% | 0% | 0% |
| C5 | 0% | 0% | 0% | 0% | 100% | 0% |
| C6 | 0% | 0% | 0% | 0% | 0% | 100% |

**Table 5.6** Confusion matrix for KTH dataset using onFS-MGGMM

| | C1 | C2 | C3 | C4 | C5 | C6 |
|---|---|---|---|---|---|---|
| C1 | 90% | 2.5% | 0% | 0% | 7.5% | 0% |
| C2 | 0% | 92.5% | 2.5% | 0% | 0% | 5% |
| C3 | 0% | 0% | 100% | 0% | 0% | 0% |
| C4 | 0% | 2.5% | 0% | 97.5% | 0% | 0% |
| C5 | 0% | 0% | 0% | 2.5% | 97.5% | 0% |
| C6 | 0% | 0% | 0% | 0% | 0% | 100% |

**Table 5.7** Confusion matrix for Cohn–Kanade dataset using onFP-MGGMM

| | C1 | C2 | C3 | C4 | C5 | C6 |
|---|---|---|---|---|---|---|
| C1 | 100% | 0% | 0% | 0% | 0% | 0% |
| C2 | 0% | 91.66% | 8.34% | 0% | 0% | 0% |
| C3 | 0% | 0% | 93.75% | 0% | 6.25% | 0% |
| C4 | 0% | 0% | 0% | 100% | 0% | 0% |
| C5 | 0% | 0% | 0% | 0% | 100% | 0% |
| C6 | 4.35% | 0% | 0% | 0% | 0% | 95.65% |

**Table 5.8** Confusion matrix for Cohn–Kanade dataset using onRA-FP-MGGMM

| | C1 | C2 | C3 | C4 | C5 | C6 |
|---|---|---|---|---|---|---|
| C1 | 86.66% | 0% | 6.6% | 0% | 6.74% | 0% |
| C2 | 0% | 91.66% | 0% | 8.34% | 0% | 0% |
| C3 | 6.25% | 0% | 87.5% | 0% | 6.25% | 0% |
| C4 | 0% | 14.05% | 0% | 80.95% | 0% | 5% |
| C5 | 0% | 0% | 0% | 19.05% | 80.95% | 0% |
| C6 | 0% | 0% | 0% | 17.4% | 0% | 82.60% |

**Table 5.9** Confusion matrix for Cohn–Kanade dataset using onFS-MGGMM

| | C1 | C2 | C3 | C4 | C5 | C6 |
|---|---|---|---|---|---|---|
| C1 | 100% | 0% | 0% | 0% | 0% | 0% |
| C2 | 8.34% | 91.66% | 0% | 0% | 0% | 0% |
| C3 | 6.25% | 0% | 87.5% | 0% | 0% | 6.25% |
| C4 | 3.4% | 0% | 11.13% | 66.66% | 18.81% | 0% |
| C5 | 0% | 9.53% | 0% | 0% | 90.47% | 0% |
| C6 | 8.7% | 0% | 0% | 8.7% | 0% | 82.60% |

**Table 5.10** Confusion matrix for OSU-thermal dataset using onFP-MGGMM

| | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
|---|---|---|---|---|---|---|---|
| C1 | 94.73% | 0% | 5.27% | 0% | 0% | 0% | 0% |
| C2 | 0% | 86.66% | 0% | 13.34% | 0% | 0% | 0% |
| C3 | 9.1% | 0% | 90.90% | 0% | 0% | 0% | 0% |
| C4 | 0% | 7.69% | 0% | 84.61% | 0% | 7.7% | 0% |
| C5 | 11.38% | 0% | 11.7% | 0% | 76.92% | 0% | 0% |
| C6 | 0% | 36.37% | 0% | 0% | 0% | 63.63% | 0% |
| C7 | 0% | 0% | 0% | 16.67% | 0% | 0% | 83.33% |

**Table 5.11** Confusion matrix for OSU-thermal dataset using onRA-FP-MGGMM

| | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
|---|---|---|---|---|---|---|---|
| C1 | 94.73% | 0% | 5.27% | 0% | 0% | 0% | 0% |
| C2 | 13.3% | 80% | 0% | 6.7% | 0% | 0% | 0% |
| C3 | 0% | 0% | 90.90% | 0% | 9.1% | 0% | 0% |
| C4 | 0% | 7.7% | 0% | 92.30% | 0% | 0% | 0% |
| C5 | 0% | 0% | 23.08% | 0% | 76.92% | 0% | 0% |
| C6 | 0% | 36.37% | 0% | 0% | 0% | 63.63% | 0% |
| C7 | 0% | 0% | 0% | 8.34% | 0% | 0% | 91.66% |

**Table 5.12** Confusion matrix for OSU-thermal dataset using onFS-MGGMM

| | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
|---|---|---|---|---|---|---|---|
| C1 | 92.10% | 0% | 0% | 7.9% | 0% | 0% | 0% |
| C2 | 0% | 76.66% | 0% | 11.6% | 0% | 11.74% | 0% |
| C3 | 0% | 0% | 90.90% | 0% | 0% | 9.1% | 0% |
| C4 | 11.54% | 0% | 0% | 88.46% | 0% | 0% | 0% |
| C5 | 0% | 19.24% | 0% | 0% | 80.76% | 0% | 0% |
| C6 | 0% | 0% | 0% | 0% | 13.64% | 86.36% | 0% |
| C7 | 0% | 16.67% | 0% | 0% | 0% | 0% | 83.33% |

**Table 5.13** Performance of human recognition using offline and online algorithms for KTH dataset

| | Offline | | Online | |
|---|---|---|---|---|
| | Accuracy (%) | Run time (s) | Accuracy (%) | Run time (s) |
| GMM | 87.14 | 27.78 | 91.66 | 8.63 |
| GGMM | 88.33 | 38.36 | 89.16 | 3.39 |
| FP-MGGMM | 91.97 | 19.18 | 99.16 | 10.29 |
| RA-FP-MGGMM | 94.16 | 22.71 | 99.37 | 10.44 |
| FS-MGGMM | 94.58 | 15.87 | 96.25 | 10.03 |

**Table 5.14** Performance of facial expression recognition using offline and online algorithms for Cohn–Kanade dataset

| | Offline | | Online | |
|---|---|---|---|---|
| | Accuracy (%) | Run time (s) | Accuracy (%) | Run time (s) |
| GMM | 76.46 | 14.74 | 77.50 | 2.31 |
| GGMM | 85.85 | 9.55 | 85.85 | 1.36 |
| FP-MGGMM | 88.98 | 7.1 | 96.84 | 2.35 |
| RA-FP-MGGMM | 91.61 | 8.76 | 85.94 | 2.61 |
| FS-MGGMM | 93.87 | 3.8 | 86.62 | 2.24 |

**Table 5.15** The average recognition rates using offline and online algorithms for OSU-thermal dataset

| | Offline | | Online | |
|---|---|---|---|---|
| | Accuracy (%) | Run time (s) | Accuracy (%) | Run time (s) |
| GMM | 79.61 | 15.17 | 90.76 | 1.78 |
| GGMM | 84.26 | 8.56 | 89.06 | 1.05 |
| FP-MGGMM | 89.32 | 6.65 | 92.06 | 1.7 |
| RA-FP-MGGMM | 92.51 | 6.75 | 94.34 | 2.1 |
| FS-MGGMM | 89.81 | 6.21 | 96.64 | 2.02 |

## 5.5 Conclusion

In this paper, we have proposed an online learning based on deterministic framework that is able to estimate the multivariate parameters of the generalized Gaussian mixture model. To this aim, we were motivated by developing a robust maximum likelihood approach based on recent techniques, namely fixed-point, Riemannian-averaged fixed-point, and Fisher scoring and a stochastic gradient descent algorithm. We applied our algorithms on extensive experiments including challenging applications namely recognizing human action, facial expression and detecting pedestrian in infrared images. Comparisons revealed that our online methods achieve better recognition rates with respect to other offline methods, proposed methods in literature, and other Gaussian-based models. In spite of the promising results achieved, further enhancement could be done using online variational learning.

## References

1. Allili, M.S., Bouguila, N., Ziou, D.: Online video foreground segmentation using general gaussian mixture modeling. In: 2007 IEEE International Conference on Signal Processing and Communications, pp. 959–962. IEEE, New York (2007)
2. Amayri, O., Bouguila, N.: Online spam filtering using support vector machines. In: 2009 IEEE Symposium on Computers and Communications, pp. 337–340. IEEE, New York (2009)
3. Amayri, O., Bouguila, N.: Online news topic detection and tracking via localized feature selection. In: The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE, New York (2013)
4. Bdiri, T., Bouguila, N., Ziou, D.: A statistical framework for online learning using adjustable model selection criteria. Eng. Appl. Artif. Intell. **49**, 19–42 (2016)
5. Beringer, J., Hüllermeier, E.: Online clustering of parallel data streams. Data Knowl. Eng. **58**(2), 180–204 (2006)
6. Bosch, A., Zisserman, A., Muñoz, X.: Scene classification via pLSA. In: Computer Vision–ECCV 2006, pp. 517–530 (2006)
7. Boucenna, S., Gaussier, P., Andry, P., Hafemeister, L.: Imitation as a communication tool for online facial expression learning and recognition. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5323–5328. IEEE, New York (2010)

8. Bouguila, N., Ziou, D.: Online clustering via finite mixtures of Dirichlet and minimum message length. Eng. Appl. Artif. Intell. **19**(4), 371–379 (2006)
9. Boukouvalas, Z., Fu, G.S., Adalı, T.: An efficient multivariate generalized gaussian distribution estimator: application to IVA. In: 2015 49th Annual Conference on Information Sciences and Systems (CISS), pp. 1–4. IEEE, New York (2015)
10. Boukouvalas, Z., Said, S., Bombrun, L., Berthoumieu, Y., Adalı, T.: A new Riemannian averaged fixed-point algorithm for MGGD parameter estimation. IEEE Signal Process. Lett. **22**(12), 2314–2318 (2015)
11. Channoufi, I., Bourouis, S., Bouguila, N., Hamrouni, K.: Color image segmentation with bounded generalized gaussian mixture model and feature selection. In: 4th International Conference on Advanced Technologies for Signal and Image Processing, ATSIP 2018, Sousse, March 21–24, 2018, pp. 1–6 (2018)
12. Channoufi, I., Bourouis, S., Bouguila, N., Hamrouni, K.: Image and video denoising by combining unsupervised bounded generalized gaussian mixture modeling and spatial information. Multimedia Tools Appl. **77**(19), 25591–25606 (2018)
13. Channoufi, I., Bourouis, S., Bouguila, N., Hamrouni, K.: Spatially constrained mixture model with feature selection for image and video segmentation. In: Image and Signal Processing - 8th International Conference, ICISP 2018, Proceedings, Cherbourg, July 2–4, 2018, pp. 36–44 (2018)
14. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, Prague, vol. 1, pp. 1–2 (2004)
15. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B (Methodol.) **39**(1), 1–38 (1977)
16. Elguebaly, T., Bouguila, N.: Semantic scene classification with generalized gaussian mixture models. In: International Conference Image Analysis and Recognition, pp. 159–166. Springer, New York (2015)
17. Fan, W., Bouguila, N.: Online facial expression recognition based on finite Beta-Liouville mixture models. In: 2013 International Conference on Computer and Robot Vision, pp. 37–44. IEEE, New York (2013)
18. Fan, W., Bouguila, N.: Online variational learning of generalized Dirichlet mixture models with feature selection. Neurocomputing **126**, 166–179 (2014)
19. Fan, W., Sallay, H., Bouguila, N., Bourouis, S.: A hierarchical Dirichlet process mixture of generalized Dirichlet distributions for feature selection. Comput. Electr. Eng. **43**, 48–65 (2015)
20. Gilbert, A., Bowden, R.: Image and video mining through online learning. Comput. Vis. Image Underst. **158**, 72–84 (2017)
21. Hong, H., Neven, H., Von der Malsburg, C.: Online facial expression recognition based on personalized galleries. In: Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition, pp. 354–359. IEEE, New York (1998)
22. Kanade, T., Tian, Y., Cohn, J.F.: Comprehensive database for facial expression analysis. In: Conference Paper, Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), p. 46. IEEE, New York (2000)
23. Ko, B.C., Kwak, J.Y., Nam, J.Y.: Human tracking in thermal images using adaptive particle filters with online random forest learning. Opt. Eng. **52**(11), 113105 (2013)
24. Kose, M., Incel, O.D., Ersoy, C.: Online human activity recognition on smart phones. In: Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data, vol. 16, pp. 11–15 (2012)
25. Kotz, S.: Multivariate distributions at a cross-road. In: Statistical Distributions in Scientific Work, vol. 1, pp. 247–270 (1975)
26. Kumar, K.N., Rao, K.S., Srinivas, Y., Satyanarayana, C.: Studies on texture segmentation using D-dimensional generalized gaussian distribution integrated with hierarchical clustering. Int. J. Image Graph. Signal Process. **8**(3), 45–54 (2016)

27. Kwak, J.Y., Ko, B.C., Nam, J.Y.: Pedestrian tracking using online boosted random ferns learning in far-infrared imagery for safe driving at night. IEEE Trans. Intell. Transp. Syst. **18**(1), 69–81 (2017)
28. Laptev, I.: On space-time interest points. Int. J. Comput. Vis. **64**(2–3), 107–123 (2005)
29. Lee, D.S.: Effective gaussian mixture learning for video background subtraction. IEEE Trans. Patt. Anal. Mach. Intell. **27**(5), 827–832 (2005)
30. Lowe, D.G., et al.: Object recognition from local scale-invariant features. In: ICCV, vol. 99, pp. 1150–1157 (1999)
31. McDuff, D., Kaliouby, R., Picard, R.: Crowdsourcing facial responses to online videos: extended abstract. In: Proceedings of the 2015 International Conference on Affective Computing and Intelligent Interaction (ACII) (Xi'an), pp. 512–518 (2015)
32. Naiar, F., Bourouis, S., Bouguila, N., Belghith, S.: A fixed-point estimation algorithm for learning the multivariate GGMM: application to human action recognition. In: 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE), pp. 1–4. IEEE, New York (2018)
33. Najar, F., Bourouis, S., Bouguila, N., Belghith, S.: A comparison between different gaussian-based mixture models. In: 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), pp. 704–708. IEEE, New York (2017)
34. Najar, F., Bourouis, S., Zaguia, A., Bouguila, N., Belghith, S.: Unsupervised human action categorization using a Riemannian averaged fixed-point learning of multivariate GGMM. In: International Conference Image Analysis and Recognition, pp. 408–415. Springer, New York (2018)
35. Najar, F., Bourouis, S., Bouguila, N., Belghith, S.: Unsupervised learning of finite full covariance multivariate generalized gaussian mixture models for human activity recognition. Multimed. Tools Appl. **1**, 1–23 (2019)
36. Ordóñez, F.J., Iglesias, J.A., De Toledo, P., Ledezma, A., Sanchis, A.: Online activity recognition using evolving classifiers. Expert Syst. Appl. **40**(4), 1248–1255 (2013)
37. Panzner, M., Beyer, O., Cimiano, P.: Human activity classification with online growing neural gas. In: Workshop on New Challenges in Neural Computation (NC2) (2013)
38. Pascal, F., Bombrun, L., Tourneret, J.Y., Berthoumieu, Y.: Parameter estimation for multivariate generalized gaussian distributions. IEEE Trans. Signal Process. **61**(23), 5960–5971 (2013)
39. Perkins, S., Theiler, J.: Online feature selection using grafting. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03), pp. 592–599 (2003)
40. Sailaja, V., Srinivasa Rao, K., Reddy, K.: Text independent speaker identification with finite multivariate generalized gaussian mixture model and hierarchical clustering algorithm. Int. J. Comput. Appl. **11**(11), 0975–8887 (2010)
41. Sato, M.A., Ishii, S.: On-line EM algorithm for the normalized gaussian network. Neural Comput. **12**(2), 407–432 (2000)
42. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, vol. 3, pp. 32–36. IEEE, New York (2004)
43. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of the 15th ACM International Conference on Multimedia, pp. 357–360. ACM, New York (2007)
44. Sun, X., Kashima, H., Ueda, N.: Large-scale personalized human activity recognition using online multitask learning. IEEE Trans. Knowl. Data Eng. **25**(11), 2551–2563 (2013)
45. Verdoolaege, G., Scheunders, P.: On the geometry of multivariate generalized gaussian models. J. Math. Imag. Vis. **43**(3), 180–193 (2012)
46. Vieira, T., Faugeroux, R., Martínez, D., Lewiner, T.: Online human moves recognition through discriminative key poses and speed-aware action graphs. Mach. Vis. Appl. **28**(1–2), 185–200 (2017)
47. Yao, J.F.: On recursive estimation in incomplete data models. Statistics **34**(1), 27–51 (2000)

48. Zhang, J., Ghahramani, Z., Yang, Y.: A probabilistic model for online document clustering with application to novelty detection. In: Advances in Neural Information Processing Systems, pp. 1617–1624 (2005)
49. Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans. Patt. Anal. Mach. Intell. **29**(6), 915–928 (2007)
50. Zhu, C., Sheng, W.: Motion-and location-based online human daily activity recognition. Pervasive Mob. Comput. **7**(2), 256–269 (2011)
51. Zivkovic, Z., van der Heijden, F.: Recursive unsupervised learning of finite mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **26**(5), 651–656 (2004)

# Part III
# Spherical and Count Data Clustering

# Chapter 6
# $L_2$ Normalized Data Clustering Through the Dirichlet Process Mixture Model of von Mises Distributions with Localized Feature Selection


**Wentao Fan, Nizar Bouguila, Yewang Chen, and Ziyi Chen**

**Abstract** In this chapter, we propose a probabilistic model based-approach for clustering $L_2$ normalized data. Our approach is based on the Dirichlet process mixture model of von Mises (VM) distributions. Since it assumes an infinite number of clusters (i.e., the mixture components), the Dirichlet process mixture model of VM distributions can also be considered as the infinite VM mixture model. Comparing with finite mixture model in which the number of mixture components have to be determined through extra efforts, the infinite mixture VM model is a nonparametric model such that the number of mixture components is assumed to be infinite initially and will be inferred automatically during the learning process. To improve clustering performance for high-dimensional data, a localized feature selection scheme is integrated into the infinite VM mixture model which can effectively detect irrelevant features based on the estimated feature saliencies. In order to learn the proposed infinite mixture model with localized feature selection, we develop an effective approach using variational inference that can estimate model parameters and feature saliencies with closed-form solutions. Our model-based clustering approach is validated through two challenging applications, namely topic novelty detection and unsupervised image categorization.

## 6.1 Introduction

During the last two decades, finite mixture models have shown their effectiveness in tackling unsupervised learning problems, such as clustering, for both univariate

W. Fan (✉) · Y. Chen · Z. Chen
Department of Computer Science and Technology, Huaqiao University, Xiamen, China
e-mail: fwt@hqu.edu.cn; ywchen@hqu.edu.cn; chenziyihq@hqu.edu.cn

N. Bouguila
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca



N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6_6

and multivariate data. The target of clustering is to automatically partition a data set into different groups such that data of the same group are as similar as possible and data of different groups are as different as possible. Finite mixture models have demonstrated promising performance for clustering and have been widely applied in many computer vision and pattern recognition applications [1].

In clustering approaches based on finite mixture models, choosing an appropriate basic distribution that best describes the given data is a crucial problem. Although the Gaussian distribution has been a popular choice for constructing mixture models, other distributions may provide better performance for modeling other types of data. For instance, according to recent studies, the Dirichlet [2], the generalized Dirichlet [3, 4], and the Beta-Liouville distributions [5, 6] are better alternatives than Gaussian for modeling proportional data vectors (i.e., normalized histograms). Recently, several works [7–10] have shown that $L_2$ normalized data are very important and can be found in various practical applications particularly in the fields of text, image, and video classification. As discussed in [7], $L_2$ normalized feature vectors can be naturally modeled using spherical distributions such as the von Mises (VM) distribution, since they can be visualized as points on a hypersphere after the normalization.

In this chapter, we propose a probabilistic model based-approach for clustering $L_2$ normalized data. Our approach is based on VM distributions with a nonparametric framework known as the Dirichlet process mixture model [11–13]. Since the Dirichlet process mixture model assumes an infinite number of clusters (i.e., the mixture components), the Dirichlet process mixture model of VM distributions can also be considered as the infinite VM mixture model. Comparing with finite mixture model in which the number of mixture components have to be determined through extra efforts, the infinite mixture VM model is a nonparametric model such that the number of mixture components is assumed to be infinite initially and will be inferred automatically during the learning process. To improve clustering performance for high-dimensional data, a localized feature selection scheme [14] is integrated into the infinite VM mixture model which can effectively detect irrelevant features based on the estimated feature saliencies. In order to learn the proposed infinite mixture model with localized feature selection, we develop an effective approach using variational inference [15–17] that can estimate model parameters and feature saliencies with closed-form solutions. Our model-based clustering approach is validated through two challenging applications, namely topic novelty detection and unsupervised image categorization.

The remaining part of this chapter can be listed as follows. In Sect. 6.2, we introduce the Dirichlet process mixture of VM distributions with localized feature selection. In Sect. 6.3, we develop a learning algorithm based on variational inference to estimate the parameters of our model. In Sect. 6.4, we provide experimental results of our model on topic novelty detection and unsupervised image categorization. Finally, conclusion is given in Sect. 6.5.

## 6.2 The Dirichlet Process Mixture of VM Distributions with Localized Feature Selection

### 6.2.1 *Finite VM Mixture Model with Localized Feature Selection*

If we have a data set $\mathcal{X} = \mathbf{X}_1, \ldots, \mathbf{X}_N$ that contains $N$ data points, where each $\mathbf{X}_i$ is a $L_2$-normalized $D$-dimensional feature vector (i.e., $(\mathbf{X}_i)^T \mathbf{X}_i = 1$), then each $\mathbf{X}_i$ can be modeled using a von Mises (VM) distribution $\mathcal{V}(\cdot)$ [7, 10] with the assumption that features of $\mathbf{x}_i$ are independent

$$p(\mathbf{X}_i|\boldsymbol{\mu}, \boldsymbol{\lambda}) = \prod_{d=1}^{D} \mathcal{V}(\mathbf{Y}_{id}|\boldsymbol{\mu}_d, \lambda_d) = \prod_{d=1}^{D} \frac{1}{2\pi I_0(\lambda_d)} e^{\lambda_d \boldsymbol{\mu}_d^{\mathrm{T}} \mathbf{Y}_{id}} \tag{6.1}$$

where $Y_{id1} = X_{id}$ and $\mathbf{Y}_{id} = (Y_{id1}, Y_{id2})$. $Y_{id2}$ is used to ensure that the vector $\mathbf{Y}_{id}$ is a spherical (i.e., $L_2$ normalized) vector. $\boldsymbol{\mu}_d = (\mu_{d1}, \mu_{d2})$ and $\lambda_d \geq 0$ are the mean direction and the concentration parameter, respectively, for the VM distribution. $I_0(\cdot)$ denotes the modified Bessel function of the first kind of order 0.

If the vector $\mathbf{X}_i$ is distributed according to a VM mixture model with $M$ components, then we have

$$p(\mathbf{X}_i|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = \sum_{j=1}^{M} \pi_j \prod_{d=1}^{D} \mathcal{V}(\mathbf{y}_{id}|\boldsymbol{\mu}_{jd}, \lambda_{jd}) \tag{6.2}$$

where the vector $\boldsymbol{\pi} = \pi_1, \ldots, \pi_M$ denotes mixing coefficients with the constraints that $\sum_{j=1}^{M} \pi_j = 1$ and $0 \leq \pi_j \leq 1$.

In mixture modeling, it is often convenient for a latent variable to indicate the membership assignment. As in our case, each vector $\mathbf{X}_i$ is assigned with a latent variable $\mathbf{z}_i$ as the membership indicator variable, such that $z_{ij} = 1$ if $\mathbf{X}_i$ is drawn from the $j$th component, otherwise it equals 0. Then, the conditional distribution of the observed data vectors $\mathcal{X}$, given the latent variables and the component parameters

$$p(\mathcal{X}|Z, \boldsymbol{\mu}, \boldsymbol{\lambda}) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left( \prod_{d=1}^{D} \mathcal{V}(\mathbf{y}_{id}|\boldsymbol{\mu}_{jd}, \lambda_{jd}) \right)^{z_{ij}} \tag{6.3}$$

where $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$. The probability distribution of the latent variable $Z$ is given by

$$p(\mathbf{Z}|\boldsymbol{\pi}) = \prod_{i=1}^{N} \prod_{j=1}^{M} \pi_j^{z_{ij}} \tag{6.4}$$

As we may notice that in Eq. (6.3), the finite VM mixture model considers all features are equally important with equal contribution to the clustering process. However, this assumption is not realistic in practice, since high-dimensional data may contain noisy features or features that are irrelevant to the clustering analysis. To improve the clustering performance, we adopt localized features selection which has shown promising results in unsupervised clustering in previous research works [14, 18, 19]. In contrast to performing features selection in a *global* way (i.e., feature saliencies are the same for all clusters) [20], localized feature selection assumes that the importance of a feature is different for different clusters. Motivated by its better performance, we integrate localized feature selection into our VM mixture model. Then, the probability distribution of each feature $X_{id}$ in our model can be defined by

$$p(X_{id}) = \mathcal{V}(\mathbf{Y}_{id}|\boldsymbol{\mu}_{jd}, \lambda_{jd})^{\phi_{ijd}} \mathcal{V}(\mathbf{Y}_{id}|\boldsymbol{\mu}'_{jd}, \lambda'_{jd})^{1-\phi_{ijd}} \tag{6.5}$$

where the binary variable $\phi_{ijd}$ denotes the feature relevance, such that $\phi_{ijd} = 0$ if the $d$th feature of $\mathbf{X}_i$ in the $j$th component is irrelevant and follows a VM distribution parameterized by $\boldsymbol{\mu}'_{jd}$ and $\lambda'_{jd}$

Then, for the data set $\mathcal{X}$, the VM mixture model with localized feature selection is given by

$$p(\mathcal{X}|\Theta) = \prod_{i=1}^{N} \prod_{j=1}^{M} \prod_{d=1}^{D} \left( \mathcal{V}(\mathbf{Y}_{id}|\theta_{jd})^{\phi_{ijd}} \mathcal{V}(\mathbf{Y}_{id}|\theta'_{jd})^{1-\phi_{ijd}} \right)^{z_{ij}} \tag{6.6}$$

where $\Theta = \{\mathbf{Z}, \boldsymbol{\theta}, \boldsymbol{\theta}', \boldsymbol{\phi}\}$, $\theta_{jd} = (\boldsymbol{\mu}_{jd}, \lambda_{jd})$ and $\theta'_{jd} = (\boldsymbol{\mu}'_{jd}, \lambda'_{jd})$.

### 6.2.2 *Infinite VM Mixture Model with Localized Feature Selection*

Although finite mixture models are effective for clustering, the determination of the number of components is a crucial problem in mixture modeling and often requires extra efforts to handle. An elegant solution to this problem is to assume that the number of mixture components is infinite and will be determined automatically during the learning process. A common way to extend a finite mixture model into its infinite counterpart is through a Bayesian nonparametric framework known as the Dirichlet process mixture model [11, 12].

In our work, we extend finite VM mixture model with localized feature selection into the infinite VM mixture model by assuming that the mixing coefficients $\boldsymbol{\pi}$ follows the Dirichlet process mixture model with a stick-breaking representation [13, 21] as defined by

$$\pi_j = \pi'_j \prod_{k=1}^{j-1}(1-\pi'_k), \qquad G = \sum_{j=1}^{\infty} \pi_j \delta_{\theta_j},$$
$$\pi'_j \sim \text{Beta}(1, \xi), \qquad \theta_j \sim H, \tag{6.7}$$

where $\delta_{\theta_j}$ represents the Dirac delta measure centered at $\theta_j$, $G$ is a random distribution that follows a Dirichlet process $G \sim \text{DP}(\xi, H)$, and $\xi$ is a positive real number. The variables $\{\pi_j\}$ denote the mixing weights where $\sum_{j=1}^{\infty} \pi_j = 1$.

Consequently, the infinite VM mixture model is given by

$$p(\mathbf{X}_i|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = \sum_{j=1}^{\infty} \pi_j \prod_{d=1}^{D} \mathcal{V}(\mathbf{y}_{id}|\boldsymbol{\mu}_{jd}, \lambda_{jd}) \tag{6.8}$$

By including the latent variable $\mathbf{Z}$ and the localized feature selection, we then have

$$p(\mathcal{X}|\Theta) = \prod_{i=1}^{N} \prod_{j=1}^{\infty} \prod_{d=1}^{D} \left( \mathcal{V}(\mathbf{Y}_{id}|\theta_{jd})^{\phi_{ijd}} \mathcal{V}(\mathbf{Y}_{id}|\theta'_{jd})^{1-\phi_{ijd}} \right)^{z_{ij}} \tag{6.9}$$

where the probability distribution of $\mathbf{Z}$ can be redefined based on the stick-breaking representation as

$$p(\mathbf{Z}|\boldsymbol{\pi}') = \prod_{i=1}^{N} \prod_{j=1}^{\infty} \left[ \pi'_j \prod_{k=1}^{j-1}(1-\pi'_k) \right]^{z_{ij}} \tag{6.10}$$

The prior of $\boldsymbol{\pi}'$ follows a Beta distribution as shown in Eq. (6.7)

$$p(\boldsymbol{\pi}') = \prod_{j=1}^{\infty} \text{Beta}(1, \xi_j) \tag{6.11}$$

For the feature relevance parameter $\boldsymbol{\phi}$, a Bernoulli distribution is introduced as its prior

$$p(\boldsymbol{\phi}|\boldsymbol{\epsilon}) = \prod_{i=1}^{N} \prod_{j=1}^{\infty} \prod_{d=1}^{D} \epsilon_{jd}^{\phi_{ijd}} (1-\epsilon_{jd})^{1-\phi_{ijd}} \tag{6.12}$$

where $\boldsymbol{\epsilon} = (\epsilon_{j1}, \ldots, \epsilon_{jD})$ denotes the probabilities that the features are relevant (also known as features saliencies) in the $j$th component.

Then, a von Mises-gamma prior is used as the prior for the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ of the von Mises components that account for "useful" features

$$p(\boldsymbol{\mu}, \boldsymbol{\lambda}) = \prod_{j=1}^{\infty} \prod_{d=1}^{D} p(\mu_{jd}|\lambda_{jd}) p(\lambda_{jd})$$
$$= \prod_{j=1}^{\infty} \prod_{d=1}^{D} \mathcal{V}(\mu_{jd}|\mathbf{m}_{jd}, \beta_{jd}\lambda_{jd}) \mathcal{G}(\lambda_{jd}|u_{jd}, v_{jd}) \tag{6.13}$$

where $\mathbf{m}_{jd} = (m_{jd1}, m_{jd2})$.

Similarly, a von Mises-gamma prior is adopted for parameters $\boldsymbol{\mu}'$ and $\boldsymbol{\lambda}'$ that account for "noisy" features

$$p(\boldsymbol{\mu}', \boldsymbol{\lambda}') = \prod_{j=1}^{D} \prod_{d=1}^{D} \mathcal{V}(\mu'_{jd}|\mathbf{m}'_{jd}, \beta'_{jd}\lambda'_{jd}) \mathcal{G}(\lambda'_{jd}|u'_{jd}, v'_{jd}) \tag{6.14}$$

where $\mathbf{m}'_{jd} = (m'_{jd1}, m'_{jd2})$.

## 6.3 Model Learning via Variational Inference

In this section, a variational inference (VI) algorithm [15, 16] is developed to learn the proposed infinite von Mises mixture model with localized feature selection. The VI algorithm is a deterministic approximation framework with the goal to find an approximation to the true posterior distribution by maximizing the lower bound on the logarithm of the model evidence. By applying Jensen's inequality, the lower bound $\mathcal{L}$ of the logarithm of $p(\mathcal{X})$ can be found as

$$\ln p(\mathcal{X}) \geq \int q(\Theta) \ln \frac{p(\mathcal{X}, \Theta)}{q(\Theta)} d\Theta = \mathcal{L}(q) \tag{6.15}$$

where $q(\Theta)$ is an approximation to the true posterior distribution $p(\Theta|\mathcal{X})$.

In this work, the stick-breaking representation for the infinite VM mixture model is truncated at a level of $M$

$$\pi'_M = 1, \quad \sum_{j=1}^{M} \pi_j = 1, \quad \pi_j = 0 \ \text{ when } \ j > M \tag{6.16}$$

where $M$ is a variational parameter and will be estimated by the SVI algorithm.

Then, by applying the *structured mean-field* approximation [17], the variational posterior $q(\Lambda)$ can be factorized into the product of variational distributions as

$$q(\Lambda) = q(\mathbf{Z}) q(\boldsymbol{\phi}) q(\boldsymbol{\pi}') q(\boldsymbol{\mu}, \boldsymbol{\lambda}) q(\boldsymbol{\mu}', \boldsymbol{\lambda}') \tag{6.17}$$

By maximizing the variational lower bound $\mathcal{L}(q)$ with respect to each variational variable, we can update the variational posteriors as

$$q(\mathbf{Z}) = \prod_{i=1}^{N}\prod_{j=1}^{M} r_{ij}^{z_{ij}} \tag{6.18}$$

$$q(\boldsymbol{\phi}) = \prod_{i=1}^{N}\prod_{j=1}^{M}\prod_{d=1}^{D} f_{ijd}^{\phi_{ijd}}(1-f_{ijd})^{(1-\phi_{ijd})}, \tag{6.19}$$

$$q(\boldsymbol{\mu},\boldsymbol{\lambda}) = \prod_{j=1}^{M}\prod_{d=1}^{D} \mathcal{V}(\boldsymbol{\mu}_{jd}|\mathbf{m}^*_{jd},\beta^*_{jd}\lambda_{jd})\mathcal{G}(\lambda_{jd}|u^*_{jd},v^*_{jd}) \tag{6.20}$$

$$q(\boldsymbol{\mu}',\boldsymbol{\lambda}') = \prod_{j=1}^{M}\prod_{d=1}^{D} \mathcal{V}(\boldsymbol{\mu}'_{jd}|\mathbf{m}'^*_{jd},\beta'^*_{jd}\lambda'_{jd})\mathcal{G}(\lambda'_{jd}|u'^*_{jd},v'^*_{jd}) \tag{6.21}$$

$$q(\boldsymbol{\pi}') = \prod_{j=1}^{M} \text{Beta}(\pi'_j|g_j,h_j) \tag{6.22}$$

where the associated hyperparameters are defined by

$$r_{ij} = \frac{\exp(\tilde{r}_{ij})}{\sum_{k=1}^{M}\exp(\tilde{r}_{ik})} \tag{6.23}$$

$$\begin{aligned}\tilde{r}_{ij} = &\sum_{d=1}^{D}\langle\phi_{ijd}\rangle\left[\langle\lambda_{jd}\boldsymbol{\mu}_{jd}^{\mathrm{T}}\mathbf{Y}_{id}\rangle - \left(\frac{\partial}{\partial\lambda_{jd}}\ln I_0(\bar{\lambda}_{jd})\right)(\langle\lambda_{jd}\rangle-\bar{\lambda}_{jd}) - \ln I_0(\bar{\lambda}_{jd})\right]\\ &+\sum_{d=1}^{D}\langle 1-\phi_{ijd}\rangle\left[\langle\lambda'_{jd}\boldsymbol{\mu}'^{\mathrm{T}}_{jd}\mathbf{Y}_{id}\rangle\right.\\ &\left.-\left(\frac{\partial}{\partial\lambda'_{jd}}\ln I_0(\bar{\lambda}'_{jd})\right)(\langle\lambda'_{jd}\rangle-\bar{\lambda}'_{jd}) - \ln I_0(\bar{\lambda}'_{jd})\right]\\ &+\langle\ln\pi'_j\rangle + \sum_{k=1}^{j-1}\langle\ln(1-\pi'_k)\rangle\end{aligned} \tag{6.24}$$

$$f_{ijd} = \frac{\exp(\widetilde{f}_{ijd})}{\exp(\widetilde{f}_{ijd})+\exp(\widehat{f}_{ijd})} \tag{6.25}$$

$$\widetilde{f}_{ijd} = \langle z_{ij} \rangle \Bigg[ \langle \lambda_{jd} \boldsymbol{\mu}_{jd}^{\mathrm{T}} \mathbf{Y}_{id} \rangle - \Big( \frac{\partial}{\partial \lambda_{jd}} \ln I_0(\bar{\lambda}_{jd}) \Big) \times (\langle \lambda_{jd} \rangle - \bar{\lambda}_{jd}^{(t-1)}) - \ln I_0(\bar{\lambda}_{jd}) \Bigg] + (\ln \epsilon_{jd}) \tag{6.26}$$

$$\widehat{f}_{ijd} = \langle z_{ij} \rangle \Bigg[ \langle \lambda'_{jd} \boldsymbol{\mu}'^{\mathrm{T}}_{jd} \mathbf{Y}_{id} \rangle - \Big( \frac{\partial}{\partial \lambda'_{jd}} \ln I_0(\bar{\lambda}'_{jd}) \Big) \big( \langle \lambda'_{jd} \rangle - \bar{\lambda}'_{jd} \big) - \ln I_0(\bar{\lambda}'_{jd}) \Bigg] + (\ln(1 - \epsilon_{jd})) \tag{6.27}$$

$$\beta^*_{jd} = \| \beta_{jd} \mathbf{m}_{jd} + \sum_{i=1}^{N} \langle z_{ij} \rangle \langle \phi_{ijd} \rangle \mathbf{Y}_{id} \| \tag{6.28}$$

$$\mathbf{m}^*_{jd} = \frac{1}{\beta^*_{jd}} \Bigg( \beta_{jd} \mathbf{m}_{jd} + \sum_{i=1}^{N} \langle z_{ij} \rangle \langle \phi_{ijd} \rangle \mathbf{Y}_{id} \Bigg) \tag{6.29}$$

$$u^*_{jd} = u_{jd} + \beta^*_{jd} \bar{\lambda}_{jd} \Bigg( \frac{\partial}{\partial \beta^*_{jd} \lambda_{jd}} \ln I_0(\hat{\beta}^*_{jd} \bar{\lambda}_{jd}) \Bigg) \tag{6.30}$$

$$v^*_{jd} = v_{jd} + \sum_{i=1}^{N} \langle z_{ij} \rangle \langle \phi_{ijd} \rangle \Bigg( \frac{\partial}{\partial \lambda_{jd}} \ln I_0(\bar{\lambda}_{jd}) \Bigg) + \beta_{jd} \Bigg( \frac{\partial}{\partial \beta_{jd} \lambda_{jd}} \ln I_0(\beta_{jd} \bar{\lambda}_{jd}) \Bigg) \tag{6.31}$$

$$\beta'^*_{jd} = \| \beta'_{jd} \mathbf{m}'_{jd} + \sum_{i=1}^{N} \langle z_{ij} \rangle \langle 1 - \phi_{ijd} \rangle \mathbf{Y}_{id} \| \tag{6.32}$$

$$\mathbf{m}'^*_{jd} = \frac{1}{\beta'^*_{jd}} \Bigg( \beta'_{jd} \mathbf{m}'_{jd} + \sum_{i=1}^{N} \langle z_{ij} \rangle \langle 1 - \phi_{ijd} \rangle \mathbf{Y}_{id} \Bigg) \tag{6.33}$$

$$u'^*_{jd} = u'_{jd} + \hat{\beta}'^*_{jd} \bar{\lambda}'_{jd} \Bigg( \frac{\partial}{\partial \hat{\beta}'^*_{jd} \lambda'_{jd}} \ln I_0(\hat{\beta}'^*_{jd} \bar{\lambda}'_{jd}) \Bigg) \tag{6.34}$$

$$v'^*_{jd} = v'_{jd} + \sum_{i=1}^{N} \langle z_{ij} \rangle \langle 1 - \phi_{ijd} \rangle \Bigg( \frac{\partial}{\partial \lambda'_{jd}} \ln I_0(\bar{\lambda}'_{jd}) \Bigg) + \beta'_{jd} \Bigg( \frac{\partial}{\partial \beta'_{jd} \lambda'_{jd}} \ln I_0(\beta'_{jd} \bar{\lambda}'_{jd}) \Bigg) \tag{6.35}$$

$$g_j = 1 + \sum_{i=1}^{N} \langle z_{ij} \rangle \tag{6.36}$$

$$h_j = \xi_j + \sum_{i=1}^{N} \sum_{k=j+1}^{M} \langle z_{ik} \rangle \tag{6.37}$$

where $\frac{\partial}{\partial \lambda_{jd}} \ln I_0(\bar{\lambda}_{jd}) \equiv \frac{\partial}{\partial \lambda_{jd}} \ln I_0(\lambda_{jd})|_{\lambda_{jd}=\bar{\lambda}_{jd}} = \frac{I_1(\bar{\lambda}_{jd})}{I_0(\bar{\lambda}_{jd})}$ which is obtained based on the property $I_0'(\kappa) = I_1(\kappa)$ of the modified Bessel function [22]. The expected values in the above formulas are defined as

$$\langle z_{ij} \rangle = r_{ij}, \qquad \langle \lambda_{jd} \boldsymbol{\mu}_{jd}^{\mathrm{T}} \mathbf{Y}_{id} \rangle = \frac{u_{jd}^*}{v_{jd}^*} \mathbf{m}_{jd}^{*\mathrm{T}} \mathbf{Y}_{id} \tag{6.38}$$

$$\langle \phi_{ijd} \rangle = f_{ijd}, \qquad \langle \lambda'_{jd} \boldsymbol{\mu}_{jd}'^{\mathrm{T}} \mathbf{Y}_{id} \rangle = \frac{u_{jd}'^*}{v_{jd}'^*} \mathbf{m}_{jd}'^{*\mathrm{T}} \mathbf{Y}_{id} \tag{6.39}$$

$$\langle \ln \pi_j' \rangle = \psi(g_j) - \psi(g_j + h_j) \tag{6.40}$$

$$\langle \ln(1 - \pi_j') \rangle = \psi(h_j) - \psi(g_j + h_j) \tag{6.41}$$

$$\langle \lambda_{jd} \rangle = \frac{u_{jd}^*}{v_{jd}^*}, \qquad \langle \lambda'_{jd} \rangle = \frac{u_{jd}'^*}{v_{jd}'^*} \tag{6.42}$$

The features saliencies can be obtained by setting the derivative of the variational lower bound $\mathcal{L}(q)$ with respect to $\epsilon_{jd}$ to zero as

$$\epsilon_{jd} = \frac{1}{N} \sum_{i=1}^{N} f_{ijd} \tag{6.43}$$

Since the variational solutions are coupled together through the expected values of other factors, these solutions can be obtained iteratively through an EM-like algorithm as described in Algorithm 1.

**Algorithm 1** Variational inference of the infinite VM model

1: Choose the initial truncation level $M = 15$.
2: Initialize hyperparameters $\xi_j$, $\mathbf{m}_{jd}$, $\beta_{jd}$, $u_{jd}$, $v_{jd}$, $u'_{jd}$, $v'_{jd}$, $\mathbf{m}'_{jd}$, $\beta'_{jd}$.
3: **repeat**
4: *The variational E-step*: use the current values of model parameters to evaluate the expected values in Eqs. (6.38)∼(6.42).
5: *The variational M-step*: update the variational factors using Eqs. (6.18)∼(6.22).
6: Calculate features saliencies using Eq. (6.43).
7: **until** Convergence criteria is reached.

## 6.4 Experimental Results

In this section, we validate the proposed infinite VM mixture model with localized feature selection through two challenging applications, namely topic novelty detection and unsupervised images categorization. In our experiments, the initial truncation level $M$ was set to 15. The prior parameters $\mathbf{m}_{jd}$ and $\mathbf{m}'_{jd}$ were randomly initialized from the data such that $\|\mathbf{m}_{jd}\| = 1$ and $\|\mathbf{m}'_{jd}\| = 1$. The prior parameters $\beta_{jd}, u_{jd}, v_{jd}, u'_{jd}, v'_{jd}$, and $\beta'_{jd}$ were initialized as $(\beta_{jd}, u_{jd}, v_{jd}, u'_{jd}, v'_{jd}, \beta'_{jd}) = (0.1, 0.1, 0.01, 0.1, 0.01, 0.1)$. The parameters of $\xi_j$ and $\epsilon_{jd}$ were set to 0.5.

### *6.4.1 Topic Novelty Detection*

The goal of topic novelty detection in terms of text data is to develop an approach that can automatically detect novel topics in a given collection of documents (e.g., news articles). In our mixture model-based approach, a novel topic is the one that is classified as a new cluster of our infinite mixture model. In this experiment, following [7], four publicly available data sets are adopted for evaluating the performance of the proposed infinite VM mixture model with localized feature selection (referred as InVmMM-LFs). The first data set is known as the CNAE-9 data set[1] which has 1080 documents of descriptions of Brazilian companies that can be divided into 9 categorizes. The categories of this data set are equally distributed (i.e., 120 instances in each of nine categories). The other three data sets were taken from the original 20-Newsgroups data set.[2] The details of these three data sets are listed as follows. The first data set is known as "News-Related-3" which contains 300 documents with 3225 attributes from three newsgroups on related topics (talk.politics.misc, talk.politics.guns, talk.politics.mideast). The second data set is "News-Similar-3" which contains 300 documents with 1864 attributes from three newsgroups on similar topics (comp.graphics, comp.os.mswindows, comp.windows.x). The third data set is "News-Different-3" which contains 300 documents with 3251 attributes from three newsgroups on related topics (alt.atheism, rec.sport.baseball, sci.space).

In order to evaluate the novelty detection performance of the proposed InVmMM-LFs on the four testing data sets, we adopt measures including Accuracy and $F_1$ as follows

$$\text{Accuracy} = \frac{\text{Number of documents that are correctly clustered}}{\text{Total number of documents}} \tag{6.44}$$

$$F_1 = \frac{2 \times \text{precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6.45}$$

[1] https://archive.ics.uci.edu/ml/datasets/CNAE-9.

[2] http://qwone.com/~jason/20Newsgroups/.

In our approach, we perform topic novelty detection using the proposed InVmMM-LFs based on the bag-of-words paradigm. The methodology of our topic novelty detection approach can be summarized as follows: First, we applied the Rainbow package[3] [23] to remove the rare (occurred less than 30 times) and stop words (such as "the", "and", "or", etc.). Next, each document is represented by a vector of counts (i.e., a histogram containing the frequency of occurrence of each word in its vocabulary). Each document is then TF-IDF normalized and thus a $L_2$ normalized vocabulary is created. These obtained vectors are then modeled by the proposed InVmMM-LFs. Finally, the classification is performed by applying Bayes' decision rule. As new documents are observed, new topics will be discovered if the number of clusters is increased. We run our method 20 times to investigate its average performance.

In order to demonstrate the advantages of the proposed approach. We compare our approach with other topic novelty detection approaches that are based on mixture models. These approaches include: the infinite VM mixture model with global feature selection (denoted by InVmMM-Fs), the infinite VM mixture model without feature selection (denoted by InVmMM), the Gaussian mixture model with localized feature selection (denoted by GMM-LFs) [14], and the infinite VM mixture model with feature selection which is learnt using Markov chain Monte Carlo (MCMC) algorithm (denoted by InVMM-MCMC) [7].

The results by different approaches for each tested data set are shown in Tables 6.1, 6.2, 6.3, 6.4. Based on these results, it is clear that the proposed InVmMM-LFs has provided the best novelty detection performance among all tested approaches in terms of the highest $F_1$ score and the accuracy rate for each data set. The advantages of using feature selection are verified since InVmMM has obtained worse performance than approaches adopting either global feature

**Table 6.1** Topic novelty detection performance by different approaches for the CNAE-9 data set

| Methods | $F_1$ | Accuracy (%) | $M^*$ |
|---|---|---|---|
| GMM-LFs | 76.69 | 77.97 | $8.23 \pm 1.28$ |
| InVMM-MCMC | 81.81 | 83.91 | $9.04 \pm 1.01$ |
| InVmMM | 80.12 | 82.08 | $9.12 \pm 1.01$ |
| InVmMM-Fs | 82.44 | 85.08 | $9.06 \pm 0.91$ |
| InVmMM-LFs | 83.56 | 86.19 | $9.02 \pm 0.75$ |

**Table 6.2** Topic novelty detection performance by different approaches for the News-Related-3 data set

| Methods | $F_1$ | Accuracy (%) | $M^*$ |
|---|---|---|---|
| GMM-LFs | 83.78 | 82.23 | $2.87 \pm 1.02$ |
| InVMM-MCMC | 87.41 | 86.59 | $3.05 \pm 0.45$ |
| InVmMM | 85.19 | 84.38 | $3.25 \pm 0.93$ |
| InVmMM-Fs | 88.46 | 87.21 | $3.07 \pm 0.85$ |
| InVmMM-LFs | 89.67 | 89.52 | $3.04 \pm 0.33$ |

[3] http://www.cs.cmu.edu/~mccallum/bow.

**Table 6.3** Topic novelty detection performance by different approaches for the News-Similar-3 data set

| Methods | $F_1$ | Accuracy (%) | $M^*$ |
|---|---|---|---|
| GMM-LFs | 80.33 | 81.57 | $2.82 \pm 1.21$ |
| InVMM-MCMC | 84.06 | 84.58 | $3.08 \pm 0.60$ |
| InVmMM | 83.72 | 83.35 | $3.17 \pm 0.52$ |
| InVmMM-Fs | 84.08 | 85.39 | $3.21 \pm 0.29$ |
| InVmMM-LFs | 86.63 | 87.94 | $3.11 \pm 0.46$ |

**Table 6.4** Topic novelty detection performance by different approaches for the News-Different-3 data set

| Methods | $F_1$ | Accuracy (%) | $M^*$ |
|---|---|---|---|
| GMM-LFs | 86.52 | 85.93 | $2.89 \pm 0.31$ |
| InVMM-MCMC | 88.99 | 90.00 | $3.12 \pm 0.11$ |
| InVmMM | 87.38 | 87.29 | $3.23 \pm 0.63$ |
| InVmMM-Fs | 88.72 | 89.64 | $3.16 \pm 0.45$ |
| InVmMM-LFs | 90.18 | 92.09 | $3.09 \pm 0.51$ |

selection (InVmMM-Fs) or localized feature selection (InVmMM-LFs). Since InVmMM-LFs outperformed InVmMM-Fs, it also confirms that localized feature selection has better performance than global feature selection in topic novelty detection. Among all approaches, GMM-LFs has obtained the worst performance in terms of the lowest $F_1$ score and the accuracy rate for each data set. This demonstrates that VM mixture model is better than Gaussian mixture model for modeling $L_2$ normalized data.

### *6.4.2 Unsupervised Images Categorization*

In this experiment, the proposed InVmMM-LFs is applied to categorize images based on the bag-of-visual-words representation. We test our image categorization approach on the Oxford flowers data set [24] which includes 17 different categories of flowers with 80 images for each category. In our experiment, we adopt a subset of this data set which contains ten classes: daffodil, snowdrop, lily valley, bluebell, pansy, tiger lily, tulip, fritillary, sunflower, and daisy. Figure 6.1 demonstrates several sample images of the Oxford flowers data set.

Our image categorization approach can be summarized as follows. First, the PCA-SIFT descriptors[4] (36-dimensional) [25] are extracted from each image using the difference-of-Gaussians (DoG) interest point detector [26]. Next, an accelerated version of the $K$-means algorithm [27] is used to construct a visual vocabulary by quantizing these PCA-SIFT vectors into visual words. As a result, each image is represented as a frequency histogram over the visual words. Then, the probabilistic latent semantic analysis (pLSA) model [28, 29] is applied to the obtained histograms

[4]Source code of PCA-SIFT: http://www.cs.cmu.edu/~yke/pcasift.

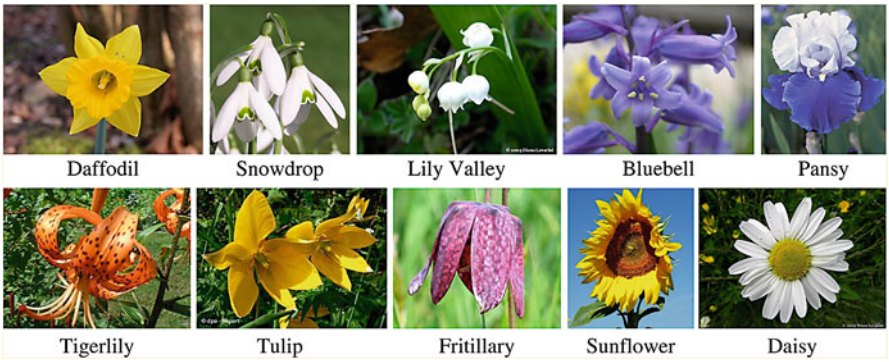



**Fig. 6.1** Sample images from each class in the Oxford flowers data set

**Table 6.5** The average classification accuracy and the number of components ($M^*$) computed over 20 runs by different algorithms

| Method | $M^*$ | Accuracy (%) |
|---|---|---|
| GMM-LFs | $9.29 \pm 0.46$ | 74.93 |
| InVMM-MCMC | $9.62 \pm 0.33$ | 79.34 |
| InVmMM | $9.51 \pm 0.25$ | 77.26 |
| InVmMM-Fs | $9.59 \pm 0.27$ | 80.01 |
| InVmMM-LFs | $9.67 \pm 0.31$ | 82.15 |

to represent each image by a 45-dimensional vector (i.e., the number of latent aspects) and then $L_2$ normalized. Finally, the proposed InVmMM-LFs is applied to cluster the testing images by assigning the image to the category which has the highest posterior probability according to Bayes' decision rule. In our case, half of the data set was used to construct the visual vocabulary, and the other half was used for testing.

We compare the proposed image categorization approach based on InVmMM-LFs with the ones based on GMM-LFs, InVMM-MCMC, InVmMM, and InVmMM-Fs. The average categorization accuracy and the number of components ($M^*$) computed over 20 runs by different algorithms are shown in Table 6.5. As we can observe from this table, it is clear that the InVmMM-LFs achieves the best performance among all tested methods in terms of the highest categorization accuracy rate (82.15%) and the most accurate estimated number of clusters (9.67). This result verified the advantages of using VM mixture model and localized feature selection for categorizing images. We may also notice that both cases InVmMM-LFs and InVmMM-Fs outperform InVmMM, which demonstrates the merits of integrating feature selection into mixture models for clustering.

## 6.5 Conclusion

In this chapter, a probabilistic model-based approach based on infinite VM mixture model has been proposed for clustering $L_2$ normalized data. Comparing with finite mixture model in which the number of mixture components have to be determined through extra efforts, the infinite mixture VM model is a nonparametric model such that the number of mixture components is assumed to be infinite initially and will be inferred automatically during the learning process. In order to improve clustering performance for high-dimensional data, a localized feature selection scheme is integrated into the infinite VM mixture model which can effectively detect irrelevant features based on the estimated feature saliencies. The proposed infinite mixture model with localized feature selection is learnt through variational inference that can estimate model parameters and feature saliencies with closed-form solutions. Our model-based clustering approach is validated through two challenging applications, namely topic novelty detection and unsupervised images categorization.

**Acknowledgements** The completion of this work was supported by the National Natural Science Foundation of China (61876068), the Natural Science Foundation of Fujian Province (2018J01094), and the Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University (ZQNPY510).

## References

1. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley, New York (2000)
2. Fan, W., Bouguila, N., Ziou, D.: Variational learning for finite Dirichlet mixture models and applications. IEEE Trans. Neural Netw. Learn. Syst. **23**(5), 762–774 (2012)
3. Fan, W., Sallay, H., Bouguila, N.: Online learning of hierarchical Pitman–Yor process mixture of generalized Dirichlet distributions with feature selection. IEEE Trans. Neural Netw. Learn. Syst. **28**(9), 2048–2061 (2017)
4. Fan, W., Bouguila, N., Liu, X.: A hierarchical Dirichlet process mixture of GID distributions with feature selection for spatio-temporal video modeling and segmentation. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, pp. 2771–2775. IEEE, Piscataway (2017)
5. Fan, W., Bouguila, N.: Online learning of a Dirichlet process mixture of Beta-Liouville distributions via variational inference. IEEE Trans. Neural Netw. Learn. Syst. **24**(11), 1850–1862 (2013)
6. Fan, W., Bouguila, N.: Expectation propagation learning of a Dirichlet process mixture of Beta-Liouville distributions for proportional data clustering. Eng. Appl. Artif. Intell. **43**, 1–14 (2015)
7. Amayri, O., Bouguila, N.: Infinite Langevin mixture modeling and feature selection. In: 2016 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016, pp. 149–155. IEEE, Piscataway (2016)
8. Amayri, O., Bouguila, N.: RJMCMC learning for clustering and feature selection of l2-normalized vectors. In: International Conference on Control, Decision and Information Technologies, CoDIT 2016, pp. 269–274. IEEE, Piscataway (2016)
9. Amayri, O., Bouguila, N.: A Bayesian analysis of spherical pattern based on finite Langevin mixture. Appl. Soft Comput. **38**, 373–383 (2016)

10. Amayri, O., Bouguila, N.: On online high-dimensional spherical data clustering and feature selection. Eng. Appl. Artif. Intell. **26**(4), 1386–1398 (2013)
11. Korwar, R.M., Hollander, M.: Contributions to the theory of Dirichlet processes. Ann. Probab. **1**, 705–711 (1973)
12. Ferguson, T.S.: A Bayesian analysis of some nonparametric problems. Ann. Stat. **1**(2), 209–230 (1973)
13. Blei, D.M., Jordan, M.I.: Variational inference for Dirichlet process mixtures. Bayesian Anal. **1**, 121–144 (2005)
14. Li, Y., Dong, M., Hua, J.: Simultaneous localized feature selection and model detection for Gaussian mixtures. IEEE Trans. Pattern Anal. Mach. Intell. **31**, 953–960 (2009)
15. Attias, H.: A variational Bayes framework for graphical models. In: Proceedings of the Advances in Neural Information Processing Systems (NIPS), pp. 209–215 (1999)
16. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Mach. Learn. **37**(2), 183–233 (1999)
17. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Berlin (2006)
18. Fan, W., Bouguila, N.: Nonparametric localized feature selection via a Dirichlet process mixture of generalized Dirichlet distributions. In: Neural Information Processing—19th International Conference, ICONIP 2012, pp. 25–33 (2012)
19. Fan, W., Bouguila, N., Ziou, D.: Unsupervised anomaly intrusion detection via localized Bayesian feature selection. In: 11th IEEE International Conference on Data Mining, ICDM 2011, pp. 1032–1037. IEEE, Piscataway (2011)
20. Law, M.H.C., Figueiredo, M.A.T., Jain, A.K.: Simultaneous feature selection and clustering using mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **26**(9), 1154–1166 (2004)
21. Sethuraman, J.: A constructive definition of Dirichlet priors. Stat. Sin. **4**, 639–650 (1994)
22. Taghia, J., Ma, Z., Leijon, A.: Bayesian estimation of the von Mises-Fisher mixture model with variational inference. IEEE Trans. Pattern Anal. Mach. Intell. **36**(9), 1701–1715 (2014)
23. McCallum, A.K.: Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/~mccallum/bow (1996)
24. Nilsback, M.-E., Zisserman, A.: A visual vocabulary for flower classification. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 1447–1454. IEEE, Piscataway (2006)
25. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 506–513. IEEE, Piscataway (2004)
26. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1615–1630 (2005)
27. Elkan, C.: Using the triangle inequality to accelerate k-means. In: Proceedings of the Twentieth International Conference on International Conference on Machine Learning, pp. 147–153 (2003)
28. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Mach. Learn. **42**(1/2), 177–196 (2001)
29. Bosch, A., Zisserman, A., Munoz, X.: Scene classification via pLSA. In: Proceedings of the 9th European Conference on Computer Vision (ECCV), pp. 517–530. Springer, Berlin (2006)

# Chapter 7
# Deriving Probabilistic SVM Kernels from Exponential Family Approximations to Multivariate Distributions for Count Data


**Nuha Zamzami and Nizar Bouguila**

**Abstract** This work aims to propose a robust hybrid probabilistic learning approach that combines appropriately the advantages of both the generative and discriminative models for modeling count data. We build new probabilistic kernels based on information divergences and Fisher score from efficient approximations to multivariate distributions for support vector machines (SVMs). More precisely, we drive probabilistic kernels from the mixture of exponential family approximation to two powerful generative models for count data, namely the multinomial compound Dirichlet (DCM) and the generalized Dirichlet multinomial (GDM). The developed hybrid models are introduced as effective SVM kernels able to incorporate prior knowledge about the nature of data involved in the problem at hand and, therefore, permits a good data discrimination. We demonstrate the flexibility and the merits of the proposed frameworks for the problem of analyzing activities in surveillance scenes.

## 7.1 Introduction

Clustering is among the significant data mining tasks that have been extensively studied in the past in order to predict the natural grouping for unlabeled data [32]. It is generally viewed as a density estimation problem, i.e., the model makes

N. Zamzami (✉)
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
e-mail: n_zamz@encs.concordia.ca

N. Bouguila
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca



N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6_7

inference through probabilistic assumptions of the data distributions [26]. Thus, they offer a principled effective way for handling uncertainty and they are practical especially when dealing with missing and incomplete data [55]. A popular model-based (generative) approach for clustering is finite mixture models which offer a considerable practical value in modeling heterogeneous data [27, 58], based on estimating the class-conditional distributions, and the prior probabilities of each class, which are then used for clustering using Bayes' rule [7]. On the other hand, discriminative approaches focus directly on the classification problem via learning the class boundaries without regard to the underlying class densities [66]. Due to their computational efficiency and good discrimination capabilities, support vector machines (SVMs) [12, 75] became standard supervised learning tool with benchmark results. Selecting an appropriate SVM kernel is really a challenging task for machine learning and data mining applications. Indeed, the classic kernels, that SVM-based discriminative classifiers often rely on, are not generally suitable for count data, as such kernels do not take into consideration the nature of data. A better approach is to generate the kernels directly from the data, which could be also called generative kernels [1]. Consequently, relevant efforts have been made recently to combine the advantages of both approaches by developing hybrid generative/discriminative algorithms (see, for instance, [5, 9, 30, 46, 81]). The idea is to capture the intrinsic properties of the data to classify, taking into account prior knowledge of the problem domain. Such algorithms have shown to be powerful tools that generally provide lower test errors and better accuracies than either fully generative or discriminative techniques [31, 64].

In this work, we address the problem of classification where the data consists of bags of count vectors by incorporating efficient mixture models into SVMs. As choosing the components probability density functions (PDFs) is at the heart of finite mixture modeling, we suggest the consideration of flexible and accurate distributions to model count data, namely the exponential approximation to Dirichlet compound multinomial (EDCM) [25] and the exponential approximation to generalized Dirichlet multinomial (EGDM) [83]. The EDCM and EGDM distributions are approximations of the DCM [52] and GDM [8], respectively, that bring them into the family of exponential distributions. Our choice for these approximations is justified by the fact that the exponential distributions are usually easier to evaluate and their parameters are simpler to estimate [4, 21]. Moreover, both EDCM and EGDM have shown to be computationally efficient and have high flexibility in count data modeling with superior performance in many challenging applications [25, 37, 82, 83]. We develop several flexible SVM kernels that make intelligent use of unlabeled count data to achieve accurate classification results. These kernels are defined on probabilistic generative models learned from the data. In particular, instead of using EDCM and EGDM mixtures directly for classification, we build probabilistic kernels based on Fisher scores and information divergence. We validated the proposed developed hybrid model using publicly available and widely used datasets for activity analysis in surveillance scenes.

The rest of this chapter is organized as follows: The next section will review the support vector machines (SVMs). In Sect. 7.3, we present the considered

mixture models and the deterministic annealing expectation–maximization (DAEM) algorithm for learning the parameters. Then, in Sect. 7.4 we introduce our statistical generative/discriminative framework and all related details. Section 7.5 is devoted to the experimental results. Finally, we conclude this work in Sect. 7.6.

## 7.2 Support Vector Machines Kernels

Support vector machines (SVMs), as a type of classifiers, are well known for supervised learning and applicable to both classification and regression problems [6, 75]. Since SVM classifier was introduced in [75], it gained popularity due to its good generalization, global solution, number of tuning parameters and their solid theoretical foundation. The development of efficient SVMs implementations led to broadening its applications [17, 51, 59]. The SVM is designed for binary-classification problems, assuming that the training set is separable by a hyperplane where the complexity of the hyperplane can be bounded in terms of another quantity, the margin. The margin is defined as the minimal distance of an example to a decision surface. Thus, if we bound the margin of a function class from below, we can control its complexity. In the statistical framework, learning means to estimate a function from a set of examples (the training sets). Thus, a learning machine must choose one function from a given set of functions, which minimizes a certain risk. Support vector learning implements this insight that the risk is minimized when the margin is maximized [70].

Let $\{(\mathbf{X}_1, C_1), \ldots, (\mathbf{X}_l, C_l)\}$, $\mathbf{X}_i \in \mathbb{R}^N$ be a training set of random independent identically distributed vectors, with labels $C_i \in \{+1, -1\}$ belonging to either of two linearly separable classes $C_1$ and $C_2$. The decision function of the SVM classifier is given by [75]:

$$f(\mathbf{X}) = \text{sign}\left(\sum_{i=1}^{l} C_i \delta_i K(\mathbf{X}.\mathbf{X}_i) + b\right), \tag{7.1}$$

where $l$ is the number of support vectors containing the relevant information about the classification problem, $\delta_i$ are the weights of the support vectors determined by solving a constrained quadratic programming problem which aims to maximize the margin between the classes, $b$ is a bias term, and $K(X.X_i)$ is a symmetric positive definite kernel function. A challenging problem in the case of SVMs is the choice of the kernel function which is actually a measure of similarity between two vectors. In case the data are not linearly separable, it can be mapped into a high dimensional feature space using a kernel function to simplify the computation of the inner product value of the transformed data in the feature space [6, 70]. The generally used kernel functions are polynomial, radial basis function (RBF), and sigmoid [40, 48].

The Fisher kernel, initially proposed in [31], is an example of previous efforts in generating more flexible kernels, which has been widely used in the literature. The main idea is to exploit the geometric structure on the statistical manifold by mapping a given individual sequence of vectors into a single feature vector, defined in the gradient log-likelihood space. Given its ability to incorporate prior knowledge about the data, Fisher kernels have shown excellent performance in [23], for instance, where Gaussian mixture model-based kernel functions are used for speech emotion recognition. Moreover, it has been successfully implemented in many applications that involve discrete data such as handwriting recognition, speech recognition, facial expression analysis, and bio-informatics based on mixture of multinomials [74], as well as, spam and text categorization and hierarchical classification of vacation images based on mixture of Dirichlet compound multinomial (DCM) [10].

An alternative to Fisher kernel is called probability product kernels investigated in [35], where two special cases were introduced, namely Bhattacharyya and expected likelihood kernels. Moreover, several SVM kernels have been generated based on information divergence between distributions, such as Kullback–Leibler (KL) divergence kernel [16, 60], Reńyi and Jensen–Shannon kernels [47, 65]. These kernels have been successfully implemented with good results in case of Gaussian [14, 63] and non-Gaussian data [5, 9, 81].

## 7.3 Finite Mixtures of Exponential Distributions

Finite mixtures are flexible and powerful probabilistic model-based approach for unsupervised learning of multivariate data [28, 71]. In mixture modeling, the data are assumed to be generated from a mixture of subpopulations. Let $\mathcal{X}$ to be an observed dataset with $N$ data instances $\mathcal{X} = \{X_1, \ldots, X_N\}$, where $\mathbf{X}_i = (x_{i1}, \ldots, x_{iD})$ is drawn from a superposition of $M$ densities of the form:

$$P(\mathbf{X}_i|\pi, \theta) = \sum_{j=1}^{M} \pi_j \, \mathcal{P}(\mathbf{X}_i|\theta_j), \tag{7.2}$$

where $\pi_j$ ($0 < \pi_j < 1$ and $\sum_{j=1}^{M} \pi_j = 1$) are the mixing proportions. Each $\mathcal{P}(\mathbf{X}|\theta_j)$ represents mixture component $j$ and has its own parameters $\theta_j$. For every observed data point $\mathbf{X}_i$, there is a corresponding latent variable $\mathbf{Z}_i$. The set $\mathcal{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_N\}$ denotes the missing group-indicator vectors for data elements in the $j$th cluster. The value of $z_{ij}$ is satisfying $z_{ij} \in \{0, 1\}$, as a particular element $z_{ij}$ is equal to one and all other elements are equal to 0. The complete data are considered to be $(\mathcal{X}, \mathcal{Z}|\Theta)$, where $\Theta$ is the set of all latent variables and parameters. The complete data log-likelihood corresponding to a mixture model, with $M$ components, is given by:

$$\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{M} z_{ij}\left(\log \mathcal{P}(\mathbf{X}_i|\theta_j) + \log \pi_j\right). \tag{7.3}$$

### 7.3.1 The Exponential Approximation to Dirichlet Compound Multinomial (EDCM)

The EDCM [25] is an approximation of the DCM (Dirichlet compound multinomial) [52]. DCM was introduced based on the fact that the Dirichlet is a conjugate prior to the multinomial probability distribution which has several computational advantages [54]. However, it has some limitations which include that it does not belong to the exponential family, its expression lacks intuitiveness, and its parameters cannot be estimated quickly. The EDCM, on the other hand, possesses several useful properties which allow it to perform fast and model well high dimensional data with bursty behaviors that appear in text [18, 39] and visual elements [36].

Define $\mathbf{X} = (x_1, \ldots, x_D)$ as a vector of counts representing a document or an image, where $x_d$ is the frequency of the word or visual word, $d$. The probability of generating the vector $\mathbf{X}$ using the DCM with parameter vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_D)$ is given by [52]:

$$\mathcal{DCM}(\mathbf{X}|\boldsymbol{\alpha}) = \frac{n!}{\prod_{d=1}^{D}(x_d)!}\frac{\Gamma(s)}{\Gamma(s+n)}\prod_{d=1}^{D}\frac{\Gamma(x_d+\alpha_d)}{\Gamma(\alpha_d)}, \tag{7.4}$$

where $n = \sum_{d=1}^{D} x_d$, and $s = \sum_{d=1}^{D} \alpha_d$. Note that comparing to the multinomial, the DCM retains one more degree of freedom as its parameters are not required to sum to one.

Elkan [25] proposed the EDCM, noting the sparsity nature of textual data, as most documents contain only a small subset of the entire vocabulary. In this approximation, only non-zero word counts $x_d$ are used for computation efficiency. EDCM, as an approximation to DCM, can be made if $\alpha \ll 1$, which found to be true for the majority of documents collections and images databases represented as bag-of-features (BoF) [20]. The EDCM is formulated as:

$$\mathcal{EDCM}(\mathbf{X}|\alpha) = \frac{n!}{\prod_{d:x_d\geq 1} x_d}\frac{\Gamma(s)}{\Gamma(s+n)}\prod_{d:x_d\geq 1}\alpha_d. \tag{7.5}$$

Thus, re-writing Eq. (7.5) as an exponential density gives:

$$q(\mathbf{X}|\alpha) = \left( \prod_{d:x_d \geq 1} x_d^{-1} \right) n! \frac{\Gamma(s)}{\Gamma(s+n)} \exp\left[ \sum_{d=1}^{D} I(x_d \geq 1) \log \alpha_d \right], \tag{7.6}$$

where $I(x_d \geq 1)$ is an indicator equals to 1 if the word $d$ appears at least once in a vector $\mathbf{X}$, and 0 otherwise.

### 7.3.2 The Exponential Approximation to Generalized Dirichlet Multinomial (EGDM)

Although the Dirichlet distribution is commonly used as a prior to the multinomial, the generalized Dirichlet distribution has shown to be a more appropriate prior for naive Bayesian classifiers. This is due to the fact that generalized Dirichlet overcomes several limitations of the Dirichlet including the negative-correlation and the equal-confidence requirements [78, 79]. Moreover, the independence property of GD distribution, defined by the ability to sample each entry of the random vector from independent beta distributions, provides more flexibility than the Dirichlet distribution [13].

Bouguila [8] introduced the generalized Dirichlet multinomial (GDM), which is the composition of the generalized Dirichlet and the multinomial in the same way of DCM. The probability density function of the GDM distribution with parameters $\theta = \{\alpha_1, \beta_1, \ldots, \alpha_D, \beta_D\}$ is given by:

$$\mathcal{GDM}(\mathbf{X}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{\Gamma(n+1)}{\prod_{d=1}^{D+1} \Gamma(x_d+1)} \prod_{d=1}^{D} \frac{\Gamma(\alpha_d+\beta_d)}{\Gamma(\alpha_d)\Gamma(\beta_d)} \prod_{d=1}^{D} \frac{\Gamma(\alpha'_d)\Gamma(\beta'_d)}{\Gamma(\alpha'_d+\beta'_d)}, \tag{7.7}$$

where $n = \sum_{d=1}^{D+1} x_d$, $\alpha'_d = \alpha_d + x_d$, and $\beta'_d = \beta_d + x_{d+1} + \cdots + x_{D+1}$, for $d = 1, \ldots, D$. Note that the Dirichlet compound multinomial (DCM) distribution is a special case of GDM by taking $\beta_d = \alpha_{d+1} + \beta_{d+1}$. It is important to note that the generalized Dirichlet is a tree of beta distributions, and the GDM is a tree of 2-D DCMs [84]. Similar to DCM, GDM does not belong to the exponential family.

Indeed, the generalized Dirichlet multinomial (GDM) has shown to be an effective alternative to DCM that achieves high clustering accuracy in different applications [8, 80, 84]. However, it shares similar problems to the ones with DCM including that its parameters cannot be estimated quickly. To simplify the parameter estimation process and reduce the computation in high-dimensional spaces, we proposed, in an earlier work [83], an efficient approximation to the GDM motivated by the superior performance of EDCM. The approximation, EGDM, is a member of

the exponential family of distributions and it captures the burstiness phenomenon successfully and correctly, while being many times faster and computationally efficient compared to the corresponding GDM. It was possible to reduce GDM to a member of the exponential family via a suitable transformation and re-parameterization considering some properties of logarithm and gamma functions. Moreover, in case of GDM, we found experimentally that $\alpha_d \ll \beta_d \ll 1$ for almost all words $w$ based on different count datasets. The approximation, EGDM, can be written in the exponential family form as:

$$\mathcal{EGDM}(\mathbf{X}) = \Big( \prod_{d:x_d \geq 1} x_d \Big)^{-1} \prod_{d:x_d \geq 1} \frac{\Gamma(z_d)}{\Gamma(x_d + z_d)} n! \\ \exp\Big[ \sum_{d=1}^{D} I(x_d \geq 1) \log \frac{\alpha_d \beta_d}{(\alpha_d + \beta_d)} \Big], \tag{7.8}$$

where $z_d = x_{d+1} + \cdots + x_{D+1}$ is the cumulative sum. As in EDCM, $I(x_d \geq 1)$ is an indicator that represents whether a word $d$ appears at least once in the vector $\mathbf{X}$.

### 7.3.3 Mixture Models Parameters Estimation

A widely used method for estimating mixture parameters $\Theta$ is the maximum likelihood estimate (MLE) solution [22] through expectation–maximization (EM) approach [56] on the complete likelihood, which is commonly useful in observations that can be viewed as incomplete data. EM algorithm generates a sequence of models with non-decreasing log-likelihood on the data, and highly depends on the initialization, thus, has an issue of poor local maxima. Researchers proposed different extensions to overcome the EM problems. One of the successful extensions is the deterministic annealing method (DAEM) [73], which is used in [25] and [83], to estimate the parameters of EDCM and EGDM, respectively. Some interesting justifications about using the deterministic annealing procedure can be found in [25, 73, 83].

DAEM uses multiple phases each with a value of a temperature parameters set, where the final $\Theta$ parameters in each phase are used as initial values in the next one. In case of EDCM and EGDM, three phases are considered where each phase runs EM until convergence. The computational temperature parameter $T$ has been set to $T = 25$, $T = 5$, and lastly $T = 1$. When applying the deterministic annealing procedure, the posterior probabilities will be computed in the *E-step* as:

$$\hat{z}_{ij}^{(t)} = \frac{\Big( P(\mathbf{X}_i | \theta_j^{(t)})\, \pi_j^{(t)} \Big)^{\tau}}{\sum_{j=1}^{M} \Big( P(\mathbf{X}_i | \theta_j^{(t)})\, \pi_j^{(t)} \Big)^{\tau}}, \tag{7.9}$$

where $\tau = \frac{1}{T}$, and $T$ corresponds to the computational temperature. Then, in the *M-step*, the parameters estimates will be updated according to:

$$\hat{\Theta}^{(t+1)} = \arg\max_{\Theta} \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta, \Theta^{(t)}). \tag{7.10}$$

The interested reader is referred to Elkan's paper [25] for details of the algorithm for estimating EDCM parameters, where a similar approach is used in case of EGDM.

## 7.4 Generative/Discriminative Models for Count Data

In this section, we develop SVM kernels based on EDCM and EGDM finite mixture models that address certain practical shortcomings of classic kernels. Let two multimedia objects $O$ and $O'$ represent two sequences of count vectors $\mathcal{X} = \{X_1, \ldots, X_N\}$ and $\mathcal{X}' = \{X'_1, \ldots, X'_{N'}\}$, respectively. It is possible to assume that each sequence has been generated by an $M$-component finite mixture model $p(\mathcal{X}|\Theta)$ and $q(\mathcal{X}|\Theta')$ defined on $\Omega$ space, where $\Omega$ is the $p$-dimensional space of the considered distribution. Each individual object has its own size $N$ as a given image, for instance, can be represented by a bag of pixel vectors of a set of local descriptors [29, 33]. In the following subsections, we derive different kernels based on probabilistic distances and Fisher score to tackle the problem of count data sequence classification using SVM. An important feature of probabilistic kernels is the existence of a closed-form expression in order to be able to evaluate directly the kernel function without the need of expensive Monte Carlo approximations. In the case of EGDM, however, closed-form expressions exist only for the Reńyi and Jensen–Shannon kernels.[1]

### *7.4.1 Fisher Kernels*

The Fisher kernel, proposed in [31], is based on extracting Fisher scores $U_{\mathcal{X}}(\Theta) = \bigtriangledown \log(p(\mathcal{X}|\Theta))$ from the generative model and converting them into a kernel to feed SVMs. Each component of $U_{\mathcal{X}}(\Theta)$ is the derivative of the log-likelihood of the sequence $\mathcal{X}$ with respect to a particular parameter of the mixture model. The kernel is then defined as:

$$\mathcal{K}(\mathcal{X}, \mathcal{X}') = U^{tr}_{\mathcal{X}}(\Theta) F^{-1}(\Theta) U_{\mathcal{X}'}(\Theta'), \tag{7.11}$$

[1] Indeed, closed form expressions for the other two kernels (i.e., Bhattacharyya and Kullback–Leibler) do exist for EDCM, but not for EGDM. In order to provide a fair comparison, we considered the closed form only if it exists for both models.

where $F^{-1}(\Theta)$ is the Fisher information matrix which can be approximated by the identity matrix [31]. Despite the quadratic complexity of Fisher kernel, it is a highly principled statistical approach as it has been shown in [72]. One drawback, however, is that Fisher kernel does not generally preserve the nonlinearities implied by the corresponding generative model [35].

Here, we develop Fisher kernels for the two considered generative models. In the case of finite mixture model of EDCM (Eq. 7.6), the corresponding feature space is $(M(D+2)-1)$-dimensional. By computing the gradient of $\log p(\mathcal{X}|\Theta)$ with respect to $\pi_j$, $j = 1, \ldots, M$, which is the same for any mixture model, gives:

$$\frac{\partial \mathcal{L}(\mathcal{X}|\Theta)}{\partial \pi_j} = \sum_{i=1}^{N} \left[ \frac{\hat{z}_{ij}}{\pi_j} - \frac{\hat{z}_{ij}}{\pi_1} \right], \quad j = 2, \ldots, M. \tag{7.12}$$

Considering the unity constraint on mixing weights, we have only $M-1$ free parameters, which explains the fact that the previous gradient equation is defined for $j \geq 2$ as $\pi_1$ can be determined knowing the values of the other mixing parameters $(\pi_1 = 1 - \sum_{j=2}^{M} \pi_j)$. Furthermore, computing the gradient with respect to the model parameters $\alpha_{jd}$, $d = 1, \ldots, D+1$, reasonably straightforward manipulations give:

$$\frac{\partial \mathcal{L}(\mathcal{X}|\Theta)}{\partial \alpha_{jd}} = \sum_{i=1}^{N} \hat{z}_{ij} \left[ \Psi(s_j) - \Psi(s_j + n_i) + I(x_{id} \geq 1) \left( \frac{1}{\alpha_{jd}} \right) \right]. \tag{7.13}$$

The Fisher kernel based on finite mixture of EGDM (Eq. 7.8) has a corresponding feature space with size $(M(2D+1)-1)$, and in this case we have:

$$\frac{\partial \mathcal{L}(\mathcal{X}|\Theta)}{\partial \alpha_{jd}} = \sum_{i=1}^{N} \hat{z}_{ij} \left[ I(x_{id} \geq 1) \left( \frac{1}{\alpha_{jd}} - \frac{1}{\alpha_{jd} + \beta_{jd}} \right) \right], \tag{7.14}$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\Theta)}{\partial \beta_{jd}} = \sum_{i=1}^{N} \hat{z}_{ij} \left[ I(x_{id} \geq 1) \left( \frac{1}{\beta_{jd}} - \frac{1}{\alpha_{jd} + \beta_{jd}} \right) \right]. \tag{7.15}$$

### *7.4.2 Probability Product Kernels*

An alternative approach is to generate SVM kernels between probabilistic distributions $\mathcal{K} : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}$ that injects the domain knowledge and invariance of generative models to SVMs [16]. In particular, probability product kernels [35] map data points in the input space to distributions over the sample space and a general inner product is then evaluated as the integral of the product of pairs of distributions and defined as:

$$\mathcal{K}\Big(p(\mathbf{X}|\Theta), q(\mathbf{X}|\Theta')\Big) = \int_0^{+\infty} p(\mathbf{X}|\Theta)^{\rho} q(\mathbf{X}|\Theta')^{\rho} d\mathbf{X}, \tag{7.16}$$

where $\rho$ is a positive parameter. An important special case of probability product kernels (when $\rho = 1/2$) is the Bhattacharyya kernel, originally proposed by Jebara and Kondor [34] which, despite its cubic complexity, has a main advantage of nonlinear flexibility [35]. The Bhattacharyya kernel is defined as follows:

$$\mathcal{K}_{BH}\Big(p(\mathbf{X}|\Theta), q(\mathbf{X}|\Theta')\Big) = \int_0^{+\infty} \sqrt{p(\mathbf{X}|\Theta), q(\mathbf{X}|\Theta')} d\mathbf{X}. \tag{7.17}$$

In case of EDCM, we could find a closed form for this kernel (see Appendix 1):

$$\mathcal{K}_{BH}\Big(p(\mathbf{X}|\Theta), q(\mathbf{X}|\Theta')\Big) = \frac{\Gamma(\frac{1}{2}(s+n+s'+n))\sqrt{\Gamma(s)\Gamma(s')}}{\Gamma(\frac{1}{2}s+\frac{1}{2}s')\sqrt{\Gamma(s+n)\Gamma(s'+n)}}. \tag{7.18}$$

However, we note that it is not possible to compute the Bhattacharyya kernel in a closed form from EGDM mixture density. Thus, in the absence of closed form, we can approximate the kernel using Monte Carlo simulation [16], as:

$$\begin{aligned}\mathcal{K}_{BH}\Big(p(\mathbf{X}|\Theta), q(\mathbf{X}|\Theta')\Big) \approx \frac{\beta}{N_1}\sum_{i=1}^{N_1}\frac{p^{1/2}(\mathbf{X}_i|\Theta)}{Z_1}p^{1/2}(\mathbf{X}_i|\Theta) \\ + \frac{1-\beta}{N_2}\sum_{i=1}^{N_2}\frac{q^{1/2}(\mathbf{X}_i|\Theta')}{Z_2}q^{1/2}(\mathbf{X}_i|\Theta'),\end{aligned} \tag{7.19}$$

where $\beta \in [0, 1]$, and $\mathbf{X}_1, \ldots, \mathbf{X}_{N_1}$ and $\mathbf{X}_1, \ldots, \mathbf{X}_{N_2}$ are generated from $p(\mathbf{X}|\Theta)$ and $q(\mathbf{X}|\Theta')$ densities, respectively. $Z_1$, and $Z_2$ are normalized factors for $p$ and $q$ densities after they are taken to the power of $\rho$.

### *7.4.3 Kernels Based on Information Divergence*

The main idea of information divergence kernel is to replace the kernel computation in the original sequence space by computation in the probability density functions (PDFs) space (i.e., the kernel becomes a measure of similarity between probability distributions) [16, 35]. For instance, researchers have derived a kernel distance based on the symmetric Kullback–Leibler (KL) divergence [43], which was applied successfully for speaker identification, image classification, and visual recognition [60, 62, 76]. The information divergence-based kernels between distributions is given by:

$$\mathcal{K}\Big(p(\mathbf{X}|\Theta), q(\mathbf{X}|\Theta')\Big) = \exp[-a\ F(p(\mathbf{X}|\Theta), q(\mathbf{X}|\Theta'))], \tag{7.20}$$

where $a > 0$ is a kernel parameter included for numerical stability, and $F(p(\mathbf{X}|\Theta), q(\mathbf{X}|\Theta'))$ is any information divergence measure as shown below.

#### 7.4.3.1 Kullback–Leibler Divergence Kernels

The Kullback–Leibler kernel is based on the symmetric Kullback–Leibler divergence (KL) [43], that measure the dissimilarity between two probability distributions $p(\mathbf{X}|\Theta)$, and $q(\mathbf{X}|\Theta')$, and is given by:

$$F(p(\mathbf{X}|\Theta), q(\mathbf{X}|\Theta')) = \int_0^{+\infty} \Big[p(\mathbf{X}|\Theta) \log \frac{p(\mathbf{X}|\Theta)}{q(\mathbf{X}|\Theta')} + q(\mathbf{X}|\Theta') \log \frac{q(\mathbf{X}|\Theta')}{p(\mathbf{X}|\Theta)}\Big] d\mathbf{X}. \tag{7.21}$$

The KL divergence has a closed-form expression in case of EDCM distribution and is given by (see Appendix 2):

$$\begin{aligned} &KL(p(X|\Theta), q(X|\Theta')) \\ &\quad = \log\left[\frac{\Gamma(s)\Gamma(s'+n)}{\Gamma(s')\Gamma(s+n)}\right] + \sum_{d=1}^{D} \Big(\Psi(s+n) - \Psi(s)\Big)(\alpha_d - \alpha'_d). \end{aligned} \tag{7.22}$$

In the case of EGDM distribution, we cannot find a closed-form expression for the KL-divergence, thus like the previous kernel we consider Monte Carlo simulation.

#### 7.4.3.2 Reńyi and Jensen–Shannon Kernels

We will consider two other special probabilistic kernels, the Reńyi and Jensen–Shannon kernels, which have been introduced in [16], as a generalization of the symmetric Kullback–Leibler kernel. The Reńyi kernel is based on the symmetric Reńyi divergence [65], such that:

$$\begin{aligned} \mathcal{K}_R\Big(p(\mathbf{X}|\Theta), q(\mathbf{X}|\Theta')\Big) = \Bigg[ &\int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} q(\mathbf{X}|\Theta')^{1-\sigma} dX \\ &\times \int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} q(\mathbf{X}|\Theta')^{1-\sigma} dX \Bigg]^{a/(1-\sigma)}, \end{aligned} \tag{7.23}$$

where $\sigma > 0$ and $\sigma \neq 1$ is the order of Rényi divergence, which control the amount of smoothing for the distribution. In case of our generative models, closed-form

expressions exist for the Reńyi divergence. These expressions for the EDCM and EGDM distributions, respectively, are given by (see Appendix 3):

$$\int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} q(\mathbf{X}|\Theta')^{1-\sigma} dX = \left[\frac{\Gamma(s)}{\Gamma(s+n)}\right]^{\sigma} \left[\frac{\Gamma(s')}{\Gamma(s'+n)}\right]^{1-\sigma} \times \frac{\Gamma(\sigma s+n)\Gamma((1-\sigma)s'+n)}{\Gamma(\sigma s)\Gamma((1-\sigma)s')} \tag{7.24}$$

$$\int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} q(\mathbf{X}|\Theta')^{1-\sigma} dX = \left[\prod_{d=1}^{D} \frac{\alpha_d \beta_d}{\alpha_d+\beta_d}\right]^{\sigma} \left[\prod_{d=1}^{D} \frac{\alpha'_d \beta'_d}{\alpha'_d+\beta'_d}\right]^{1-\sigma} \times \prod_{d=1}^{D} \frac{(\sigma\alpha_d+\sigma\beta_d)(1-\sigma)(\alpha'_d+\beta'_d)}{\sigma(\alpha_d\beta_d)(1-\sigma)(\alpha'_d\beta'_d)}. \tag{7.25}$$

The other kernel is the Jensen–Shannon (JS) kernel, generated according to the Jensen–Shannon divergence [47], which is given by:

$$JS_{\omega}(p(\mathbf{X}|\Theta), q(\mathbf{X}|\Theta')) = H\left[\omega\, p(\mathbf{X}|\Theta) + (1-\omega) q(\mathbf{X}|\Theta')\right] - \omega\, H\left[p(\mathbf{X}|\Theta)\right] - (1-\omega)\, H\left[q(\mathbf{X}|\Theta')\right], \tag{7.26}$$

where $\omega$ is a parameter, and $H[p(\mathbf{X}|\Theta)] = -\int p(\mathbf{X}|\Theta) \log p(\mathbf{X}|\Theta)\, d\mathbf{X}$ is the Shannon entropy.

## 7.5 Experimental Results

### *7.5.1 Methodology and Performance Measures*

In this section we present, analyze, and discuss the performance of the proposed approach through a set of experiments that we have performed. The applications concern activity analysis and action recognition in video surveillance scenes. There are two main goals in our experiments. The first goal is to investigate and compare the different generative kernels based on the exponential approximation to the Dirichlet compound multinomial (EDCM), and the exponential approximation to the generalized Dirichlet multinomial (EGDM) as we have proposed in the previous section. The second goal is to compare the proposed hybrid generative discriminative approaches to their generative counterparts and to the widely used discriminative approaches. We present experimental evidence that our generative kernels based on the exponential distributions perform better than their counterparts,

generated from the DCM and GDM, and better than the mixture models from which they are generated. In all our experiments we have used the one-vs-all training approach and the values for all design parameters were obtained by performing five fold cross-validation and all experimental results were averaged over 20 runs.

Each object (i.e., video sequence or frame), in the considered datasets was represented as a bag-of-features. For video sequences, we start by detecting the spatio-temporal interest points (STIP) where the local neighborhood has a significant variations in both spatial and temporal domains [44]. Then, we used 3D SIFT descriptor [69] that has shown to accurately capture the spatio-temporal nature of the video data. Moreover, images (frames) were encoded as a bag of scale-invariant feature transform (SIFT) feature vectors [49] (i.e., we used SIFT to detect the key points and extract their descriptors). Thus, the generative stage is done by fitting directly the EDCM, or EGDM, model to the feature vectors extracted from the images or videos. Consequently, each object in our datasets is represented by a finite mixture model of distributions. The discriminative stage, on the other hand, is represented by computing the Fisher, probability product or information-divergence based, kernel between each of these mixture models giving us kernel matrices to feed the SVM classifier.

To evaluate obtained results, we have used two measures which are usually used to evaluate classification models, as follows:

- *Accuracy:* is a performance metric that gives an indication of overall well classified elements and is defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}},$$

  where TP, FP, TN, FN denote: true positive, false positive, true negative, and false negative, respectively.
- *AUC:* Area under the (receiver operating characteristics (ROC) curve, which measures the performance of the multi-class classification problem.

## 7.5.2 *Classification Using Generative/Discriminative Approach*

### 7.5.2.1 Classification of Traffic Scene Based on Density

Recently, video monitoring and surveillance systems have been widely used in traffic management. Due to the high number of cameras in use, developing intelligent systems that extract useful information such as traffic density and vehicle classification information from traffic surveillance systems has become a significant and challenging task. The importance of knowing the traffic density of the roads is justified by its use for signal control and effective traffic management, time estimation of reaching from one location to another, and recommendation of different route alternatives [61]. For traffic scene classification, we used the UCSD

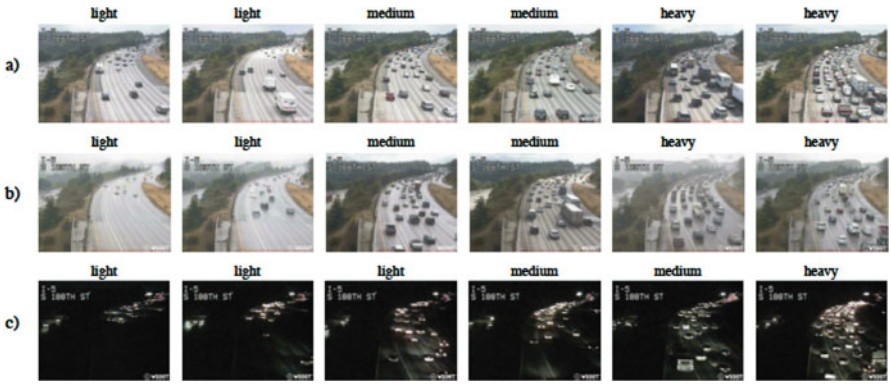



**Fig. 7.1** Classification of traffic congestion in variable lighting conditions: (**a**) sunny, (**b**) overcast, and (**c**) nighttime

traffic video dataset [15] to categorize videos based on the density of traffic. The dataset has been partitioned into three classes corresponding to light, medium, and heavy highway traffic congestion with a variety of weather conditions (see sample frames in Fig. 7.1). It consists of 254 video sequences of highway traffic in Seattle, collected from a single stationary traffic camera over 2 days. Each video contains between 42 and 52 frames of size 320 × 240, which, following common practice [15], are normalized and downsized to 48 × 48 grayscale images.

#### 7.5.2.2 Detection of Unusual Events in Traffic Flows

For detecting unusual events in busy public scenes, we used the junction dataset [50] by Queen Mary University of London (QMUL). The length of the video is approximately 1 h (9000 frames) captured with 360 × 288 frame size at 25 fps. The traffic is regulated by traffic lights and dominated with three traffic flows (sample frames are shown in Fig. 7.3). Following the practice in [50], the video was segmented into non-overlapping clips of 50 frames long each, resulting in 1800 clips. Each clip was manually labeled into different event classes as follows:

- Vertical traffic flow (1078 clips)
- Rightward traffic flow (323 clips)
- Leftward traffic flow (355 clips)
- *Unusual:* Illegal u-turns (29 clips)
- *Unusual:* Emergency vehicles using an improper lane of traffic (3 clips)
- *Unusual:* Traffic interruptions by fire engines (12 clips)

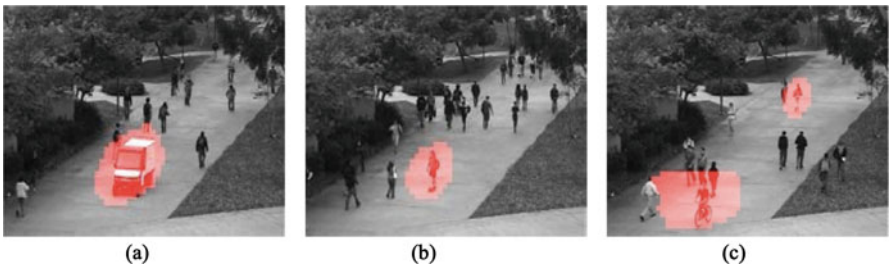



**Fig. 7.2** Examples of abnormal events in UCSD ped1 due to the circulation of non-pedestrian entities in the walkways: (**a**) small cart, (**b**) skater, and (**c**) bike

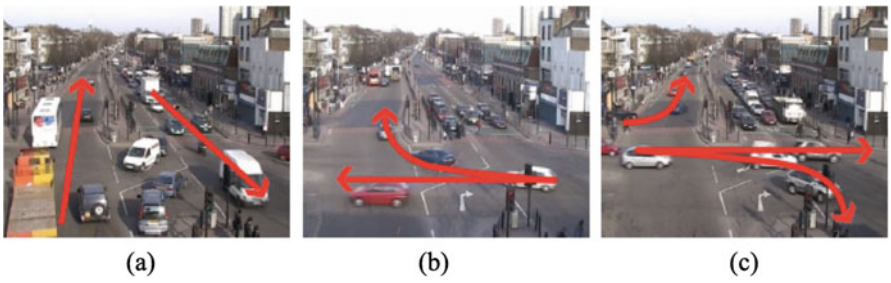



**Fig. 7.3** A traffic scene dominated by three different traffic flows (arranged in order): (**a**) vertical flow, (**b**) rightward flow, (**c**) leftward flow

#### 7.5.2.3 Anomaly Detection in Crowded Scenes

The analysis of densely crowded environments has recently received much interest within computer vision (e.g., [2, 42]). Most of the efforts are motivated by the ubiquity of surveillance cameras, the challenges of crowd modeling, and the importance of crowd monitoring for various applications. In this context, the goal is to detect deviations from normal crowd behavior (i.e., anomalous or abnormal events). In this experiment, our concern is to detect anomalies in the video surveillance sequences (at frame level) of the public UCSD Ped1 and Ped2 dataset [53].

This dataset was acquired with a stationary camera mounted at an elevation, overlooking pedestrian walkways with variable crowd density ranging from sparse to very crowded. In the normal setting, the video contains only pedestrians. Abnormal events are due to either the circulation of non-pedestrian entities in the walkways or anomalous pedestrian motion patterns. Commonly occurring anomalies examples include bikers, skaters, and small carts (see samples of abnormal frames in Fig. 7.2). The data was split into two subsets each with a different camera viewpoint. The video footage recorded from each scene was split into various clips of around 200 frames. For each clip, the groundtruth annotation includes a binary flag per frame, indicating whether an anomaly is present in that frame.

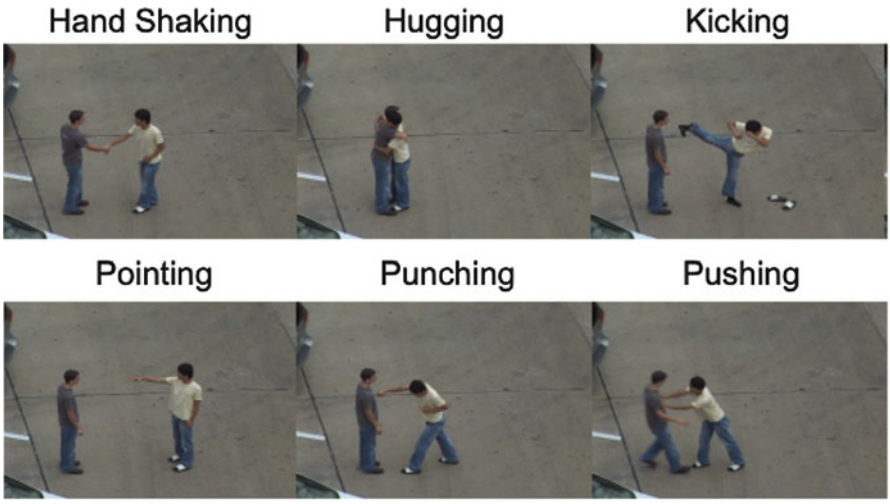



**Fig. 7.4** Types of activities in the interaction challenge

#### 7.5.2.4 Human–Human Interaction Recognition

Recognizing human interaction from videos is important in computer vision for several applications. Indeed, semantic analysis of activity videos enables construction of various vision-based intelligent systems, including smart surveillance systems, intelligent robots, action-based human–computer interfaces, and monitoring systems for children and elderly persons [67]. It is a challenging problem partially due to the lack of discriminability and expressiveness in single feature based representation, especially in motion ambiguity and partial occlusion [24]. In this experiment, we aim at recognizing multiple high level activities from a video composing several actors performing multiple interactions. We considered the UT interaction dataset [67], which contains six types of two-person interactions, shake-hands, point, hug, push, kick, and punch (see Fig. 7.4).

#### 7.5.2.5 Human Action Recognition in Video Sequences

In this experiment, we used one challenging video database, namely the KTH human action database [45], containing six types of human actions (walking, jogging, running, boxing, hand waving, and hand clapping) performed several times by 25 subjects in four different scenarios: outdoors s1, outdoors with scale variation s2, outdoors with different clothes s3 and indoors s4 (see Fig. 7.5). It contains 2391 sequences, all were taken over homogeneous backgrounds with a static camera with 25 fps frame rate. All sequences were downsampled to the spatial resolution of $160 \times 120$ pixels and have a length of 4 s on average.

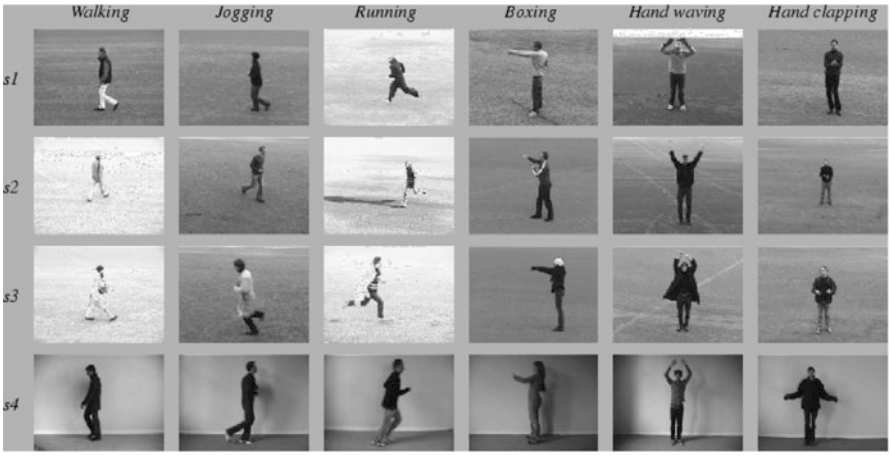



**Fig. 7.5** Different types of outdoor human actions in KTH database

**Table 7.1** Performance comparison of different generative kernels in classifying traffic density scenes

| | Accuracy (%) | AUC |
|---|---|---|
| DCM + Fisher Kernel | 78.40 | 0.7552 |
| EDCM + Fisher Kernel | 92.31 | 0.9175 |
| EDCM + Bhattacharyya Kernel | 84.00 | 0.7950 |
| EDCM + Kullback–Leibler | 84.62 | 0.7600 |
| EDCM + Bhattacharyya Kernel (MC) | 85.13 | 0.7777 |
| EDCM + Kullback–Leibler (MC) | 85.50 | 0.7878 |
| EDCM + Rényi Kernel | 80.77 | 0.7530 |
| EDCM + Jensen–Shannon | 80.46 | 0.7389 |
| GDM + Fisher Kernel | 92.31 | 0.9327 |
| EGDM + Fisher Kernel | **96.15** | **0.9722** |
| EGDM + Bhattacharyya Kernel (MC) | 87.90 | 0.8723 |
| EGDM + Kullback–Leibler (MC) | 86.50 | 0.8344 |
| EGDM + Rényi Kernel | 88.00 | 0.7389 |
| EGDM+ Jensen–Shannon | 84.00 | 0.7083 |

#### 7.5.2.6 Results and Discussion: Comparing Different Generative Kernels

The quantitative performances obtained based on the ground truth and in terms of AUC and accuracy metrics when deploying different kernels generated from EDCM and EGDM are presented in Tables 7.1, 7.2, 7.3, 7.4, and 7.5, where **bold** font indicates **best result** obtained for each dataset. According to these tables, we can see clearly that the SVM based on Fisher kernel generated from the EGDM mixture model provides the best results for all the considered datasets. Moreover, all the kernels generated from EGDM perform slightly generally better than those

**Table 7.2** Performance comparison of different generative kernels in detecting unusual event in traffic flows

| | Accuracy (%) | AUC |
|---|---|---|
| DCM + Fisher Kernel | 65.96 | 0.6422 |
| EDCM + Fisher Kernel | 76.44 | 0.6972 |
| EDCM + Bhattacharyya Kernel | 64.91 | 0.5825 |
| EDCM + Kullback–Leibler | 66.90 | 0.6479 |
| EDCM + Bhattacharyya Kernel (MC) | 65.44 | 0.6032 |
| EDCM + Kullback–Leibler(MC) | 66.80 | 0.6411 |
| EDCM + Rényi Kernel | 70.18 | 0.6632 |
| EDCM + Jensen–Shannon | 67.92 | 0.6655 |
| GDM + Fisher Kernel | 73.33 | 0.7101 |
| EGDM + Fisher Kernel | **79.66** | **0.7632** |
| EGDM + Bhattacharyya Kernel (MC) | 75.87 | 0.7323 |
| EGDM + Kullback–Leibler (MC) | 75.55 | 0.7277 |
| EGDM + Rényi Kernel | 76.66 | 0.7473 |
| EGDM+ Jensen–Shannon | 78.87 | 0.7537 |

**Table 7.3** Performance comparison of different generative kernels in detecting anomaly in crowded scenes

| | Ped1 | | Ped2 | |
|---|---|---|---|---|
| | Accuracy (%) | AUC | Accuracy (%) | AUC |
| DCM + Fisher Kernel | 74.62 | 0.7245 | 80.35 | 0.7207 |
| EDCM + Fisher Kernel | 75.85 | 0.7444 | 82.59 | 0.7544 |
| EDCM + Bhattacharyya Kernel | 76.53 | 0.6979 | 87.57 | 0.8007 |
| EDCM + Kullback–Leibler | 77.36 | 0.6999 | 84.83 | 0.7828 |
| EDCM + Bhattacharyya Kernel (MC) | 75.90 | 0.7212 | 84.55 | 0.7765 |
| EDCM + Kullback–Leibler (MC) | 76.00 | 0.7200 | 83.88 | 0.7643 |
| EDCM + Rényi Kernel | 78.47 | 0.7167 | 84.33 | 0.8134 |
| EDCM + Jensen–Shannon | 78.56 | 0.7038 | 83.58 | 0.8184 |
| GDM + Fisher Kernel | 82.72 | 0.7076 | 86.31 | 0.8999 |
| EGDM + Fisher Kernel | **86.86** | **0.7513** | **88.55** | **0.9000** |
| EGDM + Bhattacharyya Kernel (MC) | 85.00 | 0.7244 | 84.30 | 0.8187 |
| EGDM + Kullback–Leibler (MC) | 85.50 | 0.7444 | 84.50 | 0.8329 |
| EGDM + Rényi Kernel | 85.33 | 0.7209 | 84.83 | 0.8299 |
| EGDM+ Jensen–Shannon | 85.28 | 0.7402 | 85.80 | 0.8234 |

generated from EDCM. This can be interpreted by the flexibility of this recently proposed finite mixture model and it is ability to fit better the extracted feature vectors.

For classifying traffic scenes based on density (Table 7.1), the result obtained using Fisher kernel based on EDCM is, indeed, similar to the one reached by the SVM approach with GDM Fisher kernel. Table 7.2 displays the classification results for the QMUL dataset. For this dataset, the Rényi kernel generated from

**Table 7.4** Performance comparison of different generative kernels in recognizing human–human interaction

| | Accuracy (%) | AUC |
|---|---|---|
| DCM + Fisher Kernel | 79.63 | 0.7000 |
| EDCM + Fisher Kernel | 80.00 | 0.7544 |
| EDCM + Bhattacharyya Kernel | 80.00 | 0.7273 |
| EDCM + Kullback–Leibler | 79.20 | 0.7143 |
| EDCM + Bhattacharyya Kernel (MC) | 79.50 | 0.7256 |
| EDCM + Kullback–Leibler (MC) | 79.00 | 0.7173 |
| EDCM + Rényi Kernel | 81.67 | 0.8700 |
| EDCM + Jensen–Shannon | 83.33 | 0.8091 |
| GDM + Fisher Kernel | 85.00 | 0.8166 |
| EGDM + Fisher Kernel | **88.35** | **0.8767** |
| EGDM + Bhattacharyya Kernel (MC) | 82.88 | 0.8093 |
| EGDM + Kullback–Leibler (MC) | 83.40 | 0.8152 |
| EGDM + Rényi Kernel | 81.67 | 0.7800 |
| EGDM+ Jensen–Shannon | 83.33 | 0.8091 |

**Table 7.5** Performance comparison of different generative kernels in recognizing human action in video sequences

| | Accuracy (%) | AUC |
|---|---|---|
| DCM + Fisher Kernel | 69.29 | 0.6257 |
| EDCM + Fisher Kernel | 74.07 | 0.6326 |
| EDCM + Bhattacharyya Kernel | 72.62 | 0.6858 |
| EDCM + Kullback–Leibler | 78.24 | 0.7290 |
| EDCM + Bhattacharyya Kernel (MC) | 72.62 | 0.6858 |
| EDCM + Kullback–Leibler (MC) | 76.50 | 0.7233 |
| EDCM + Rényi Kernel | 76.21 | 0.7014 |
| EDCM + Jensen–Shannon | 76.93 | 0.6905 |
| GDM + Fisher Kernel | 70.53 | 0.6227 |
| EGDM + Fisher Kernel | **79.97** | **0.7500** |
| EGDM + Bhattacharyya Kernel (MC) | 76.00 | 0.6710 |
| EGDM + Kullback–Leibler (MC) | 76.49 | 0.6986 |
| EGDM + Rényi Kernel | 75.87 | 0.6863 |
| EGDM+ Jensen–Shannon | 78.20 | 0.6903 |

EGDM outperforms the other kernels with a very close result to the one reached by the SVM approach based on EGDM Fisher kernel. The classification results related to UCSD ped1 and ped2 sets are summarized in Table 7.3. For both datasets, we can notice that the different kernels generated from each distribution perform comparably. It is noteworthy that the Rényi and Jensen–Shannon kernels based on EDCM perform slightly better on UCSD ped1 than other kernels including the Fisher kernel generated from the same. For UCSD ped2, considering the accuracy, the Bhattacharyya kernel generated from the EDCM clearly outperforms the other

**Table 7.6** Comparison results based on accuracy of our method to the state of the art for UCSD traffic dataset

| Approach | Accuracy |
|---|---|
| Linear dynamical systems (LDS) [68] | 87.50% |
| Compressive sensing LDS (CS-LDS) [68] | 89.06% |
| Probabilistic kernels (KL-SVM) [15] | 95.00% |
| Spatio-temporal orientation analysis (SOA) [77] | 95.20% |
| NLSSA-RBF kernel [3] | 94.20% |
| DNLSSA-RBF kernel [3] | 94.50% |
| Proposed method: EGDM + Fisher kernel | **96.15%** |

**Table 7.7** Comparison results based on AUC of our method to the state of the art for UCSD ped1 dataset

| Approach | AUC |
|---|---|
| Social force [57] | 0.1790 |
| MPPCA [41] | 0.2050 |
| SF-MPPCA [41] | 0.2130 |
| Mixture of dynamic texture (MDT) [53] | 0.4410 |
| Sparse [19] | 0.4610 |
| Sparse + LSDS [19] | 0.4870 |
| Proposed method 1: EGDM + Fisher kernel | **0.7513** |
| Proposed method 2: EDCM + Fisher kernel | **0.7444** |

kernels despite the fact that the Fisher kernel based on EGDM performs generally slightly better. The same conclusion is valid for the UT interaction and KTH human actions datasets as shown in Tables 7.4 and 7.5, respectively. For these two datasets we can notice that the different kernels generated from each generative model perform slightly similarly, and the kernels generated from EGDM outperform the ones generated from EDCM.

Furthermore, we compared the obtained results with other methods from the literature for UCSD traffic and UCSD ped1 datasets, in Tables 7.6 and 7.7, respectively. According to the considered measures, i.e., accuracy and AUC, our approach achieves competitive results to the state of the art as we can notice that the proposed methods attain the highest metrics.

### *7.5.3 Results Using Fully Generative Models and Discriminative Techniques*

In this section, the experiments are conducted based solely on our generative models by fitting different models to the local descriptors directly. The results of this experiment are shown in Table 7.8. According to the results, it is clear that hybrid models improve the classification accuracy compared to their fully generative counterparts. For instance, the accuracy of classifying the QMUL traffic dataset by fitting EDCM and EGDM directly to the descriptor is 66.43% and 75.19%, respectively, which have been improved to 76.44% and 79.66% when using SVM

**Table 7.8** Results when deploying directly the generative models to the different datasets

| | DCM | EDCM | GDM | EGDM |
|---|---|---|---|---|
| UCSD traffic | 76.43 | 78.14 | 80.18 | 88.98 |
| QMUL traffic | 63.22 | 66.43 | 70.50 | 75.19 |
| UCSD ped1 | 70.47 | 73.94 | 73.59 | 79.77 |
| UCSD ped2 | 65.72 | 68.51 | 66.58 | 71.97 |
| UT interaction | 74.31 | 77.30 | 76.60 | 80.00 |
| KTH actions | 67.25 | 70.99 | 68.26 | 74.80 |

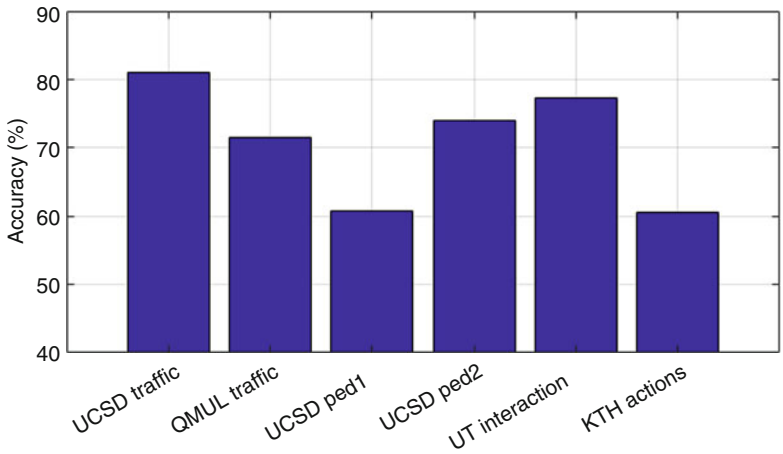



**Fig. 7.6** $k$-nearest neighbors K-NN accuracy for all tested datasets ($k = 5$)

with a kernel based on EDCM and EGDM Fisher score. This can be explained by the main idea of the Fisher kernel which is to exploit the geometric structure of the statistical manifold by mapping a given individual sequence of vectors into a single feature vector, defined in the gradient log-likelihood space.

Note that the pure discriminative approach, i.e., SVM with classic kernels, cannot be applied using the previous approach, since each video/frame is represented now by a set of vectors. Thus, we considered the so-called bag-of-features (BoF) approach based on the frequency of visual words [20], where each object can be represented as a histogram of frequent features. In our experiments, each dataset was randomly splitted into 60:40 to construct the visual vocabulary and representation. The classification results using two widely used discriminative approaches, namely $k$-nearest neighbor and SVM with a classic kernel (i.e., radial basis function (RBF)), are shown in Figs. 7.6 and 7.7, respectively. According to the results, kernels generated from the generative models have provided good and promising results as compared to state-of-the-art discriminative techniques.

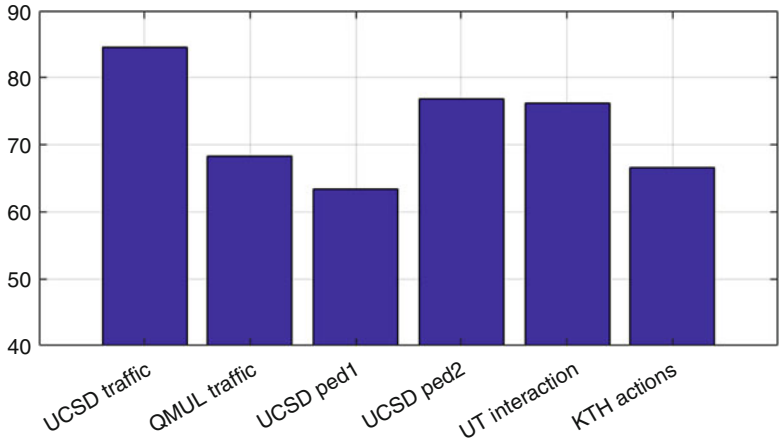



**Fig. 7.7** SVM accuracy for all tested datasets considering a classic kernel (RBF)

## 7.6 Conclusion

In this work, we have developed a hybrid generative discriminative framework to tackle the problem of modeling and classifying count data. In particular, we derived probabilistic kernels from the exponential approximation to both the Dirichlet compound multinomial (EDCM) and the generalized Dirichlet multinomial (EGDM). The proposed approaches are motivated by the flexibility and efficiency of these generative models, as well as the advantages of both SVMs and finite mixture models. Our experiments concerned classifying video sequences and frames from surveillance scenes for different purposes. The obtained results have shown that the developed kernels are promising and could be then applied for other classification problems that involve count vectors. According to the results, we can say also that the EGDM has better modeling capabilities than EDCM finite mixtures, and both outperform their corresponding models.

## Appendix 1: Proof of Eq. (7.18)

It is possible to compute the Bhattacharyya kernel in closed form for densities that belong to the exponential family of distributions, as:

$$\mathcal{K}_{BH} = \exp\left[\frac{1}{2}\Phi(\Theta) + \frac{1}{2}\Phi(\Theta') - \Phi\left(\frac{1}{2}\Theta + \frac{1}{2}\Theta'\right)\right]. \tag{7.27}$$

In the case of EDCM, we can show that:

$$\begin{aligned}\mathcal{K}_{BH} &= \exp\Bigg[\frac{1}{2}\left(\log\Gamma(s) - \log\Gamma(s+n)\right) + \frac{1}{2}\left(\log\Gamma(s') - \log\Gamma(s'+n)\right) \\ &\quad - \log\Gamma(\frac{1}{2}s + \frac{1}{2}s') + \log\Gamma(\frac{1}{2}(s+n+s'+n))\Bigg] \\ &= \frac{\Gamma(\frac{1}{2}(s+n+s'+n))\sqrt{\Gamma(s)\Gamma(s')}}{\Gamma(\frac{1}{2}s + \frac{1}{2}s')\sqrt{\Gamma(s+n)\Gamma(s'+n)}}. \end{aligned} \tag{7.28}$$

## Appendix 2: Proof of Eq. (7.22)

The KL-divergence between two exponential distributions is given by [38].

$$KL(p(X|\Theta),\, p(X|\Theta')) = \Phi(\Theta) - \Phi(\Theta') + [G(\Theta) - G(\Theta')]^{tr} E_\Theta[T(X)], \tag{7.29}$$

where $E_\Theta$ is the expectation with respect to $p(X|\Theta)$. Moreover, we have the following [11]:

$$E_\Theta[T(X)] = -\Phi'(\Theta). \tag{7.30}$$

Thus, according to Eq. (7.6), we have:

$$E_\Theta\Big[\sum_{d=1}^{D} I(x_d \geq 1)\Big] = -\frac{\partial\Phi(\Theta)}{\partial\alpha_d} = \Psi(\sum_{d=1}^{D}\alpha_d + n) - \Psi(\sum_{d=1}^{D}\alpha_d), \tag{7.31}$$

where $n = \sum_{d=1}^{D} x_d$, and $\Psi(.)$ is the digamma function. By substituting the previous two equations into Eq. (7.29), we obtain:

$$\begin{aligned} &KL(p(X|\Theta), q(X|\Theta')) \\ &\quad = \log(\Gamma(s)) - \log\left(\Gamma(s')\right) - \log(\Gamma(s+n)) + \log\left(\Gamma(s'+n)\right) \\ &\qquad + \sum_{d=1}^{D}\left(\Psi(\sum_{d=1}^{D}\alpha_d + n) - \Psi(\sum_{d=1}^{D}\alpha_d)\right)(\alpha_d - \alpha'_d) \\ &\quad = \log\left[\frac{\Gamma(s)\Gamma\left(s'+n\right)}{\Gamma(s')\Gamma(s+n)}\right] + \sum_{d=1}^{D}(\Psi(s+n) - \Psi(s))(\alpha_d - \alpha'_d). \end{aligned} \tag{7.32}$$

## Appendix 3: Proof of Eqs. (7.24) and (7.25)

Reńyi kernel is given by:

$$\mathcal{K}_R(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) = \Bigg[\int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} p'(\mathbf{X}|\Theta')^{1-\sigma} dX \times \int_0^{+\infty} p'(\mathbf{X}|\Theta')^{\sigma} p(\mathbf{X}|\Theta)^{1-\sigma} dX\Bigg]^{A/(1-\sigma)}. \tag{7.33}$$

In case of EDCM, we can show that:

$$\begin{aligned}
&\int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} p'(\mathbf{X}|\Theta')^{1-\sigma} dX \\
&= \Big[\frac{\Gamma(\sum_{d=1}^{D}\alpha_d)}{\Gamma(\sum_{d=1}^{D}\alpha_d + n)}\Big]^{\sigma}\Big[\frac{\Gamma(\sum_{d=1}^{D}\alpha'_d)}{\Gamma(\sum_{d=1}^{D}\alpha'_d + \sum_{d=1}^{d} x_d)}\Big]^{1-\sigma} \\
&\times \int_0^{+\infty}\Big[n!\prod_{d=1}^{D}\frac{\alpha_d}{x_d}\Big]^{\sigma}\Big[n!\prod_{d=1}^{D}\frac{\alpha'_d}{x_d}\Big]^{1-\sigma} dX \\
&= \Big[\frac{\Gamma(s)}{\Gamma(s+n)}\Big]^{\sigma}\Big[\frac{\Gamma(s')}{\Gamma(s'+n)}\Big]^{1-\sigma} \\
&\times \int_0^{+\infty}\frac{n!}{\prod_{d=1}^{D} x_d}\prod_{d=1}^{D}\alpha_d^{\sigma} dX \times \int_0^{+\infty}\frac{n!}{\prod_{d=1}^{D} x_d}\prod_{d=1}^{D}{\alpha'_d}^{1-\sigma} dX.
\end{aligned} \tag{7.34}$$

We have the PDF of an EDCM distribution that integrates to one which gives:

$$\int_0^{+\infty}\frac{n!}{\prod_{d=1}^{D} x_d}\prod_{d=1}^{D}\alpha_d dX = \frac{\Gamma(s+n)}{\Gamma(s)}. \tag{7.35}$$

By substituting Eq. (7.35) into Eq. (7.34), we obtain:

$$\begin{aligned}
&\int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} p'(\mathbf{X}|\Theta')^{1-\sigma} dX \\
&= \Big[\frac{\Gamma(s)}{\Gamma(s+n)}\Big]^{\sigma}\Big[\frac{\Gamma(s')}{\Gamma(s'+n)}\Big]^{1-\sigma} \times \frac{\Gamma(\sigma s+n)\Gamma((1-\sigma)s'+n)}{\Gamma(\sigma s)\Gamma((1-\sigma)s')}.
\end{aligned} \tag{7.36}$$

Similarly, in case of EGDM, we can show that:

$$\int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} p'(\mathbf{X}|\Theta')^{1-\sigma} dX = \left[\prod_{d=1}^{D} \frac{\alpha_d \beta_d}{\alpha_d + \beta_d}\right]^{\sigma} \left[\prod_{d=1}^{D} \frac{\alpha'_d \beta'_d}{\alpha'_d + \beta'_d}\right]^{1-\sigma} \times \int_0^{+\infty} \left[\frac{n!}{\prod_{d=1}^{D} x_d} \prod_{d=1}^{D} \frac{\Gamma(z_d)}{\Gamma(x_d + z_d)}\right]^{\sigma} dX \times \int_0^{+\infty} \left[\frac{n!}{\prod_{d=1}^{D} x_d} \prod_{d=1}^{D} \frac{\Gamma(z_d)}{\Gamma(x_d + z_d)}\right]^{1-\sigma} dX. \tag{7.37}$$

Then considering that the PDF of an EGDM distribution integrates to one, we have:

$$\int_0^{+\infty} \frac{n!}{\prod_{d=1}^{D} x_d} \prod_{d=1}^{D} \frac{\Gamma(z_d)}{\Gamma(x_d + z_d)} dX = \prod_{d=1}^{D} \frac{\alpha_d + \beta_d}{\alpha_d \beta_d} \tag{7.38}$$

and by substituting Eq. (7.38) into Eq. (7.37), we obtain:

$$\int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} p'(\mathbf{X}|\Theta')^{1-\sigma} dX = \left[\prod_{d=1}^{D} \frac{\alpha_d \beta_d}{\alpha_d + \beta_d}\right]^{\sigma} \left[\prod_{d=1}^{D} \frac{\alpha'_d \beta'_d}{\alpha'_d + \beta'_d}\right]^{1-\sigma} \times \prod_{d=1}^{D} \frac{(\sigma\alpha_d + \sigma\beta_d)(1-\sigma)(\alpha'_d + \beta'_d)}{\sigma(\alpha_d \beta_d)(1-\sigma)(\alpha'_d \beta'_d)}. \tag{7.39}$$

## References


1. Agarwal, A., Daumé III, H.: Generative kernels for exponential families. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 85–92 (2011)
2. Ali, S., Shah, M.: Floor fields for tracking in high density crowd scenes. In: European Conference on Computer Vision, pp. 1–14. Springer, Berlin (2008)
3. Baktashmotlagh, M., Harandi, M., Lovell, B.C., Salzmann, M.: Discriminative non-linear stationary subspace analysis for video classification. IEEE Trans. Pattern Anal. Mach. Intell. **36**(12), 2353–2366 (2014)
4. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman divergences. J. Mach. Learn. Res. **6**(Oct), 1705–1749 (2005)
5. Bdiri, T., Bouguila, N.: Bayesian learning of inverted Dirichlet mixtures for SVM kernels generation. Neural Comput. Appl. **23**(5), 1443–1458 (2013)
6. Bishop, C.: Pattern Recognition and Machine Learning. Information Science and Statistics. Springer, New York (2006)

7. Bishop, C.M., et al.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
8. Bouguila, N.: Clustering of count data using generalized Dirichlet multinomial distributions. IEEE Trans. Knowl. Data Eng. **20**(4), 462–474 (2008)
9. Bouguila, N.: Hybrid generative/discriminative approaches for proportional data modeling and classification. IEEE Trans. Knowl. Data Eng. **24**(12), 2184–2202 (2012)
10. Bouguila, N., Amayri, O.: A discrete mixture-based kernel for SVMs: application to spam and image categorization. Inf. Process. Manag. **45**(6), 631–642 (2009)
11. Brown, L.D.: Fundamentals of statistical exponential families: with applications in statistical decision theory. Institute of Mathematical Statistics, Hayward (1986)
12. Burges, C.J.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. **2**(2), 121–167 (1998)
13. Caballero, K.L., Barajas, J., Akella, R.: The generalized Dirichlet distribution in enhanced topic detection. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 773–782. ACM, New York (2012)
14. Campbell, W.M., Sturim, D.E., Reynolds, D.A.: Support vector machines using GMM super vectors for speaker verification. IEEE Signal Process. Lett. **13**(5), 308–311 (2006)
15. Chan, A.B., Vasconcelos, N.: Probabilistic kernels for the classification of auto-regressive visual processes. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 846–851. IEEE, Piscataway (2005)
16. Chan, A.B., Vasconcelos, N., Moreno, P.J.: A family of probabilistic kernels based on information divergence. University of California, San Diego, CA, Technical Report. SVCL-TR-2004-1 (2004)
17. Christianini, N., Shawe-Taylor, J.: Support Vector Machines, vol. 93(443), pp. 935–948. Cambridge University Press, Cambridge (2000)
18. Church, K.W., Gale, W.A.: Poisson mixtures. Nat. Lang. Eng. **1**(2), 163–190 (1995)
19. Cong, Y., Yuan, J., Liu, J.: Abnormal event detection in crowded scenes using sparse representation. Pattern Recogn. **46**(7), 1851–1864 (2013)
20. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, (ECCV), Prague, vol. 1, pp. 1–2 (2004)
21. DasGupta, A.: The exponential family and statistical applications. In: Probability for Statistics and Machine Learning, pp. 583–612. Springer, New York (2011)
22. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B Methodol. **39**, 1–38 (1977)
23. Deng, J., Xu, X., Zhang, Z., Frühholz, S., Grandjean, D., Schuller, B.: Fisher kernels on phase-based features for speech emotion recognition. In: Dialogues with Social Robots, pp. 195–203. Springer, Singapore (2017)
24. Dong, Z., Kong, Y., Liu, C., Li, H., Jia, Y.: Recognizing human interaction by multiple features. In: The First Asian Conference on Pattern Recognition, pp. 77–81. IEEE, Piscataway (2011)
25. Elkan, C.: Clustering documents with an exponential-family approximation of the Dirichlet compound multinomial distribution. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 289–296. ACM, New York (2006)
26. Fayyad, U.M., Reina, C., Bradley, P.S.: Initialization of iterative refinement clustering algorithms. In: KDD, pp. 194–198 (1998)
27. Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE Trans. Pattern Anal. Mach. Intell. 24(3), 381–396 (2002)
28. Geary, D.: Mixture models: inference and applications to clustering. J. R. Stat. Soc. Ser. A **152**(1), 126–127 (1989)
29. Grauman, K., Darrell, T.: The pyramid match kernel: discriminative classification with sets of image features. In: Tenth IEEE International Conference on Computer Vision (ICCV), vol. 2, pp. 1458–1465. IEEE, Piscataway (2005)
30. Holub, A.D., Welling, M., Perona, P.: Hybrid generative-discriminative visual categorization. Int. J. Comput. Vis. **77**(1–3), 239–258 (2008)

31. Jaakkola, T., Haussler, D.: Exploiting generative models in discriminative classifiers. In: Advances in Neural Information Processing Systems, pp. 487–493. The MIT Press, Cambridge (1999)
32. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recogn. Lett. **31**(8), 651–666 (2010)
33. Jebara, T.: Images as bags of pixels. In: ICCV, pp. 265–272 (2003)
34. Jebara, T., Kondor, R.: Bhattacharyya and expected likelihood kernels. In: Learning Theory and Kernel Machines, pp. 57–71. Springer, Berlin (2003)
35. Jebara, T., Kondor, R., Howard, A.: Probability product kernels. J. Mach. Learn. Res. **5**(Jul), 819–844 (2004)
36. Jégou, H., Douze, M., Schmid, C.: On the burstiness of visual elements. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1169–1176. IEEE, Piscataway (2009)
37. Johnston, J., Hamerly, G.: Improving SimPoint accuracy for small simulation budgets with EDCM clustering. In: Workshop on Statistical and Machine Learning Approaches to ARchitectures and compilaTion (SMART08) (2008)
38. Kailath, T.: The divergence and Bhattacharyya distance measures in signal selection. IEEE Trans. Commun. Technol. **15**(1), 52–60 (1967)
39. Katz, S.M.: Distribution of content words and phrases in text and language modelling. Nat. Lang. Eng. **2**(1), 15–59 (1996)
40. Keerthi, S.S., Lin, C.J.: Asymptotic behaviors of support vector machines with Gaussian kernel. Neural Comput. **15**(7), 1667–1689 (2003)
41. Kim, J., Grauman, K.: Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2921–2928. IEEE, Piscataway (2009)
42. Kong, D., Gray, D., Tao, H.: Counting pedestrians in crowds using viewpoint invariant training. In: BMVC, vol. 1, p. 2. Citeseer (2005)
43. Kullback, S.: Information Theory and Statistics. Courier Corporation, Chelmsford, MA (1997)
44. Laptev, I.: On space-time interest points. Int. J. Comput. Vis. **64**(2–3), 107–123 (2005)
45. Laptev, I., Caputo, B., et al.: Recognizing human actions: a local SVM approach. In: null, pp. 32–36. IEEE, Piscataway (2004)
46. Li, Y., Shapiro, L., Bilmes, J.A.: A generative/discriminative learning algorithm for image classification. In: Tenth IEEE International Conference on Computer Vision (ICCV'05), vol. 2, pp. 1605–1612. IEEE, Piscataway (2005)
47. Lin, J.: Divergence measures based on the Shannon entropy. IEEE Trans. Inf. Theory **37**(1), 145–151 (1991)
48. Lin, H.T., Lin, C.J.: A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Neural Comput. **3**, 1–32 (2003)
49. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
50. Loy, C.C., Xiang, T., Gong, S.: Stream-based active unusual event detection. In: Asian Conference on Computer Vision, pp. 161–175. Springer, Berlin (2010)
51. Ma, Y., Guo, G.: Support Vector Machines Applications. Springer, Cham (2014)
52. Madsen, R.E., Kauchak, D., Elkan, C.: Modeling word burstiness using the Dirichlet distribution. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 545–552. ACM, New York (2005)
53. Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N.: Anomaly detection in crowded scenes. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1975–1981. IEEE, Piscataway (2010)
54. Margaritis, D., Thrun, S.: A Bayesian multiresolution independence test for continuous variables. In: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, pp. 346–353. Morgan Kaufmann Publishers, Burlington (2001)
55. McLachlan, G.J.: Mixture Models: Inference and Applications to Clustering. Marcel Dekker, New York (1988)

56. McLachlan, G., Krishnan, T.: The EM algorithm and extensions, vol. 382. Wiley, Hoboken (2007)
57. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 935–942. IEEE, Piscataway (2009)
58. Melnykov, V., Maitra, R., et al.: Finite mixture models and model-based clustering. Stat. Surv. **4**, 80–116 (2010)
59. Moguerza, J.M., Muñoz, A., et al.: Support vector machines with applications. Stat. Sci. **21**(3), 322–336 (2006)
60. Moreno, P.J., Ho, P.P., Vasconcelos, N.: A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. In: Advances in Neural Information Processing Systems, pp. 1385–1392 (2004)
61. Ozkurt, C., Camci, F.: Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks. Math. Comput. Appl. **14**(3), 187–196 (2009)
62. Penny, W.D.: Kullback-Leibler divergences of normal, gamma, Dirichlet and Wishart densities. Technical report, Wellcome Department of Cognitive Neurology (2001)
63. Pérez-Cruz, F.: Kullback-Leibler divergence estimation of continuous distributions. In: IEEE International Symposium on Information Theory (ISIT), pp. 1666–1670. IEEE, Piscataway (2008)
64. Raina, R., Shen, Y., Mccallum, A., Ng, A.Y.: Classification with hybrid generative/discriminative models. In: Advances in Neural Information Processing Systems, pp. 545–552 (2004)
65. Rényi, A., et al.: On measures of entropy and information. In: Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics. The Regents of the University of California, Oakland (1961)
66. Rubinstein, Y.D., Hastie, T., et al.: Discriminative vs informative learning. In: KDD, vol. 5, pp. 49–53 (1997)
67. Ryoo, M.S., Aggarwal, J.K.: Spatio-temporal relationship match: video structure comparison for recognition of complex human activities. In: IEEE 12th International Conference on Computer Vision (ICCV), pp. 1593–1600. IEEE, Piscataway (2009)
68. Sankaranarayanan, A.C., Turaga, P.K., Baraniuk, R.G., Chellappa, R.: Compressive acquisition of dynamic scenes. In: European Conference on Computer Vision, pp. 129–142. Springer, Berlin (2010)
69. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of the 15th ACM International Conference on Multimedia, pp. 357–360. ACM, New York (2007)
70. Shmilovici, A.: Support vector machines. In: Data Mining and Knowledge Discovery Handbook, pp. 231–247. Springer, New York (2010)
71. Titterington, D.M., Smith, A.F., Makov, U.E.: Statistical Analysis of Finite Mixture Distributions. Wiley, London (1985)
72. Tsuda, K., Akaho, S., Kawanabe, M., Müller, K.R.: Asymptotic properties of the fisher kernel. Neural Comput. **16**(1), 115–137 (2004)
73. Ueda, N., Nakano, R.: Deterministic annealing EM algorithm. Neural Netw. **11**(2), 271–282 (1998)
74. Van Der Maaten, L.: Learning discriminative fisher kernels. In: ICML, vol. 11, pp. 217–224 (2011)
75. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)
76. Vasconcelos, N., Ho, P., Moreno, P.: The Kullback-Leibler kernel as a framework for discriminant and localized representations for visual recognition. In: European Conference on Computer Vision, pp. 430–441. Springer, Berlin (2004)
77. Wang, Y., Mori, G.: Human action recognition by semilatent topic models. IEEE Trans. Pattern Anal. Mach. Intell. **31**(10), 1762–1774 (2009)
78. Wong, T.T.: Alternative prior assumptions for improving the performance of naïve Bayesian classifiers. Data Min. Knowl. Discov. **18**(2), 183–213 (2009)

79. Wong, T.T.: Generalized Dirichlet priors for naïve Bayesian classifiers with multinomial models in document classification. Data Min. Knowl. Discov. **28**(1), 123–144 (2014)
80. Zamzami, N., Bouguila, N.: Consumption behavior prediction using hierarchical Bayesian frameworks. In: First International Conference on Artificial Intelligence for Industries (AI4I), pp. 31–34. IEEE, Piscataway (2018)
81. Zamzami, N., Bouguila, N.: Hybrid generative discriminative approaches based on multinomial scaled Dirichlet mixture models. Appl. Intell., 1–18 (2019, in press)
82. Zamzami, N., Bouguila, N.: Model selection and application to high-dimensional count data clustering – via finite EDCM mixture models. Appl. Intell. **49**(4), 1467–1488 (2019)
83. Zamzami, N., Bouguila, N.: Sparse count data clustering using an exponential approximation to generalized Dirichlet multinomial distributions. Manuscript submitted to IEEE Transactions on Neural Networks and Learning Systems for review (2019)
84. Zhou, H., Lange, K.: MM algorithms for some discrete multivariate distributions. J. Comput. Graph. Stat. **19**(3), 645–665 (2010)

# Chapter 8
# Toward an Efficient Computation of Log-Likelihood Functions in Statistical Inference: Overdispersed Count Data Clustering


**Masoud Daghyani, Nuha Zamzami, and Nizar Bouguila**

**Abstract** This work presents an unsupervised learning algorithm, using the mesh method for computing the log-likelihood function. The multinomial Dirichlet distribution (MDD) is one of the widely used methods of modeling multicategorical count data with overdispersion. Recently, it has been shown that traditional numerical computation of the MDD log-likelihood function either results in instability or leads to long run times that make its use infeasible in case of large datasets. Thus, we propose to use the mesh algorithm that involves approximating the MDD log-likelihood function based on Bernoulli polynomials. Moreover, we extend the mesh algorithm approach for computing the log-likelihood function of a more flexible distribution, namely the multinomial generalized Dirichlet (MGD). We demonstrate the efficiency of this method in statistical inference, i.e., maximum likelihood estimation, for fitting finite mixture models based on MDD and MGD as efficient distributions for count data. Through a set of experiments, the proposed approach shows its merits in two real-world clustering problems, namely natural scenes categorization and facial expression recognition.

M. Daghyani (✉)
Department of Electrical and Computer Engineering (ECE), Concordia University, Montreal, QC, Canada
e-mail: m_daghya@encs.concordia.ca

N. Zamzami
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
e-mail: n_zamz@encs.concordia.ca

N. Bouguila
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca



N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6_8

## 8.1 Introduction

The analysis of categorical data, or count data, has huge number of applications in different fields such as machine learning, computer vision, big data analysis, pattern recognition (e.g., [1, 4, 8, 53]). The Poisson distribution is a primary distribution for modeling count data that has an equal mean and variance (equidispersion). However, in many practical situations this assumption is not valid as real data exhibits the phenomenon of overdispersion (i.e., the variance of the count variable exceeds its mean) [39]. Consider, for example, an image represented using bag-of-features approach where some of the features appear too many times and others appear less frequently or do not appear at all making the variance greater than the mean. For addressing this issue, the negative-binomial distribution has been widely used in high-throughput sequencing data [2, 18]. Moreover, multinomial (MN) distribution is another fundamental model in count data analysis which is useful for analyzing count proportions between multiple categories. However, in real data we encounter another phenomenon caused by dependencies or the similarity of responses of members of the same cluster. This leads to extra-multinomial variation [17], i.e., overdispersion with respect to the MN distribution. Thus, the multinomial distribution has been extended to the multinomial Dirichlet distribution (MDD) [34, 38] to model the overdispersion of the MN distribution. The MDD distribution has been used in various fields, including topic and magazine exposure modeling [20, 32, 43], word burstiness modeling [28], and language modeling [27]. The generalized Dirichlet multinomial distribution (MGD) [4] has been also considered for more flexible modeling of count data with overdispersion.

On the other hand, given the importance of count data, there have been numerous efforts for analyzing this kind of data using both supervised and unsupervised learning approaches. Finite mixture models [30] are among the most widely used techniques for unsupervised learning, i.e., clustering. In fact, many studies have proved that the adoption of discrete finite mixture models can have higher performance as compared with other common used approaches such as neural networks and decision trees [45]. Finite Mixtures are popular for modeling univariate and multivariate data [31]. Novel machine learning applications have changed the direction of the current research activities from working on mixtures for continues data to other types of data such as binary or integer-valued features [36] applied in text classification and binned and truncated multivariate data [7]. In the majority of the cases, the probability density functions (PDFs) of mixture models are considered to be Gaussian, which is not the best choice, specially where the partitions are clearly non-Gaussian [3]. For example, it has been shown that in the case of modeling discrete data in computer vision, Gaussian assumption is an inadequate choice, and most of the researchers use the multinomial distribution [36, 46].

To fit finite mixture models to the observed data, the most common method used is the expectation–maximization (EM) algorithm, for locating a maximum likelihood (ML) estimation of the mixture parameters [14]. Indeed, the log-likelihood function plays an essential role in such statistical inference method

[9, 35]. Thus, Yu and Shaw [52] proposed a novel parameterization of the MDD log-likelihood function based on a truncated series consisting of Bernoulli polynomials, and developed a mesh algorithm for computing this log-likelihood to extend its applicability. In this work, we first adopted this mesh algorithm for the computation of the MDD log-likelihood within a mixture model framework. Afterwards, we extended the approach in [52] to reparameterize the MGD log-likelihood function, along with utilizing the computation of log-likelihood using mesh algorithm in the parameter estimation process.

The remainder of this chapter is organized as follows: Section 8.2 reviews the parameterization of the MDD log-likelihood function that allows smooth transition from the overdispersed to the non-overdispersed case, and Sect. 8.3 discusses the MGD distribution and proposes its new parameterization. In Sect. 8.4, we describe the mesh algorithm for computing the log-likelihood functions. In Sect. 8.5 we discuss clustering using finite mixture models. Section 8.6 is devoted to experimental results, and we end the chapter in Sect. 8.7 with some concluding remarks.

## 8.2 Computing the MDD Log-Likelihood Function

In this section, we first discuss the multinomial Dirichlet distribution (MDD) in details. Later on, the approximation of the paired log-gamma difference of the log-likelihood function is explained.

### *8.2.1 The Multinomial Dirichlet Distribution*

We assume that $O = (O_1, \ldots, O_N)$ is a finite set of abstract objects and $e = (e_1, \ldots, e_K)$ the domain of some events. Also, we consider that the counts for each object $O_i$ are available as a co-occurrence vector $X_i = (X_{i1}, \ldots, X_{iK})$, where $X_{ij}$ refers to the number of times events $e_j$ happens in the object $O_i$. Hence, we represent the object by $\mathbf{X}$ supposing that $\mathbf{X}$ follows a multinomial distribution. However, using frequencies for obtaining the probabilities gives a weak estimation, due to the fact that the events are considered independent, which is not always true [10, 19]. Several attempts have been made to address this issue. Teevan and Karger discovered a model that fits discrete vectors in an exponential family of models [44]. Rennie et al. tried to reduce the impact of dependency by log-normalizing the counts [41].

Consider the observations (vector of counts) $\mathbf{X} = (X_1, \ldots, X_k)$, satisfying $\sum_{k=1}^{K} X_k = N$, and $\mathbf{P} = (P_1, \ldots, P_k)$ satisfying $\sum_{k=1}^{K} P_k = 1$, where $P_k$ is the probability of seeing the $k$th feature. The probability mass function (PMF) of K categories of the MN distribution having N-independent trials is given by:

$$\mathcal{M}(\mathbf{X}|\mathbf{P}) = \frac{N!}{\prod_{k=1}^{K} X_k!} \prod_{k=1}^{K} P_k^{X_k} \tag{8.1}$$

The Dirichlet distribution is a conjugate prior of $\mathbf{P}$. Suppose the random vector $\mathbf{P} = (P_1, \ldots, P_k)$ follows a Dirichlet distribution with parameters $\alpha = (\alpha_1, \ldots, \alpha_k)$, the joint density function is equal to [23]:

$$\mathcal{D}(P_1, \ldots, P_k) = \frac{\Gamma(A)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} P_k^{(\alpha_k - 1)} \tag{8.2}$$

where $\Gamma(\alpha_k)$ is the gamma function and $A = \sum_{k=1}^{K} \alpha_k$.

The marginal distribution of $\mathbf{X}$ is obtained by taking the integral of the product of the Dirichlet prior and the multinomial likelihood, with respect to the probabilities $\mathbf{P}$ [34]:

$$\mathcal{MDD}(\mathbf{X}|\boldsymbol{\alpha}) = \frac{N!}{\prod_{k=1}^{K} X_k!} \frac{\Gamma(A)}{\Gamma(A+N)} \prod_{k=1}^{K} \frac{\Gamma(\alpha_k + X_k)}{\Gamma(\alpha_k)} \tag{8.3}$$

We call this density the MDD (multinomial Dirichlet distribution) and the mean and variance of this distribution are given by [34]:

$$E(X_i) = \frac{|\mathbf{X}|\alpha_i}{\boldsymbol{\alpha}} \tag{8.4}$$

$$Var(X_i) = \frac{|\mathbf{X}|(|\mathbf{X}| - 1)\alpha_i(|\alpha| - \alpha_i)}{(|\alpha|)^2(|\alpha| + 1)} + \frac{|\mathbf{X}|\alpha_i}{\boldsymbol{\alpha}} \tag{8.5}$$

A special case of the MDD distribution with just two parameters ($K = 2$) is named the beta-binomial distribution that has been widely studied by [16, 24]. By taking the logarithm of both sides of the above equation, we achieve the log-likelihood function:

$$\begin{aligned} \ln \mathcal{L}(P, \psi; X) = &-(\ln \Gamma(1/\psi + N) - \ln \Gamma(1/\psi)) \\ &+ \sum_{k=1}^{K} \left( \ln \Gamma\left(1/\frac{\psi}{P_k} + X_k\right) - \ln \Gamma\left(1/\frac{\psi}{p_k}\right) \right) \end{aligned} \tag{8.6}$$

where $\psi = 1/A$ and $p = \psi\alpha$. In this work, we call $\psi$ the overdispersion parameter, which has a direct relation with the variance, and specifies the difference between a MDD distribution and its corresponding MN distribution in the same probability category. This formula has some deficiencies including that it is undefined for $\psi = 0$, also it is unstable when $\psi \to 0$, since each $\ln \Gamma$ term becomes very large, and the paired differences become relatively small which result in computation errors.

The mesh algorithm, proposed in [52], applies a new formula based on a truncated series consisting of Bernoulli polynomials to solve the instability problem without incurring long run times.

### 8.2.2 *Approximating the Paired Log-Gamma Difference in MDD Log-Likelihood Function*

As proposed in [52], we use the approximation of the paired log-gamma difference method and the properties of analytic functions as follows [6, 42]:

$$\ln\Gamma(1/x+y)-\ln\Gamma(1/x)\approx -y\ln x+D_m(x,y) \tag{8.7}$$

when $y$ is an integer, $|x|\min(|y-1|,|y|)<1$, $xy\le\delta$ and:

$$D_m(x,y)=\sum_{n=2}^{m}\frac{(-1)^n\phi_n(y)}{n(n-1)}x^{(n-1)} \tag{8.8}$$

where

$$\phi_n(y)=B_n(y)-B_n \tag{8.9}$$

is the old type Bernoulli polynomial [48], $B_n(y)$ and $B_n$ indicate the $n$th Bernoulli polynomial, and $n$th Bernoulli number ($B_n=B_n(0)$), respectively.

Using floating-point arithmetic, high order polynomials are hard to compute [13]. Hence, we cannot use too large $m$'s, because the error of each terms of $D_m(x,y)$ may be large, which eventually makes it inaccurate. Following Yu and Shaw [52], for computing the log-likelihood of the MDD distribution using the mesh method, we used $m=20$, as it makes $\phi_n(y)$ $(n\le m)$ numerically accurate. We also choose $\delta=0.2$ that results to an error bound of $\sim 1.30\times10^{-16}$, which is just little less than the machine epsilon double precision data type $\approx 2.22\times10^{-16}$.

Let $X^+$ be the vector of the non-zero elements in $X_i$, $P^+$ be a vector of the corresponding elements in $P$, and $K^+$ be the length of $X^+$, then Eq. (8.6) becomes:

$$\begin{aligned}\ln\mathcal{L}(P^+,\psi;X^+)=&-\Bigg(\underbrace{\ln\Gamma(1/\psi+N)-\ln\Gamma(1/\psi)}_{*}\Bigg)\\&+\sum_{k=1}^{K^+}\Bigg(\underbrace{\ln\Gamma\left(1/\frac{\psi}{P_k^+}+X_k^+\right)-\ln\Gamma\left(1/\frac{\psi}{P_k^+}\right)}_{**}\Bigg)\end{aligned} \tag{8.10}$$

As mentioned earlier, when condition $xy \leq \delta$ is met, we can use the approximation in Eq. (8.7) for all $K^+ + 1$ paired log-gamma differences in Eq. (8.10), as [52]:

$$\begin{aligned}\ln \mathcal{L}(P^+, \psi; X^+) &\approx -(-N \ln \psi + D_m(\psi, N)) \\ &\quad + \sum_{k=1}^{K^+} \left( - X_k^+ \ln \left( \frac{\psi}{P_k^+} \right) + D_m \left( \frac{\psi}{P_k^+}, X_k^+ \right) \right) \\ &= -D_m(\psi, N) + \sum_{k=1}^{K^+} \left( X_k^+ \ln p_k^+ + D_m \left( \frac{\psi}{P_k^+}, X_k^+ \right) \right)\end{aligned} \tag{8.11}$$

## 8.3 Computing the MGD Log-Likelihood

In this section we discuss the MGD distribution in sufficient details. Then, we propose the approximation of the paired log-gamma differences technique for computing the MGD log-likelihood function.

### *8.3.1 The Multinomial Generalized Dirichlet Distribution*

Despite its flexibility and its several interesting properties, such as the consistency of its estimates as a prior, the fact that it is conjugate to the multinomial, and its ease of use, the Dirichlet distribution has a very restrictive negative covariance structure that makes its use as a prior in the case of positively correlated data inappropriate. Another restriction of the Dirichlet distribution is that the variables with the same mean must have the same variance [22]. Recent works have shown that all these disadvantages can be handled by using the generalized Dirichlet distribution which has many convenient properties that make it more useful and practical, as a prior to the multinomial, than the Dirichlet in real-life applications [4, 50].

Define $\mathbf{X} = (X_1, \ldots, X_{K+1})$ as an overdispersed vector of counts of $K + 1$ events. Then, the composition of the generalized Dirichlet and the multinomial gives the multinomial generalized Dirichlet (MGD), as [4]:

$$\mathcal{MGD}(\mathbf{X}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{\Gamma(N+1)}{\prod_{k=1}^{K+1} \Gamma(X_k + 1)} \prod_{k=1}^{K} \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \prod_{k=1}^{K} \frac{\Gamma(\alpha'_k)\Gamma(\beta'_k)}{\Gamma(\alpha'_k + \beta'_k)} \tag{8.12}$$

where $\alpha'_k = \alpha_k + X_k$, and $\beta'_k = \beta_k + X_{k+1} + \cdots + X_{K+1}$ for $k = 1, \ldots, K$. Given that the generalized Dirichlet includes the Dirichlet as a special case, MGD is reduced to a MDD when $\beta_k = \alpha_{k+1} + \beta_{k+1}$.

The mean and the variance of the generalized Dirichlet distribution satisfy the following conditions [11, 49]:

$$E(P_k) = \frac{\alpha_k}{\alpha_k + \beta_k} \prod_{l=1}^{k-1} \frac{\beta_l}{\alpha_l + \beta_l}, \tag{8.13}$$

$$Var(P_k) = E(P_k)\left( \frac{\alpha_k + 1}{\alpha_k + \beta_k + 1} \prod_{l=1}^{k-1} \frac{\beta_l + 1}{\alpha_l + \beta_l + 1} - E(P_k) \right) \tag{8.14}$$

and the covariance between $P_{k1}$ and $P_{k2}$ is:

$$Cov(P_{k1}, P_{k2}) = E(P_{k2})\left( \frac{\alpha_{k1}}{\alpha_{k1} + \beta_{k1} + 1} \prod_{l=1}^{k_1-1} \frac{\beta_l + 1}{\alpha_l + \beta_l + 1} - E(P_{k1}) \right) \tag{8.15}$$

Like the Dirichlet, the generalized Dirichlet is conjugate to the multinomial distribution but has a more general covariance structure than the Dirichlet distribution and the variables with the same mean do not need to have the same variance [4, 5], thus, it is more practical to be used in modeling data with overdispersion. Moreover, it remains $K$ degrees of freedom, which makes it more flexible for several real-world applications [21].

### 8.3.2 *Approximating the Paired Log-Gamma Difference in MGD Log-Likelihood Function*

The first term on the right side of Eq. (8.12) does not depend on the parameters $\alpha$ and $\beta$. For the maximum likelihood estimation, we are not interested in the first term but in the product of the remaining two terms of the MGD likelihood function in Eq. (8.12):

$$\mathcal{L}(\alpha, \beta; X) = \prod_{k=1}^{K} \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \prod_{k=1}^{K} \frac{\Gamma(\alpha'_k)\Gamma(\beta'_k)}{\Gamma(\alpha'_k + \beta'_k)} \tag{8.16}$$

By taking the logarithm of both sides of Eq. (8.16), we get the log-likelihood function, as:

$$\begin{aligned} \ln \mathcal{L}(\alpha, \beta; X) = & \sum_{k=1}^{K} \left( \ln \Gamma(\alpha_k + \beta_k) - \ln \Gamma(\alpha_k) - \ln \Gamma(\beta_k) \right) \\ & + \sum_{k=1}^{K} - \left( \ln \Gamma(\alpha'_k + \beta'_k) - \ln \Gamma(\alpha'_k) - \ln \Gamma(\beta'_k) \right) \end{aligned} \tag{8.17}$$

Similar to the case of MDD, we consider the parameter $\psi$ as the overdispersion parameter that gives the MGD distribution the ability to capture data variation. Using $\psi = 1/A$, where $A = \Sigma_{k=1}^{K}\alpha_k$ and $P = \psi\alpha$, thus, Eq. (8.17) becomes:

$$\begin{aligned}\ln \mathcal{L}(P, \psi, \beta; X) = &\sum_{k=1}^{K}\left(\ln\Gamma\left(1/\frac{\psi}{P_k} + \beta_k\right) - \ln\Gamma\left(1/\frac{\psi}{P_k}\right) - \ln\Gamma(\beta_k)\right) \\ &+ \sum_{k=1}^{K} - \left(\ln\Gamma\left(1/\left(\frac{\psi}{P_k + X_k\psi}\right) + \beta'_k\right)\right. \\ &\left. - \ln\Gamma\left(1/\left(\frac{\psi}{P_k + X_k\psi}\right)\right) - \ln\Gamma(\beta'_k)\right)\end{aligned} \tag{8.18}$$

Considering $X^+$, $P^+$, $\beta^+$, $\beta'^+$ vectors of non-zero elements in $X$, $P$, $\beta$, and $\beta'$, respectively, where $K^+$ is the length of $X^+$, then, Eq. (8.18) becomes:

$$\begin{aligned}\ln \mathcal{L}(P^+, \psi, \beta^+; X^+) = &\sum_{k=1}^{K^+}\left(\underbrace{\ln\Gamma\left(1/\frac{\psi}{P_k^+} + \beta_k^+\right) - \ln\Gamma\left(1/\frac{\psi}{P_k^+}\right)}_{*} - \ln\Gamma(\beta_k^+)\right) \\ &+ \sum_{k=1}^{K^+} - \left(\underbrace{\ln\Gamma\left(1/\left(\frac{\psi}{P_k^+ + X_k^+\psi}\right) + \beta_k'^+\right) - \ln\Gamma\left(1/\left(\frac{\psi}{P_k^+ + X_k^+\psi}\right)\right)}_{**} - \ln\Gamma(\beta_k'^+)\right)\end{aligned} \tag{8.19}$$

Similar to the approach in [52], if the condition $xy \leq \delta$ is met, we can use the approximation (8.7) for all $K^+(**)$ and $K^+(*)$ paired log-gamma differences in (8.19):

$$\begin{aligned}\ln \mathcal{L}(P^+, \psi, \beta^+; X^+) = &\sum_{k=1}^{K^+}\left((-\beta_k^+ \ln(\frac{\psi}{P_k^+}) + D_m(\frac{\psi}{P_k^+}, \beta_k^+)) - \ln\Gamma(\beta_k^+)\right) \\ &+ \sum_{k=1}^{K^+}\left(\beta_k'^+ \ln\left(\frac{\psi}{P_k^+ + X_k^+\psi}\right) - D_m\left(\frac{\psi}{P_k^+ + X_k^+\psi}, \beta_k'^+\right) + \ln\Gamma(\beta_k'^+)\right)\end{aligned} \tag{8.20}$$

Here, we also used the same values for $m = 20$ and $\delta = 0.2$ used for MDD.

## 8.4 The Mesh Algorithm for Computing the Log-Likelihood Functions

As discussed earlier, when condition $xy \leq \delta$ is met, it is possible to make use of the approximation in Eq. (8.7) for optimizing the log-gamma difference in computing the MDD and MGD log-likelihood functions. However, when some of the terms in Eqs. (8.10) and (8.19) do not meet this condition, we may use the mesh algorithm, in which we rewrite the vector $\mathbf{X}$ into a sum of $L$ terms, choosing the terms to meet the following condition:

$$X = \sum_{l=1}^{L} X^{(l)} \tag{8.21}$$

For convenience, we define the choice of $X^{(l)}$ as below:

$$\alpha^{(l)} = \alpha + \sum_{i=1}^{l} X^{(i)}, \quad \text{for } l = 0, \ldots, L \tag{8.22}$$

$$\beta^{(l)} = \beta + \sum_{i=1}^{l} X^{(i)}, \quad \text{for } l = 0, \ldots, L \tag{8.23}$$

and the following relation between the adjacent $\alpha^{(l)}$'s and $\beta^{(l)}$'s:

$$\alpha^{(l-1)} + X^{(l)} = \alpha^{(l)}, \quad \text{for } l = 1, \ldots, L \tag{8.24}$$

$$\beta^{(l-1)} + X^{(l)} = \alpha^{(l)}, \quad \text{for } l = 1, \ldots, L \tag{8.25}$$

or

$$P^{(l-1)}/\psi^{(l-1)} + X^{(l)} = P^{(l)}/\psi^{(l)}, \quad \text{for } l = 1, \ldots, L \tag{8.26}$$

and we also have:

$$\frac{1}{\psi^{(l)}} = \frac{1}{\psi} + \sum_{i=1}^{l} N^{(i)}, \quad \text{for } 0 = 1, \ldots, L \tag{8.27}$$

$$\frac{1}{\psi^{(l)}} = \frac{1}{\psi^{(l-1)}} + N^{(l)}, \quad \text{for } l = 1, \ldots, L \tag{8.28}$$

For all $\psi \in [0, +\infty]$ we have:

$$\psi^{(l)} = \begin{cases} \dfrac{1}{\frac{1}{\psi} + \sum_{i=1}^{l} N^{(i)}} & \text{if } \psi \geq 1 \\ \dfrac{\psi}{1 + \psi \sum_{i=1}^{l} N^{(i)}} & \text{if } 0 \leq \psi < 1 \end{cases} \tag{8.29}$$

$$P^{(l)} = \begin{cases} \dfrac{\frac{p}{\psi} + \sum_{i=1}^{l} X^{(i)}}{\frac{1}{\psi} + \sum_{i=1}^{l} N^{(i)}} & \text{if } \psi \geq 1 \\ \dfrac{p + \psi \sum_{i=1}^{1} X^{(i)}}{1 + \psi \sum_{i=1}^{1} N^{(i)}} & \text{if } 0 \leq \psi < 1 \end{cases} \tag{8.30}$$

Then, for evaluating the log-likelihood function for MDD and MGD, respectively, we use:

$$\mathcal{L}(P, \psi; X) = \sum_{l=1}^{L} \ln \mathcal{L}(P^{(l-1)+}, \psi^{(l-1)}; X^{(l)+}) \tag{8.31}$$

$$\ln \mathcal{L}(P, \psi, \beta; X) = \sum_{l=1}^{L} \ln \mathcal{L}(P^{(l-1)+}, \psi^{(l-1)}, \beta^{(l-1)+}; X^{(l)+}) \tag{8.32}$$

Each of the $L$ terms in the above formulas can be computed using Eq. (8.11) for MDD and Eq. (8.20) for the MGD. This method is called the mesh algorithm [52], since the log-likelihood of the functions (8.31) and (8.32) can be evaluated incrementally on a mesh. The mesh algorithm for computing MDD and MGD log-likelihood functions can be described as follows:

- First, we generate the mesh using the following formula:

$$X_i^{(l)} = \lfloor \alpha_i^{(l-1)} \delta \rfloor \tag{8.33}$$

- Next, we select the level of the mesh $L$, so it would be the smallest integer satisfying:

$$\sum_{l=1}^{L} X_i^{(l)} \geq X_i, \quad \text{for } i = 1, \ldots, k \tag{8.34}$$

In this work, we used the level of mesh for MDD model $L = 3$ as in [52], and we found experimentally that the best level of mesh for MGD is $L = 5$.

- Afterwards, we adjust $X_i^{L_i^{'}}$ such that $\sum_{l=1}^{L_i^{'}} X_i^{(l)} = X_i$, and all the remaining $X_i^{(l)} (l > L_i^{'})$ will be set to zero. That is, we end the mesh totals to match $X_i$ exactly, given $L_i^{'}$ the smallest number satisfying:

$$\sum_{l=1}^{L_i^{'}} X_i^{(l)} \geq X_i, \quad \text{for } i = 1, \ldots, k \tag{8.35}$$

- Then, we use Eqs. (8.11) and (8.20) to compute the MDD and MGD log-likelihood functions, respectively.

## 8.5 Finite Mixture Models Learning

Finite mixture modeling is one of the formal approaches for clustering. A finite mixture model with $M$ components can be defined as:

$$P(\mathbf{X}|\Theta) = \sum_{j=1}^{M} P(\mathbf{X}|j; \boldsymbol{\theta}_\mathbf{j}) P(j) \tag{8.36}$$

where symbol $\Theta$ is the entire set of parameters that needs to be estimated, $\boldsymbol{\theta}_\mathbf{j}$ is the vector parameter for the $j$th population, $P(j)$ are the mixing proportions, satisfying $0 < P(j) \leq 1$ and $\sum_{j=1}^{M} P(j) = 1$. The maximum likelihood (ML) technique has been the most popular method for estimating the parameters which determine a mixture, within the last two decades [40]. Considering a set of $N$ independent vectors $\mathcal{X} = (\mathbf{X_1}, \ldots, \mathbf{X_N})$, the problem of estimating $\Theta$ using the ML estimation becomes:

$$\hat{\Theta}_{ML} = \arg\max_{\Theta}, P(\mathcal{X}|\Theta) \tag{8.37}$$

where

$$P(\mathcal{X}|\Theta) = \prod_{i=1}^{N} \sum_{j=1}^{M} P(\mathbf{X_i}|j; \boldsymbol{\alpha}_\mathbf{j}) P(j) \tag{8.38}$$

After taking logarithm on both sides of the above equation, we have:

$$\Phi(\mathcal{X}, \Theta) = \sum_{i=1}^{N} \log \left( \sum_{j=1}^{M} P(\mathbf{X_i}|j; \alpha_j) P(j) \right) \tag{8.39}$$

For learning the mixture parameters, we use the expectation–maximization (EM) algorithm, which is an iterative algorithm for obtaining the local maxima of the likelihood function [14], and it relies on the interpretation of $\mathbf{X}$ as incomplete data [29].

The EM algorithm consists of the following two steps:

1. **E-step**: in which we compute the posterior probabilities, as:

$$P^{(t)}(j|\mathbf{X_i}; \boldsymbol{\alpha_j}) = \frac{P^{(t-1)}(\mathbf{X_i}|j; \boldsymbol{\alpha_j}) P^{(t-1)}(j)}{\sum_{j=1}^{M} P^{(t-1)}(\mathbf{X_i}|j; \boldsymbol{\alpha_j}) P^{(t-1)}(j)} \tag{8.40}$$

2. **M-step**: Updates the parameters estimation as:

$$\hat{\Theta}^{(t)} = \arg\max_{\Theta}, \Phi(\mathcal{X}, \Theta^{(t-1)}) \tag{8.41}$$

For estimating the parameters $\alpha_j$ for MDD, $\alpha_j$ and $\beta_j$ for MGD, by setting the derivatives of the log-likelihood to zero, it can be seen that there are no closed form solutions, because of the existence of some terms such as $\Gamma(\alpha_i)$ and $\Gamma(\beta_i)$ in both models. Therefore, we use an iterative gradient descent optimization method by computing the gradient of the MDD likelihood, along with two bounds in equations [28, 33]. For the MGD, we use the Newton–Raphson method to estimate its parameters as proposed in [4].

Our initialization method can be described as follows:

1. Generate the vector of parameters $\alpha_j$ and $\beta_j$ randomly, for each component $j$.
2. Apply the $K$-means algorithm, to assign each data point to one of the existing clusters, with the assumption that the current model is correct.
3. Initialize the mixing proportions $P(j)$ such that:

$$P(j) = \frac{\text{number of elements in cluster } j}{N}$$

where $N$ is the number of data instances.

The summary of the EM algorithm for learning MDD or MGD, finite mixture model parameters is outlined in Algorithm 1.

**Algorithm 1:** EM algorithm for learning the mixture models parameters

**Input :** $K$-dimensional data set with $N$ vectors $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$, pre-specified number of clusters $M$

**State** Initialize the set of parameters $\Theta$, as discussed above.
**repeat**
  **State** {E-step}
  **for** $i \leftarrow 1$ **to** $N$ **do**
    **for** $j \leftarrow 1$ **to** $M$ **do**
      Compute the posterior probabilities:

$$P^{(t)}(j|\mathbf{X_i}; \boldsymbol{\alpha_j}) = \frac{P^{(t-1)}(\mathbf{X_i}|j; \boldsymbol{\alpha_j}) P^{(t-1)}(j)}{\sum_{j=1}^{M} P^{(t-1)}(\mathbf{X_i}|j; \boldsymbol{\alpha_j}) P^{(t-1)}(j)}$$

      where $P^{(t-1)}(\mathbf{X_i}|j; \boldsymbol{\alpha_j})$ is computed using the mesh algorithm discussed in Sect. 8.4.
    **end**
  **end**
**until** *convergence*;
**State** {M-step}
**for** $j \leftarrow 1$ **to** $M$ **do**
  Update the model parameters according to Eq. (8.40)
**end**

## 8.6 Experimental Results

In this section, we validate the performance of the proposed approach in clustering count, multicategorical, and overdispersed data with the MDD and MGD distributions, via two different applications: natural scenes categorization and facial expression recognition. In each application, we compare the accuracy of clustering different datasets, using the normal method and the mesh algorithm for the log-likelihood calculation.

For pre-processing, we used the SIFT (scale-invariant feature transform) [25] for feature extraction and the bag-of-features (BoF) [12] for representation. BoF is based on the frequency of visual words, provided from a visual vocabulary, which is obtained by the quantization (or histogramming) of local feature vectors, computed from a set of training images. All the $128D$ descriptors calculated by SIFT are binned into a collection of local features. Afterwards, K-means is used to cluster the extracted vectors to build the visual words vocabulary. Then, every image in the datasets was represented by a vector, indicating the number of a set of visual words, coming from the constructed visual vocabulary. Since we used the iterative EM scheme, the initial parameter values might affect the convergence and the overall outcome. Hence, we run each model over 100 times with different random initializations in order to have optimum results.

### 8.6.1 Natural Scenes Categorization

Image clustering is one of the most crucial topics in computer vision. In this experiment, we investigate the mesh algorithm's performance by scene clustering, which is a challenging application in the sense that in the real-life environment, they could be captured in various positions, distances, colors. Moreover, high probability of misclustering could be caused because of the noises that come from the background surroundings, which might have similar features as our natural scene targets.

In our experiments, we considered three different scene image datasets: SUN, Oliva and Torralba, and Fei-Fei and Perona. Each dataset was split with 80:20 ratio, to form the visual vocabulary and representation.

**SUN** dataset is a subset of the extensive Scene UNderstanding (SUN) database[1] [51] that contains 899 categories and 130,519 images. We use 1849 natural scenes belonging to six categories (458 coasts, 228 rivers, 231 forests, 247 fields, 518 mountains, and 167 sky/clouds). The average size of the images is 720 × 480 (landscape format) or 480 × 720 (portrait format). Samples from the considered subset are shown in Fig. 8.1.

**Oliva and Torralba (OT)** dataset [37] contains 2688 images clustered as eight categories: 360 coasts, 328 forests, 374 mountains, 410 open countries, 260 highways, 308 inside of cities, 356 tall buildings, and 292 streets. The average size of each image is 250 × 250 pixels. The last dataset is **Fei-Fei and Perona (FP)** [15], which includes 13 categories, only available in gray scale. This dataset consists of 2688 images (eight categories) of the OT dataset plus: 241 suburb residences, 174 bedrooms, 151 kitchens, 289 living rooms, and 216 offices. The average size of each image is approximately 250 × 300 pixels. Examples of images from these datasets are given in Fig. 8.2.

Table 8.1 represents the average clustering accuracies, using both the normal and mesh approach. As we can see, there are considerable improvements when we implemented our clustering algorithm using the mesh method. Among the tested

**Fig. 8.1** Sample images from the six categories in SUN dataset [51]

[1] https://groups.csail.mit.edu/vision/SUN/.

**Fig. 8.2** Example images from two of the used datasets. First row from OT [37], the second row contains the extra categories included in Fei-Fei and Perona dataset [15]

**Table 8.1** Clustering Accuracy using MMD and MGD with normal and mesh log-likelihood calculation

| Model | MDD | | MGD | |
|---|---|---|---|---|
| Method | Normal | Mesh | Normal | Mesh |
| SUN | 86.02% | 90.99% | 88.82% | 91.33% |
| Oliva and Torallba | 75.46% | 79.12% | 78.14% | 80.84% |
| Fei-Fei and Perona | 72.64% | 74.23% | 75.67% | 76.80% |

datasets, SUN had the highest accuracy of 86.02% and 90.99%, modeled by MDD, using normal and mesh methods, respectively, from which we can observe an improvement of 4.97% in the results when we implemented the mesh algorithm. The highest clustering accuracy using the same dataset, modeled by MGD, using normal method was 88.82%, following a 2.51% growth when implemented by the mesh algorithm. The OT and FP datasets also experiment 3.66% and 1.59%, and 2.7% and 1.13% increase in the clustering accuracies when applying the mesh method, modeled by MDD and MGD mixture distributions, respectively.

In addition, Figs. 8.3, 8.4, 8.5 and 8.6 show the confusion matrices when modeling the SUN dataset, with the MDD and MGD distribution, using the normal and mesh algorithms, respectively. From these figures, we can see that the best clustered objects are *coast* and *mountain*, which their accuracy has increased by 1.8% and 2.9%, and 2.9% and 0.1%, when using the mesh algorithm, modeled by MDD and MGD finite mixture models, respectively. Furthermore, we can notice that the misclassification between coast and river is happened because of having some similar features. Likewise, for the other scenes with a considerable amount of incorrectly clustered images: mountain and sky, and forest and field.

### *8.6.2 Facial Expression Recognition*

Facial expression recognition is one of the most important topics in various fields including computer vision and artificial intelligence. In fact, it is one of the most challenging tasks in social and interpersonal communication, since it is a natural way for human being to express emotions and therefore to show their intentions. The

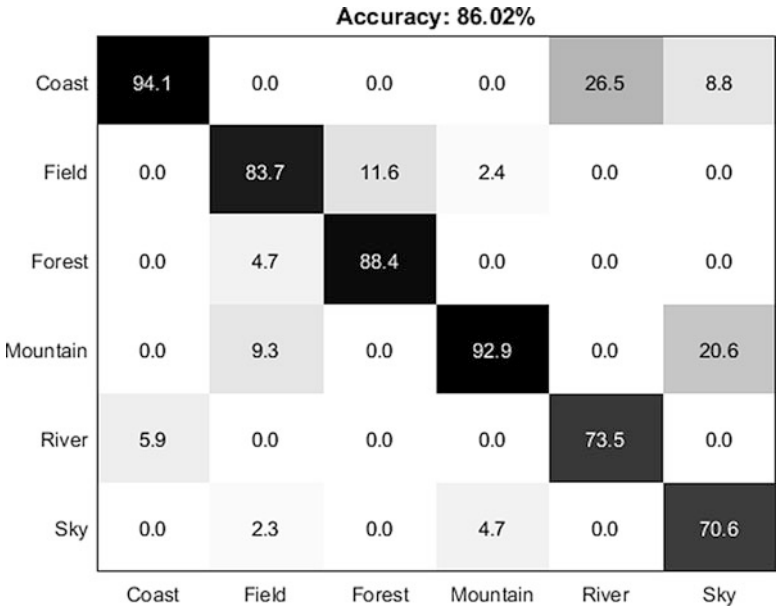



**Fig. 8.3** Confusion matrix for SUN dataset modeled by the MDD mixture, using normal method

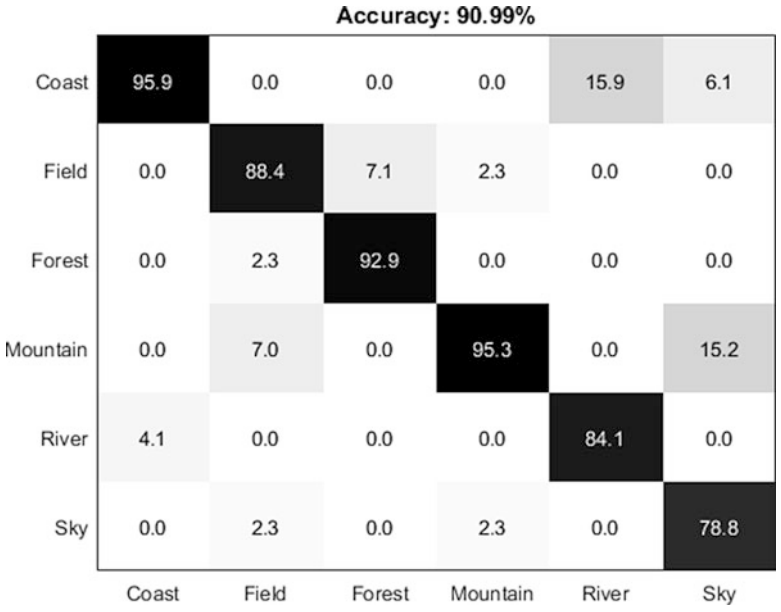



**Fig. 8.4** Confusion matrix for SUN dataset modeled by the MDD mixture, using mesh method

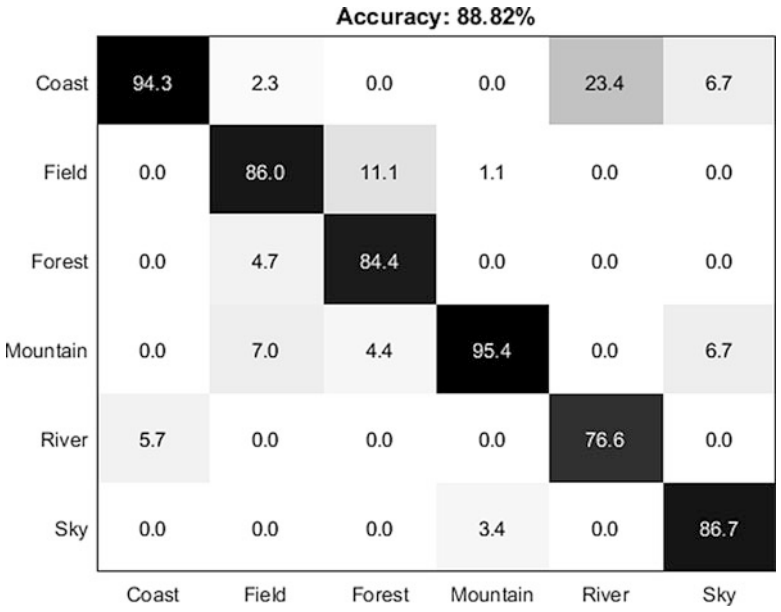



**Fig. 8.5** Confusion matrix for SUN dataset modeled by the MGD mixture, using normal method

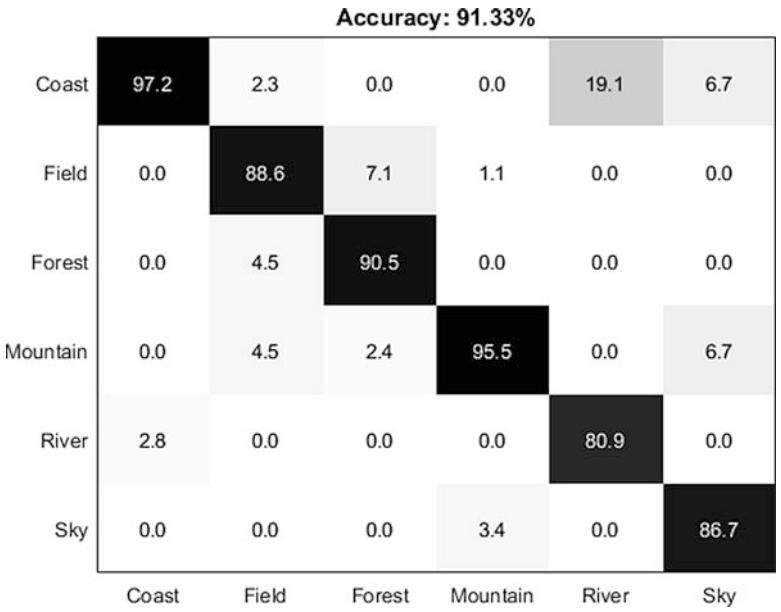



**Fig. 8.6** Confusion matrix for SUN dataset modeled by the MGD mixture, using mesh method

numerous number of expressions recognized in majority of the related databases makes the task hard as compared with other image categorization applications.

In this experiment, we used two different facial expression datasets: MMI and extended Cohn–Kanade (CK+). This time, each dataset was split into two halves, to form the visual vocabulary and representation.

**MMI** [47] database includes 19 different faces of students and research staff of 300 members of both genders (44% female), ranging in age from 19 to 62, having either a European, Asian, or South American ethnic background. Currently it contains 2894 image sequences where each image sequence has neutral face at the beginning and the end, and each with a size of $720 \times 576$ pixels. We selected the sequences that could be labeled as one of the six basic emotions. Removing the natural faces results in 1140 images.

The **extended Cohn–Kanade (CK+)** [26] dataset consists of facial behavior of 210 adults 18–50 years of age. Image sequences were digitized into either $640 \times 490$ or $640 \times 480$ pixel arrays with 8-bit grayscale value. We included all posed expressions that could be labeled as one of the six basic emotion categories which is about 4000 images (Table 8.2).

Examples from the used datasets are given in Figs. 8.7 and 8.8.

Table 8.3 demonstrates clustering accuracy, using MMD and MGD with normal and mesh log-likelihood calculation. By applying the mesh method, we are again having improvements when clustering both of the datasets as: 3.23% (MDD) and

**Table 8.2** Facial recognition expression datasets description

| | MMI dataset | | CK+ dataset | |
|---|---|---|---|---|
| Category | Number of images | Portion | Number of images | Portion |
| Anger | 150 | 13.16% | 342 | 9.5% |
| Disgust | 212 | 18.60% | 503 | 12.58% |
| Fear | 150 | 13.16% | 417 | 10.43% |
| Happiness | 255 | 22.37% | 993 | 24.83% |
| Sadness | 192 | 16.84% | 893 | 22.33% |
| Surprise | 181 | 15.88% | 852 | 21% |

**Fig. 8.7** Example images from MMI dataset [47]

**Fig. 8.8** Example images from CK+ dataset [26]

**Table 8.3** Clustering Accuracy using MMD and MGD with normal and mesh log-likelihood calculation

| Model | MDD | | MGD | |
|---|---|---|---|---|
| Method | Normal | Mesh | Normal | Mesh |
| MMI | 78.06% | 81.29% | 80.64% | 82.58% |
| CK+ | 71.08% | 73.96% | 73.24% | 74.96% |

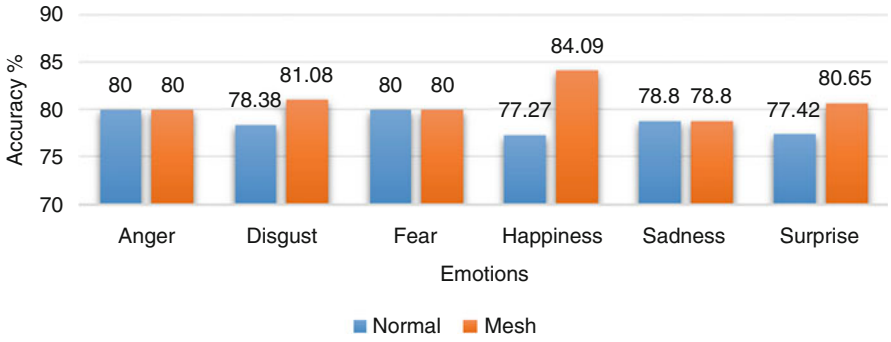



**Fig. 8.9** Accuracy comparison of each cluster for the MMI dataset, modeled by MDD mixture using both normal and mesh methods

1.94% (MGD) for MMI, and 2.88% (MDD) and 1.72% (MGD) for CK+. It is worth mentioning that both of the applications are experiencing better results, when using normal and mesh methods, modeled by the MGD mixture distribution.

Furthermore, Figs. 8.9 and 8.10 depict the accuracy comparison of each cluster for the MMI dataset, modeled by MDD and MGD mixtures, respectively, using both normal and mesh methods. From Fig 8.9, it can be observed that the disgust, happiness, and surprise emotions have gained 2.7%, 6.82%, and 3.23% of accuracy, respectively, when using the mesh algorithm. Moreover, Fig. 8.10 shows

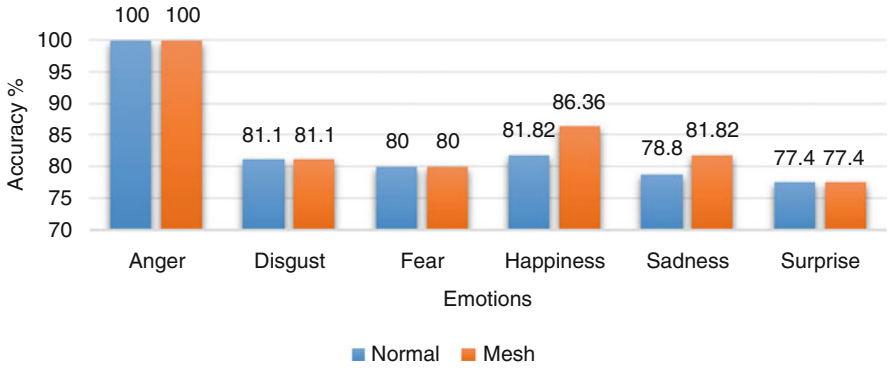



**Fig. 8.10** Accuracy comparison of each cluster for the MMI dataset, modeled by MGD mixture using both normal and mesh methods

that the happiness and sadness clusters are having 4.54% and 3.02% accuracy improvements, when mesh algorithm is implemented.

## 8.7 Conclusion

In this book chapter, we have introduced the usage of a novel method, for the computation of the log-likelihood functions, when clustering count, multicategorical data with overdispersion. The proposed approach was highly motivated because of the huge number of applications that involves such kind of data. The mesh method generally reduces the error when computing the log-likelihood function, and therefore increases the clustering accuracy. The effectiveness of this technique has been shown experimentally through two applications: such as natural scenes categorization and facial expression recognition. The presented procedure could be also applicable to other applications such as text document modeling and clustering, handwritten digit recognition, and bioinformatics including applications to metagenomics data and protein sequencing.

## References

1. Agresti, A., Kateri, M.: Categorical Data Analysis. Springer, New York (2011)
2. Anders, S., Huber, W.: Differential expression analysis for sequence count data. Genome Biol. **11**(10), R106 (2010)
3. Banfield, J.D., Raftery, A.E.: Model-based Gaussian and non-Gaussian clustering. Biometrics **49**, 803–821 (1993)
4. Bouguila, N.: Clustering of count data using generalized Dirichlet multinomial distributions. IEEE Trans. Knowl. Data Eng. **20**(4), 462–474 (2008)

5. Bouguila, N., Ziou, D., Vaillancourt, J.: Unsupervised learning of a finite mixture model based on the Dirichlet distribution and its application. IEEE Trans. Image Process. **13**(11), 1533–1543 (2004)
6. Busam, R., Freitag, E.: Complex Analysis. Springer, London (2009)
7. Cadez, I.V., Smyth, P., McLachlan, G.J., McLaren, C.E.: Maximum likelihood estimation of mixture densities for binned and truncated multivariate data. Mach. Learn. **47**(1), 7–34 (2002)
8. Cameron, A.C., Trivedi, P.K.: Regression Analysis of Count Data, vol. 53. Cambridge University Press, Cambridge (2013)
9. Casella, G., Berger, R.: Duxbury advanced series in statistics and decision sciences. Statistical Inference (2002)
10. Church, K.W., Gale, W.A.: Poisson mixtures. Nat. Lang. Eng. **1**(2), 163–190 (1995)
11. Connor, R.J., Mosimann, J.E.: Concepts of independence for proportions with a generalization of the Dirichlet distribution. J. Am. Stat. Assoc. **64**(325), 194–206 (1969)
12. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, Prague vol. 1, pp. 1–2 (2004)
13. De Dinechin, F., Lauter, C.Q.: Optimizing polynomials for floating-point implementation (2008). Preprint. arXiv:0803.0439
14. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B (Methodol.) **39**(1), 1–22 (1977)
15. Fei-Fei, L., Perona, P.: 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, pp. 524–531. IEEE, New York (2005)
16. Griffiths, D.: Maximum likelihood estimation for the beta-binomial distribution and an application to the household distribution of the total number of cases of a disease. Biometrics **29**(4), 637–648 (1973)
17. Haseman, J., Kupper, L.: Analysis of dichotomous response data from certain toxicological experiments. Biometrics **35**(1), 281–293 (1979)
18. Hilbe, J.M.: Negative Binomial Regression. Cambridge University Press, Cambridge (2011)
19. Katz, S.M.: Distribution of content words and phrases in text and language modelling. Nat. Lang. Eng. **2**(1), 15–59 (1996)
20. Leckenby, J.D., Kishi, S.: The Dirichlet multinomial distribution as a magazine exposure model. J. Market. Res. **21**(1), 100–106 (1984)
21. Lewy, P.: A generalized Dirichlet distribution accounting for singularities of the variables. Biometrics **52**(4), 1394–1409 (1996)
22. Lochner, R.H.: A generalized Dirichlet distribution in Bayesian life testing. J. R. Stat. Soc. Ser. B (Methodol.) **37**(1), 103–113 (1975)
23. Loh, W.Y.: Symmetric multivariate and related distributions. Technometrics **34**(2), 235–236 (1992)
24. Lowe, S.A.: The beta-binomial mixture model and its application to TDT tracking and detection. In: Proceedings of DARPA Broadcast News Workshop, pp. 127–131 (1999)
25. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
26. Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended Cohn-Kanade dataset (CK+): a complete dataset for action unit and emotion-specified expression. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pp. 94–101. IEEE, New York (2010)
27. MacKay, D.J., Peto, L.C.B.: A hierarchical Dirichlet language model. Nat. Lang. Eng. **1**(3), 289–308 (1995)
28. Madsen, R.E., Kauchak, D., Elkan, C.: Modeling word burstiness using the Dirichlet distribution. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 545–552. ACM, New York (2005)
29. McLachlan, G., Krishnan, T.: The EM Algorithm and Extensions, vol. 382. Wiley, Hoboken (2007)

30. McLachlan, G., Peel., D.: Finite Mixture Models. Wiley, Hoboken (2000)
31. McLachlan, G.J., Lee, S.X., Rathnayake, S.I.: Finite mixture models. Annu. Rev. Stat. Appl. **6**, 355–378 (2000)
32. Mimno, D., McCallum, A.: Topic models conditioned on arbitrary features with Dirichlet-multinomial regression (2012). Preprint. arXiv:1206.3278
33. Minka, T.: Estimating a Dirichlet distribution (2000). http://research.microsoft.com/~minka/papers/dirichlet
34. Mosimann, J.E.: On the compound multinomial distribution, the multivariate $\beta$-distribution, and correlations among proportions. Biometrika **49**(1/2), 65–82 (1962)
35. Neerchal, N.K., Morel, J.G.: An improved method for the computation of maximum likelihood estimates for multinomial overdispersion models. Comput. Stat. Data Anal. **49**(1), 33–43 (2005)
36. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Mach. Learn. **39**(2–3), 103–134 (2000)
37. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput. Vis. **42**(3), 145–175 (2001)
38. Poortema, K.: On modelling overdispersion of counts. Stat. Neerl. **53**(1), 5–20 (1999)
39. Puig, P., Valero, J.: Count data distributions: some characterizations with applications. J. Am. Stat. Assoc. **101**(473), 332–340 (2006)
40. Redner, R.A., Walker, H.F.: Mixture densities, maximum likelihood and the EM algorithm. SIAM Rev. **26**(2), 195–239 (1984)
41. Rennie, J.D., Shih, L., Teevan, J., Karger, D.R.: Tackling the poor assumptions of Naive Bayes text classifiers. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03), pp. 616–623 (2003)
42. Rowe, C.H.: A proof of the asymptotic series for log $\gamma$ (z) and log $\gamma$ (z+ a). Ann. Math. **32**(1), 10–16 (1931)
43. Rust, R.T., Leone, R.P.: The mixed-media Dirichlet multinomial distribution: a model for evaluating television-magazine advertising schedules. J. Mark. Res. **21**(1), 89–99 (1984)
44. Teevan, J., Karger, D.R.: Empirical development of an exponential probabilistic model for text retrieval: using textual analysis to build a better model. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 18–25. ACM, New York (2003)
45. Tirri, H., Kontkanen, P., Myllym Aki, P.: Probabilistic instance-based learning. In: Machine Learning: Proceedings of the Thirteenth International Conference, pp. 507–515 (1996)
46. Ueda, N., Saito, K.: Parametric mixture models for multi-labeled text. In: Advances in Neural Information Processing Systems, pp. 737–744 (2003)
47. Valstar, M., Pantic, M.: Induced disgust, happiness and surprise: an addition to the MMI facial expression database. In: Proc. 3rd Intern. Workshop on EMOTION (Satellite of LREC): Corpora for Research on Emotion and Affect, Paris, p. 65 (2010)
48. Whittaker, E., Watson, G.: A Course of Modern Analysis. Cambridge University Press, Cambridge (1990)
49. Wong, T.T.: Generalized Dirichlet distribution in Bayesian analysis. Appl. Math. Comput. **97**(2–3), 165–181 (1998)
50. Wong, T.T.: Alternative prior assumptions for improving the performance of naïve Bayesian classifiers. Data Min. Knowl. Disc. **18**(2), 183–213 (2009)
51. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: large-scale scene recognition from abbey to zoo. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3485–3492. IEEE, New York (2010)
52. Yu, P., Shaw, C.A.: An efficient algorithm for accurate computation of the Dirichlet-multinomial log-likelihood function. Bioinformatics **30**(11), 1547–1554 (2014)
53. Zamzami, N., Bouguila, N.: Consumption behavior prediction using hierarchical Bayesian frameworks. In: 2018 First International Conference on Artificial Intelligence for Industries (AI4I), pp. 31–34. IEEE, New York (2018)

# Part IV
# Bounded and Semi-bounded Data Clustering

# Chapter 9
# A Frequentist Inference Method Based on Finite Bivariate and Multivariate Beta Mixture Models


**Narges Manouchehri and Nizar Bouguila**

**Abstract** Modern technological improvement, revolutionized computers, progress in scientific methods, and other related factors led to generate a massive volume of structured and unstructured data. Such valuable data has potential to be mined for information retrieval and analyzed computationally to reveal patterns, trends, and associations that lead to better decisions and strategies. Thus, machine learning and specifically, unsupervised learning methods have become the topic of interest of much recent researches in data engineering. Finite mixture models as unsupervised learning methods, namely clustering, are considered as capable techniques for discovery, extraction, and analysis of knowledge from data. Traditionally Gaussian mixture model (GMM) has drawn lots of attention in previous literature and has been studied extensively. However, other distributions demonstrate more flexibility and convenience in modeling and describing data.

The novel aspect of this work is to develop a framework to learn mixture models based on bivariate and multivariate Beta distributions. Moreover, we tackle simultaneously the problems of parameters estimation, cluster validation, or model selection which are principal challenges in deployment of mixture models. The effectiveness, utility, and advantages of the proposed method are illustrated through extensive empirical results using real datasets and challenging applications involving image segmentation, sentiment analysis, credit approval, and medical inference.

N. Manouchehri (✉)
Department of Electrical and Computer Engineering (ECE), Concordia University, Montreal, QC, Canada
e-mail: narges.manouchehri@mail.concordia.ca

N. Bouguila
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca



N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6_9

## 9.1 Introduction

During the last few decades, scientific and technological advances have created new challenges as huge amounts of data are produced every day. To deal with large-scale data which are explosively generated, a great deal of effort has been expended on developing robust solutions to discover hidden valuable knowledge and recognize significant facts, relationships, trends, patterns, and anomalies. Machine learning has been studied extensively as a method to analyze the structure of complex data and fit it into models that can be understood with the help of statistical and computational approaches. Clustering as one of its main branches is a continuously developing field and mixture models provide flexible and convenient classes of models among the most statistically mature methods for clustering [1, 2]. In this probabilistic method, each observation belongs to one of some number of different groups. Despite their successful utilization in wide spectrum of research areas, there are crucial challenges when deploying this technique such as selecting a flexible mixture density which demonstrates more efficiency in modeling asymmetric data, parameter estimation, determination of the proper number of clusters, and defining model complexity. Most of the literature on mixtures concern Gaussian mixture model (GMM) [3]. However, GMM is not a proper tool to express the latent structure of non-Gaussian data. Recently, other distributions which are more flexible have been considered as a powerful alternative [4–15].

In this work, we introduce unsupervised learning algorithms for finite mixture models based on bivariate and multivariate Beta distributions which could be applied in various real-world challenging problems. Our proposed learning framework will deploy deterministic approaches such as maximum likelihood (ML) and Newton Raphson methods via Expectation Maximization (EM). Furthermore, for model selection, minimum message length (MML) criterion is validated to find the optimal number of clusters inherent within real data sets. We evaluated our clustering approach on different problems. In Sect. 9.2, we describe our framework by introducing the bivariate and multivariate Beta distributions. Section 9.3 is devoted to model learning and parameters estimation. In Sect. 9.4, we discuss model complexity, specifically through minimum message length (MML). The learning algorithm is summarized in Sect. 9.5. Section 9.6 is dedicated to investigating the performance of our framework by testing it on real data sets and real-life applications. Finally, in Sect. 9.6 we conclude our work and highlight some future challenges.

## 9.2 Mixture Model

We assume $\mathcal{X} = \{\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X}_N\}$ is a set of $N$ $d$-dimensional vectors and each vector $\mathbf{X}_n = (X_{n1}, \ldots, X_{nd})$ is generated from a finite but unknown mixture model

$p(\mathbf{X}|\Theta)$. Considering $\mathcal{X}$ as a composition of $M$ different clusters, we can describe it by a finite mixture model as defined below [2]:

$$p\left(\mathbf{X}|\Theta\right) = \sum_{j=1}^{M} p_j\, p(\mathbf{X}|\mathbf{a}_j) \tag{9.1}$$

$p_j$ denotes the weights of component $j$ or mixing proportions which are all positive and sum to one. $p(\mathbf{X}|\mathbf{a}_j)$ is the distribution which in this work will be multivariate Beta distribution. We introduce bivariate and multivariate Beta distributions in Sects. 9.2.1 and 9.2.2, respectively. $(p_j, \mathbf{a}_j)$ represents weight and shape parameters of component $j$ and the complete model parameters is denoted by $\Theta = \{p_1, \ldots, p_M, \mathbf{a}_1, \ldots, \mathbf{a}_M\}$.

### 9.2.1 Bivariate Beta Distribution

The bivariate Beta distribution has been proposed in [3, 16] as follows. Let $X$ and $Y$ be two random variables following a bivariate Beta distribution which are both positive real values and less than one. They are derived from $U$, $V$, and $W$ as three independent random variables arise from standard Gamma distribution and parameterized by their shape parameters $a$, $b$, and $c$, respectively and described by the following equations:

$$X = \frac{U}{(U+W)}, \quad Y = \frac{V}{(V+W)} \tag{9.2}$$

$$E(X) = \frac{a}{(a+c)}, \quad Var(X) = \frac{ac}{(a+c)^2(a+c+1)} \tag{9.3}$$

$$E(Y) = \frac{b}{(b+c)}, \quad Var(Y) = \frac{bc}{(b+c)^2(b+c+1)} \tag{9.4}$$

The joint density function of the bivariate distribution is expressed as follows:

$$f(X,Y) = \frac{X^{a-1}Y^{b-1}(1-X)^{b+c-1}(1-Y)^{a+c-1}}{B(a,b,c)(1-XY)^{(a+b+c)}} \tag{9.5}$$

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

Figures 9.1, 9.2, 9.3 and 9.4 illustrate four examples of one component Beta mixture models (BBMM) with different values for shape parameters. The mixture of two, three, four, and five components are displayed in Figs. 9.5, 9.6, 9.7 and 9.8. According to these figures, it is clear that the BBMM offers various flexible shapes.

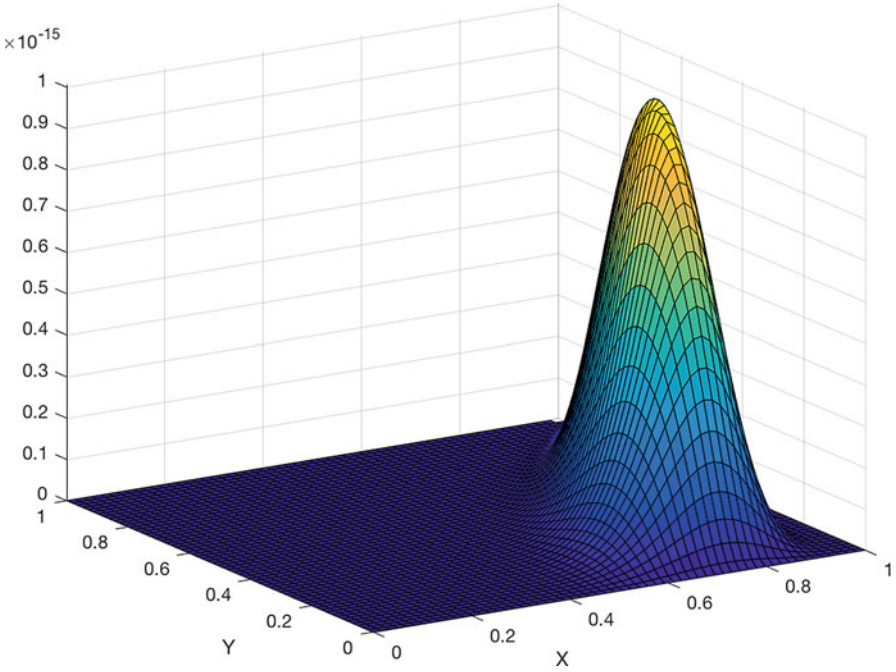



**Fig. 9.1** One-component BBMM

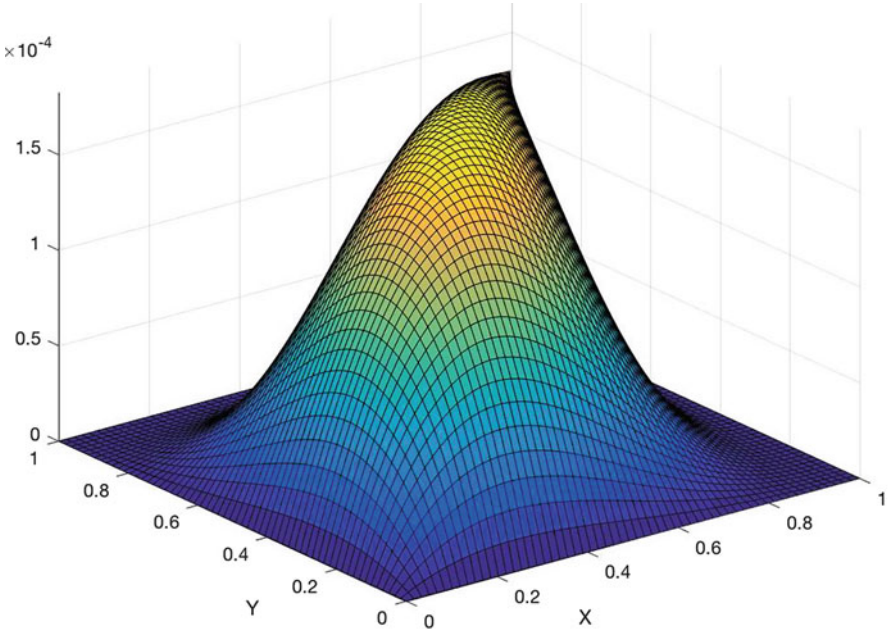



**Fig. 9.2** One-component BBMM

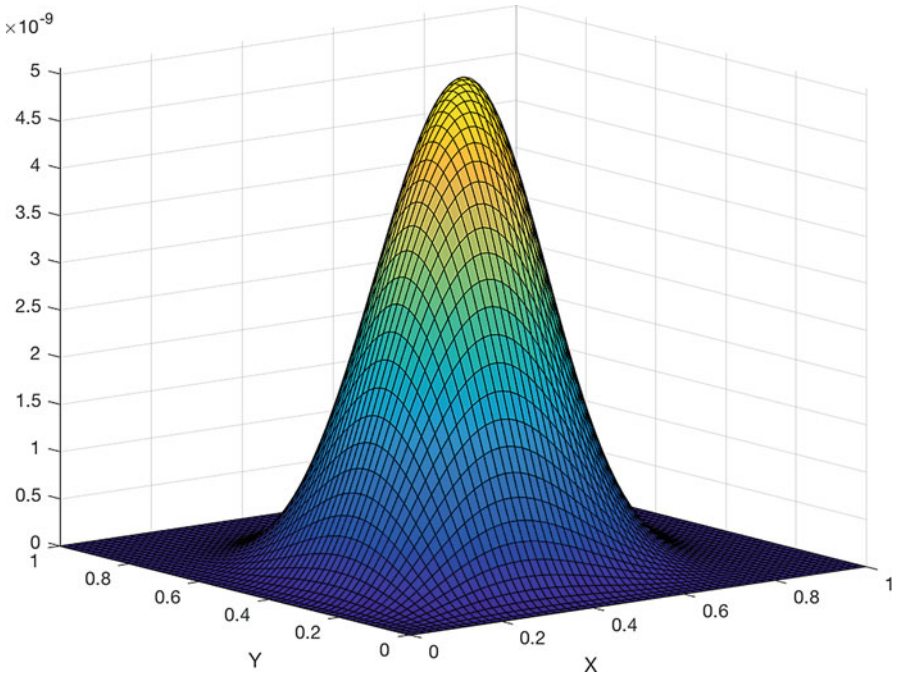



**Fig. 9.3** One-component BBMM

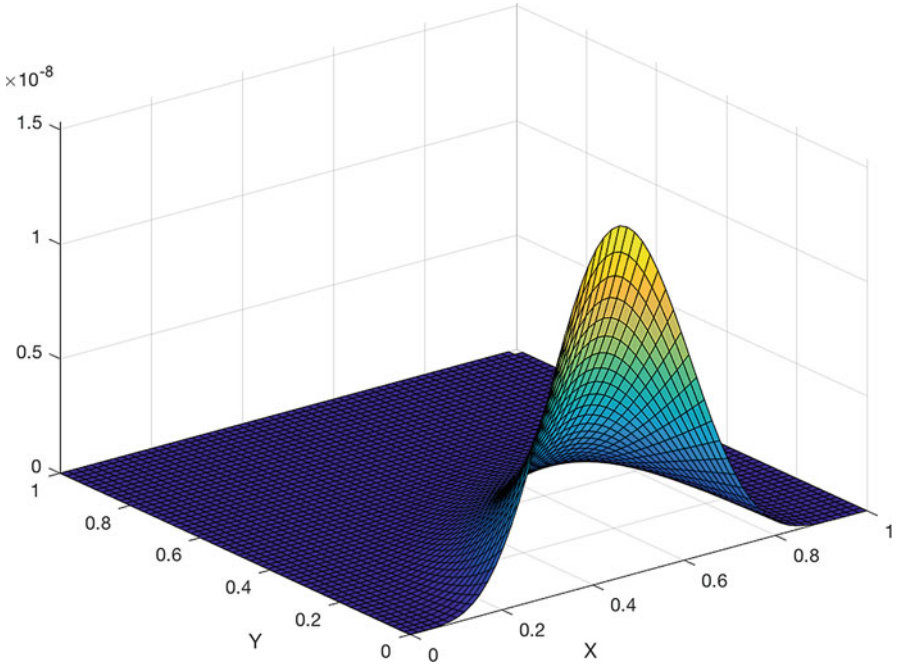



**Fig. 9.4** One-component BBMM

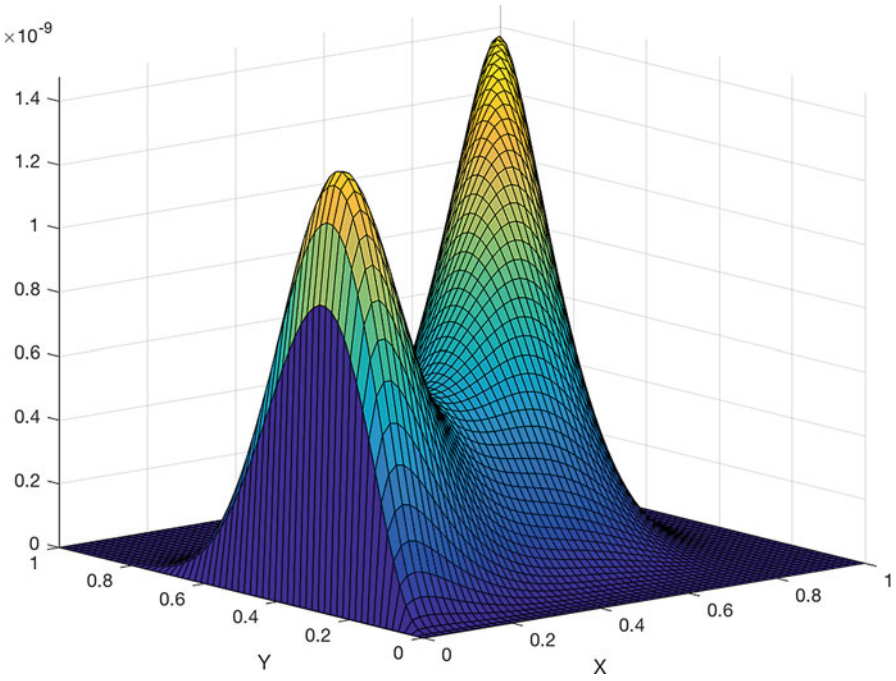



**Fig. 9.5** Two-component BBMM

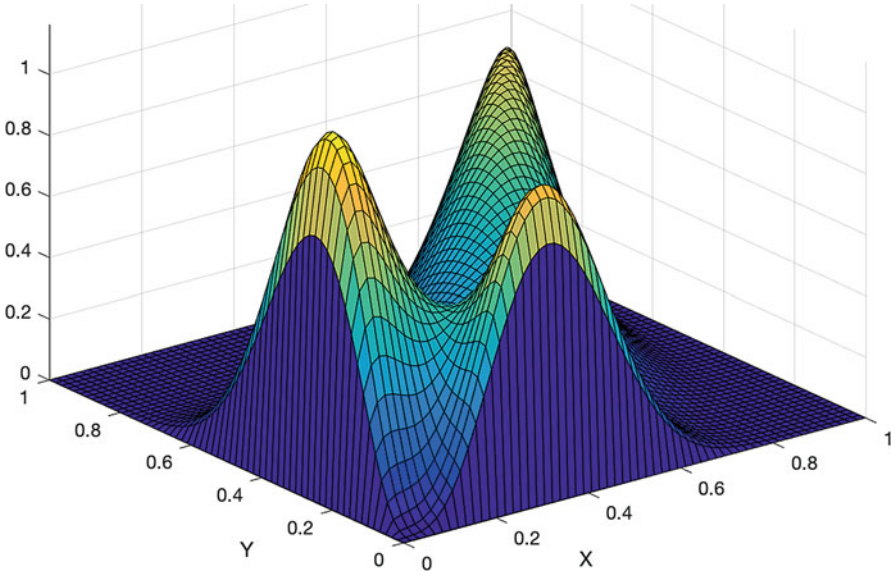



**Fig. 9.6** Three-component BBMM

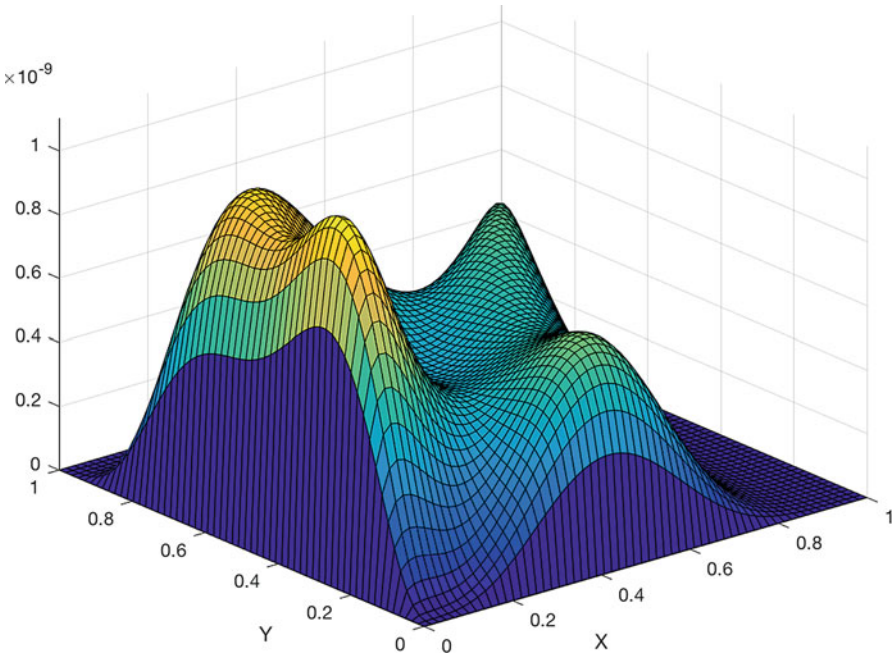



**Fig. 9.7** Four-component BBMM

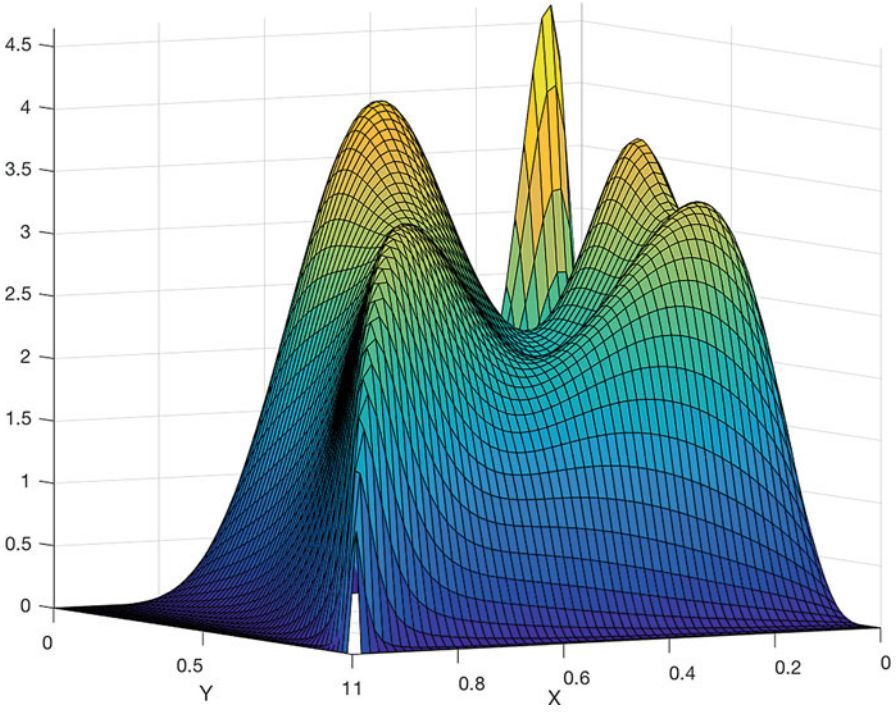



**Fig. 9.8** Five-component BBMM

### 9.2.2 Multivariate Beta Distribution

The multivariate Beta distribution is constructed by generalization of the bivariate distribution to $d$ variate distribution [3, 16]. Let $U_1, \ldots, U_d$ and $W$ be independent random variables each having a Gamma distribution and variable $X$ is defined by Eq. (9.6) where $i = 1, \ldots, d$.

$$X_i = \frac{U_i}{(U_i + W)} \tag{9.6}$$

The joint density function of $X_1, \ldots, X_d$ after integration over $W$ is expressed by:

$$f(X_1, \ldots, X_d) = c \frac{\prod_{i=1}^{d} X_i^{a_i - 1}}{\prod_{i=1}^{d} (1 - X_i)^{(a_i + 1)}} \Big[1 + \sum_{i=1}^{d} \frac{X_i}{(1 - X_i)}\Big]^{-a} \tag{9.7}$$

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

$$c = B^{-1}(a_1, \ldots, a_d) = \frac{\Gamma(a_1 + \cdots + a_d)}{\Gamma(a_1) \ldots \Gamma(a_d)} = \frac{\Gamma(a)}{\prod_{i=1}^{d} \Gamma(a_i)} \tag{9.8}$$

$a_i$ is the shape parameter of each variable $X_i$ and:

$$a = \sum_{i=1}^{d} a_i \tag{9.9}$$

## 9.3 Model Learning

To learn our model, we first apply k-means to initially cluster our data and with the help of mean and variance of clusters, the initial shape parameters of distribution can be approximated using the method of moments. To update the parameters, we apply deterministic and efficient techniques such as maximum likelihood (ML) and Newton Raphson.

### 9.3.1 Maximum Likelihood via EM Algorithm

To tackle the model estimation problem, the parameters which maximize the probability density function of data are determined using ML [17, 18] via EM framework [19]. ML is an estimation procedure to find the mixture model parameters that maximize log-likelihood function [20] which is defined by:

$$L(\Theta, \mathcal{X}) = \log p(\mathcal{X}|\Theta) = \sum_{n=1}^{N} \log \Big( \sum_{j=1}^{M} p_j \, p(\mathbf{X}_n|\boldsymbol{\alpha}_j) \Big) \tag{9.10}$$

Each $\mathbf{X}_n$ is supposed to be arisen from one of the components. Hence, membership vectors are introduced as $\mathcal{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_N\}$. If $\mathbf{X}_n$ belongs to cluster $j$, $Z_{nj} = 1$ and $Z_{nj} = 0$, otherwise [2]. In Expectation phase, we assign each vector $\mathbf{X}_n$ to one of the clusters by its posterior probability given by:

$$\hat{Z}_{nj} = p(j|\mathbf{X}_n, \mathbf{a}_j) = \frac{p_j \, p(\mathbf{X}_n|\mathbf{a}_j)}{\sum_{j=1}^{M} p_j \, p(\mathbf{X}_n|\mathbf{a}_j)} \tag{9.11}$$

The complete log-likelihood is computed as:

$$L(\Theta, Z, \mathcal{X}) = \sum_{j=1}^{M} \sum_{n=1}^{N} \hat{Z}_{nj} \big( \log p_j + \log p(\mathbf{X}_n|\mathbf{a}_j) \big) \tag{9.12}$$

In maximization step, the gradient of the log-likelihood with respect to parameters is calculated. To solve optimization problem, a solution for the following equation should be found.

$$\frac{\partial \log L(\Theta, Z, \mathcal{X})}{\partial \Theta} = 0 \tag{9.13}$$

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

$$\hat{\mathbf{a}_j}^{new} = \hat{\mathbf{a}_j}^{old} - {H_j}^{-1} G_j \tag{9.14}$$

where $G_j$ is the vector of first derivatives vector and $H_j$ is the matrix of the second derivatives, namely Hessian matrix.

For MBMM, $G_j$ as the first derivatives of Eq. (9.12) with respect to $a_{ji}$ where $i = 1, \ldots, d$ is given by:

$$\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial a_{ji}} = \sum_{j=1}^{M} \sum_{n=1}^{N} \hat{Z}_{nj} \Bigg( \log(X_{ni}) - \log(1 - X_{ni}) + \Psi(|\mathbf{a}_j|) - \Psi(a_{ji}) - \log \Big[ 1 + \sum_{i=1}^{d} \frac{X_{ni}}{(1 - X_{ni})} \Big] \Bigg) \tag{9.15}$$

$G_j$ and $H_j$ are described as follows:

$$G_j = \big( G_{1j}, \ldots, G_{dj} \big)^T \tag{9.16}$$

$$H_j = \begin{bmatrix} \frac{\partial G_{1j}}{\partial a_{1j}} & \cdots & \frac{\partial G_{1j}}{\partial a_{dj}} \\ \vdots & \ddots & \vdots \\ \frac{\partial G_{dj}}{\partial a_{1j}} & \cdots & \frac{\partial G_{dj}}{\partial a_{dj}} \end{bmatrix} = \sum_{i=1}^{N} \hat{Z}_{nj} \begin{bmatrix} \Psi'(|\mathbf{a}_j|) - \Psi'(a_{j1}) & \ldots & \Psi'(|\mathbf{a}_j|) \\ \vdots & \ddots & \vdots \\ \Psi'(|\mathbf{a}_j|) & \ldots & \Psi'(|\mathbf{a}_j|) - \Psi'(a_{dj}) \end{bmatrix} \tag{9.17}$$

where

$$|\mathbf{a}_j| = a_1 + \ldots + a_d \tag{9.18}$$

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

$$\Psi(X) = \frac{\Gamma'(X)}{\Gamma(X)}, \Psi'(X) = \frac{\Gamma''(X)}{\Gamma(X)} - \frac{\Gamma'(X)^2}{\Gamma(X)^2} \tag{9.19}$$

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

$$p_j = \frac{\sum_{n=1}^{N} p(j|\mathbf{X}_n, \mathbf{a}_j)}{N} \tag{9.20}$$

## 9.4 Selection of Model Complexity with MML

An important challenge of the modeling problem concerns determining the number of consistent components which best describes the data. In this section, we consider MML as a method that has been proved to outperform many other model selection methods. This approach is based on evaluating statistical models according to their ability to compress a message containing the data. The optimal number of components of the mixture is that which minimizes the amount of information needed to transmit data $\mathcal{X}$ efficiently from a sender to a receiver and high compression is obtained by forming good models of the data to be coded [20–25].

The formula for the message length for a mixture of distributions is given by Eq. (9.21) where $h(\Theta)$ is the prior probability, $p(\mathcal{X}|\Theta)$ is the likelihood, $F(\Theta)$ is the expected Fisher information matrix, and $|F(\Theta)|$ is its determinant. $N_p$ is the number of free parameters to be estimated and is equal to $(M(d+1)) - 1$. $\kappa N_p$ is the optimal quantization lattice constant for $\mathbb{R}^{N_p}$. We have $\kappa_1 \simeq 0.083$ for $N_p = 1$ [20–32].

$$MessLen \simeq -\log(h(\Theta)) - \log\left(p(\mathcal{X}|\Theta)\right) + \frac{1}{2}\log(|F(\Theta)|) + \frac{N_p}{2}(1 + \log(\kappa N_p)) \tag{9.21}$$

The Fisher information matrix is the expected value of the Hessian minus the logarithm of the likelihood. We use the complete data Fisher information matrix. The determinant of the complete data Fisher information matrix is:

$$|F(\Theta)| \simeq |F(\mathbf{P})| \prod_{j=1}^{M} |F(\mathbf{a}_j)| \tag{9.22}$$

where $|F(\mathbf{P})|$ is the Fisher information with regard to the mixing parameters vector, and $|F(\boldsymbol{\alpha}_j)|$ is the Fisher information with regard to the vector $\mathbf{a}_j$ of a single multivariate Beta distribution. For $|F(\mathbf{P})|$, mixing parameters satisfy the requirement $\sum_{j=1}^{M} p_j = 1$. Consequently, it is possible to consider the generalized Bernoulli process with a series of trials, each of which has $M$ possible outcomes for $M$ clusters. The determinant of the Fisher information matrix with respect to the mixing parameters is given by Eq. (9.23) where $N$ is the number of data elements.

$$|F(\mathbf{P})| = \frac{N}{\prod_{j=1}^{M} p_j} \tag{9.23}$$

The Fisher information for our mixture is given as follows:

$$\log |F(\Theta)| = \log(N) - \sum_{j=1}^{M} \log p_j + \sum_{j=1}^{M} \log |F(\mathbf{a}_j)| \tag{9.24}$$

where $F(\mathbf{a}_j)$ is:

$$F(\mathbf{a}_j) = \left(1 - \Psi'(|\mathbf{a}_j|) \sum_{d=1}^{D} \frac{1}{\Psi'(a_{jd})}\right) n_j^D \prod_{d=1}^{D} \Psi'(a_{jd}) \tag{9.25}$$

To calculate MML, we need to choose $h(\Theta)$ which can be represented as follows [33]:

$$h(\Theta) = h(\mathbf{p}) h(\mathbf{a}) \tag{9.26}$$

Considering the nature of the mixing parameters, it can be expressed by a symmetric Dirichlet distribution with parameter. It is shown in Eq. (9.27) where $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_M)$ is the parameter vector of the Dirichlet distribution:

$$h(\mathbf{p}) = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} {p_j}^{\eta_j - 1} \tag{9.27}$$

The choice of $\eta_1 = 1, \ldots, \eta_M = 1$ gives a uniform prior over the space $p_1 + \cdots + p_M = 1$. Therefore, the prior is given by

$$h(\mathbf{p}) = (M - 1)! \tag{9.28}$$

For $\mathbf{a}$, we assume that components of $\mathbf{a}_j$ are independent:

$$h(\mathbf{a}) = \prod_{j=1}^{M} h(\mathbf{a}_j) = \prod_{j=1}^{M} \prod_{d=1}^{D} h(a_{jd}) \tag{9.29}$$

For MBMM, the following simple uniform prior was experimentally found good results with it [20–32].

$$h(a_{jd}) = e^{-6} \frac{a_{jd}}{\|\mathbf{a}_j\|} \tag{9.30}$$

where $\|\mathbf{a}_j\|$ is the norm of the shape vector. The log of prior is given by:

$$\log(h(\Theta)) = \sum_{j=1}^{M-1} \log(j) - 6MD - D \sum_{j=1}^{M} \log(\|\mathbf{a}_j\|) + \sum_{j=1}^{M} \sum_{d=1}^{D} \log(a_{jd}) \tag{9.31}$$

The complete estimation framework is described as follows:

1. INPUT: $\mathcal{X}$ and $M$.
2. Initialization: Apply the k-means and method of moments to obtain initial $M$ clusters.
3. Apply the moments method for each component $j$ to obtain $\mathbf{a}_j$.
4. Expectation step: Compute $\hat{Z}_{nj}$ using Eq. (9.11).
5. Maximization step: Update $\mathbf{a}_j$ Using Eq. (9.14) and $p_j$ Eq. (9.20).
6. If $p_j < \epsilon$, discard component $j$ and go to 4.
7. If the convergence criterion passes terminate, else go to 4.
8. Calculate the associated criterion of MML and select the optimal number of components.

## 9.5 Experimental Results

In order to validate the performance of our proposed algorithm, we test bivariate and multivariate models using real data sets and real-world applications and compare them with Gaussian mixture model (GMM). As the first step, we normalize our datasets using Eq. (9.32) as one of the assumptions of our distribution is that the values of all observations are positive and less than one.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{9.32}$$

### 9.5.1 Experimental Results for BBMM

In this section, we will test the performance of our model on four real datasets from UCI repository [34] and one real-world application, namely image segmentation using images from Berkeley dataset [35].

#### 9.5.1.1 Business Mortality Dataset

This dataset is based on the results of a research conducted by R.G. Hutchinson, A.R. Hutchinson, and M. Newcomer in 1938 about business mortality including 1225 observations. This work focused on survival times of three service firms including saloons, restaurants, and express services in a period of 5 to 6 years in Poughkeepsie, New York [34]. The service type and years of activity are considered as two variables. The dataset is labeled by a binary variable 0 and one. BBMM has a performance of 91.39% while the accuracy is 89.59% for GMM. The MML result is presented in Fig. 9.9 which validates this technique.

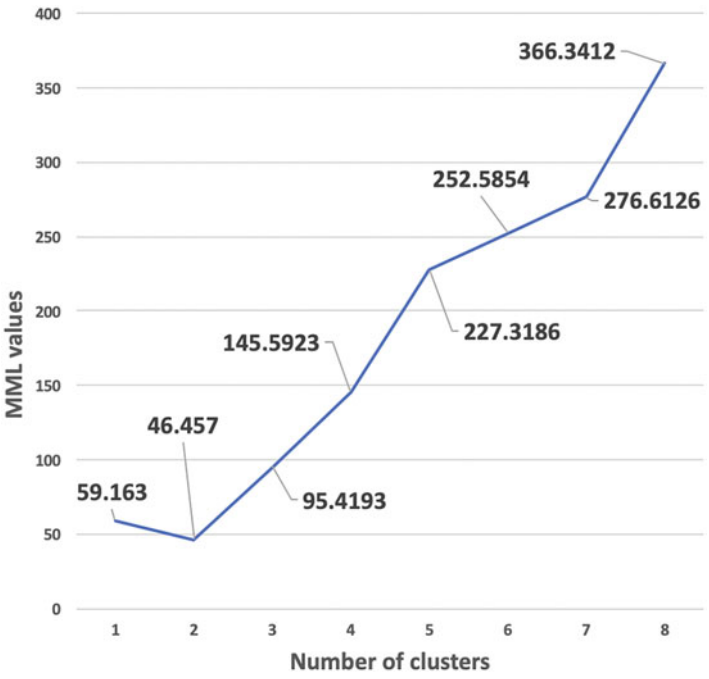



**Fig. 9.9** MML result for business mortality dataset

#### 9.5.1.2 Grain Diameters of Pollen Dataset

The second data set contains 1650 instances which report grain diameters of pollen based on five species, with varying numbers of locations within species (13, 6, 5, 19, 12) and 30 measurements per location [34]. The two variables are grain diameter and species. Moreover, location is considered as the label of dataset. BBMM is 90.27% accurate while accuracy is 88.95% for GMM. MML was validated for this dataset as shown in Fig. 9.10.

#### 9.5.1.3 Sailing Speed Optimization Dataset

This data set contains 100 observations as the result of a research about sailing speed optimization for container ships in a liner shipping network reported in 2012. In this study, the average speed (knots) and fuel consumption (tons/day) for five ship-type and voyage leg combinations were measured. We considered speed and fuel consumption as two variables [34]. Voyage leg combinations is the label of dataset measured by TEU (20-foot equivalent unit) which has five types: 1: 3000-TEU Singapore-Jakarta, 2: 3000-TEU Singapore-Kaohsiung, 3: 5000-TEU Hong Kong-Singapore, 4: 8000-TEU Yantian-Los Angeles, 5: 8000-TEU Tokyo-Xiamen. The

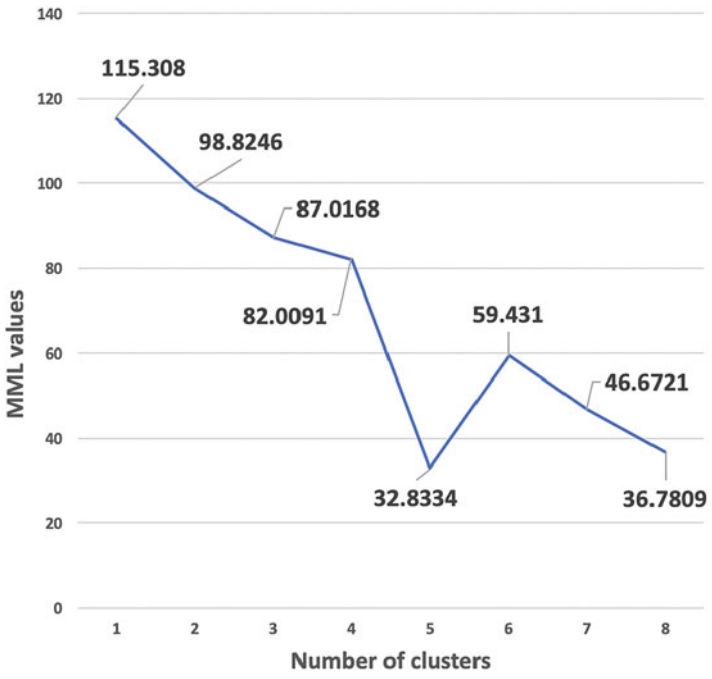



**Fig. 9.10** MML result for grain diameter dataset

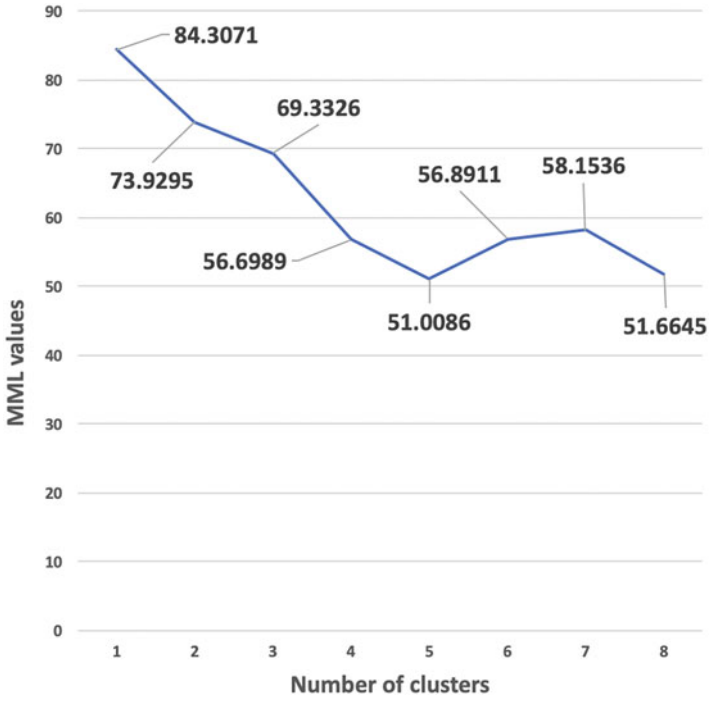



**Fig. 9.11** MML result for sailing speed optimization dataset

performances of BBMM and GMM are 94.51% and 90.19%, respectively. MML was validated as shown in Fig. 9.11.

#### 9.5.1.4 Water Levels of Florida Swamps Dataset

This data set contains 243 observations based on studying water depths at 27 locations within each of 9 swamps (3 large, 3 medium, 3 small) in Florida [34]. The water level and swamp numbers are two variables and swamp size is label. BBMM outperforms the GMM by an accuracy of 95.12% against 87.26%. Figure 9.12 illustrates the validation of MML.

### *9.5.2 Color Image Segmentation*

Image segmentation is one of the core research topics and high-level tasks in the field of computer vision. The significance of this application is highlighted by the fact that it nourishes numerous applications progressively. We validated our

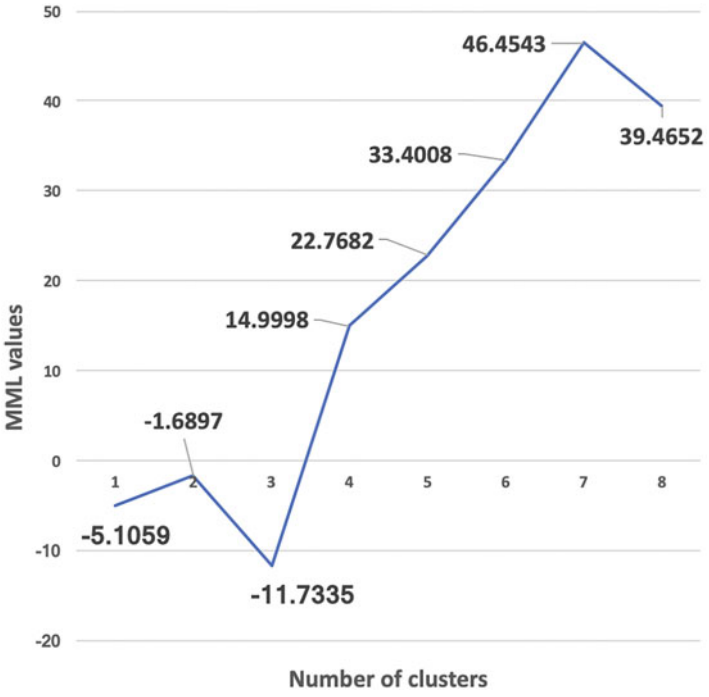



**Fig. 9.12** MML result for water levels of Florida swamps dataset

proposed framework by the well-known publicly available Berkeley segmentation data set [35]. This database is composed of a variety of natural color images generally used as a reliable way to compare image segmentation algorithms. It is noteworthy that the choice of color space is an important problem when dealing with color images and it is highly desirable that the chosen space be robust against varying illumination, conditions, and noise. We applied $rgb$ color space [29] described as below:

$$r = \frac{R}{R+G+B}; g = \frac{G}{R+G+B}; b = \frac{B}{R+G+B}; \tag{9.33}$$

As an example, Figs. 9.13, 9.14, 9.15 and 9.16 show original image, one labeled image by ground truth and a comparison between results obtained by BBMM and GMM. We compared the segmentation results of BBMM by six image segmentation evaluation metrics, Adjusted Rand Index (ARI) [36], Adjusted Mutual Information Score (AMIS) [37–40], Homogeneity Score (HS) [41, 42], Completeness Score (CS) [41, 42], Calinski-Harabaz Index (CHI) [43], Jaccard similarity score (JSS)[44–46] and their results are presented in Table 9.1. As it is shown, our model outperforms the GMM according to all metrics.

**Fig. 9.13** Original image 36046

**Fig. 9.14** Labeled image

**Fig. 9.15** GMM segmentation result

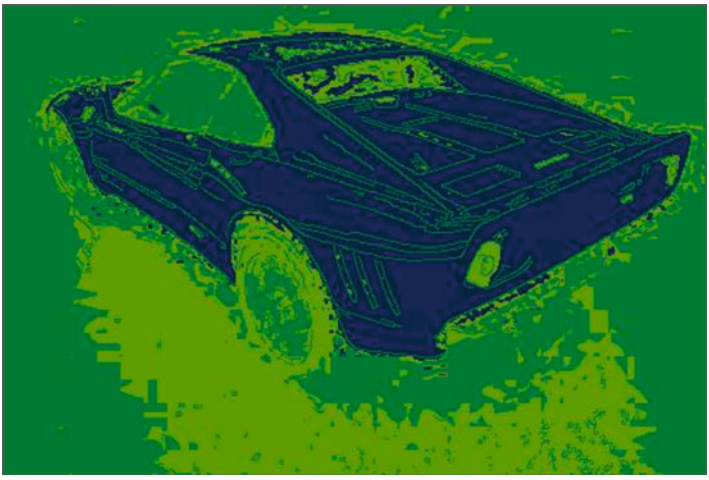


**Fig. 9.16** BBMM segmentation result

**Table 9.1** The results of testing model performance via image segmentation based on six metrics. K is the number of clusters

| | Metrics (K=4) | | | | | |
|---|---|---|---|---|---|---|
| Alg. | ARI | NMIS | MIS | HS | VM | JSS |
| BBMM | 0.52 | 0.44 | 0.48 | 0.42 | 0.46 | 0.55 |
| | 0.57 | 0.41 | 0.43 | 0.45 | 0.42 | 0.51 |
| | 0.55 | 0.45 | 0.41 | 0.41 | 0.43 | 0.55 |
| | 0.58 | 0.42 | 0.42 | 0.47 | 0.48 | 0.53 |
| | 0.56 | 0.41 | 0.45 | 0.48 | 0.41 | 0.57 |
| | 0.53 | 0.47 | 0.44 | 0.46 | 0.44 | 0.54 |
| Mean | **0.55** | **0.43** | **0.44** | **0.45** | **0.44** | **0.54** |
| GMM | 0.41 | 0.4 | 0.45 | 0.4 | 0.41 | 0.43 |
| | 0.44 | 0.4 | 0.4 | 0.41 | 0.4 | 0.46 |
| | 0.4 | 0.41 | 0.42 | 0.41 | 0.4 | 0.4 |
| | 0.42 | 0.4 | 0.41 | 0.4 | 0.42 | 0.44 |
| | 0.41 | 0.41 | 0.4 | 0.43 | 0.4 | 0.41 |
| | 0.43 | 0.42 | 0.41 | 0.41 | 0.4 | 0.45 |
| Mean | **0.42** | **0.4** | **0.41** | **0.41** | **0.4** | **0.43** |

### *9.5.3 Experimental Results for MBMM*

In this section, we will test the performance of our model on four real medical datasets and two real-world applications, namely sentiment analysis and credit approval using UCI datasets [34].

#### 9.5.3.1 Haberman Dataset

The first real dataset is a well-known one called Haberman based on a survival research at the University of Chicago's Billings Hospital between the years 1958 and 1970. It includes 306 instances of patients who had breast cancer and were monitored after having surgery. The dataset has three attributes: age of patient at time of operation, patient's year of operation, and number of positive axillary nodes detected [34]. The database is labeled based on survival status. The patients who survived 5 years or longer were classified in first class and the ones died within 5 years were second class. MBMM has a performance of 93.16% while the accuracy of GMM is 87.02%. The MML result is presented in Fig. 9.17.

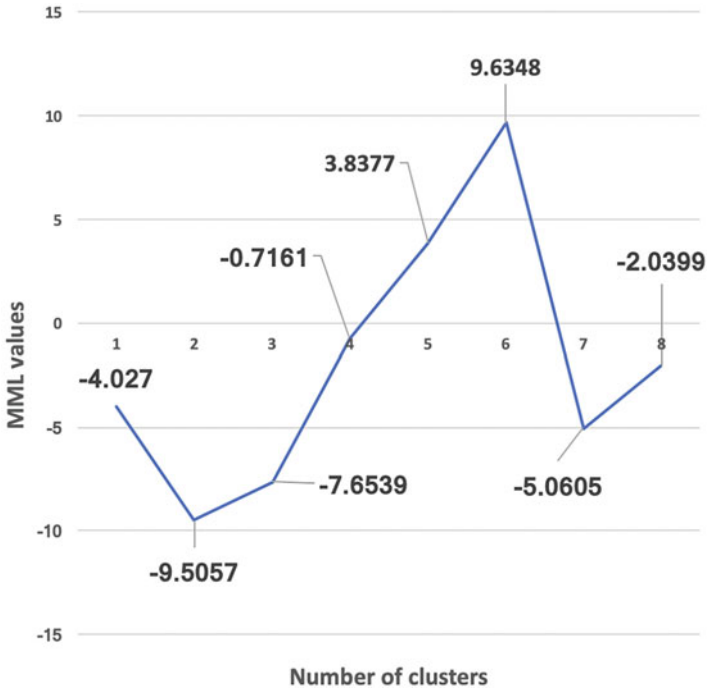



**Fig. 9.17** MML result for Haberman dataset

#### 9.5.3.2 Heart Disease Dataset

The second data set contains 2982 instances and based on the research conducted in V.A. medical center, Long Beach and Cleveland Clinic Foundation. It contains 76 attributes, but all published experiments refer to using a subset of 14 of them. The goal field refers to the presence of heart disease in the patient and experiments have concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0) with the help of angiographic disease status. The 14 attributes are age in years, sex, chest pain type (typical angina, atypical angina, non-anginal pain, and asymptomatic), resting blood pressure (in mm Hg on admission to the hospital), serum cholesterol in mg/dl, fasting blood sugar greater than 120 mg/dl, resting electrocardiographic results (normal, having ST-T wave abnormality or showing probable or definite left ventricular hypertrophy by Estes' criteria ), maximum heart rate achieved, exercise induced angina, oldpeak as the ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment (upsloping, flat or downsloping), number of major vessels (0–3) colored by fluoroscopy and thal (normal, fixed defect or reversible defect). MBMM has a performance of 92.41% while the accuracy is 90.83% for GMM. MML validation is displayed in Fig. 9.18.

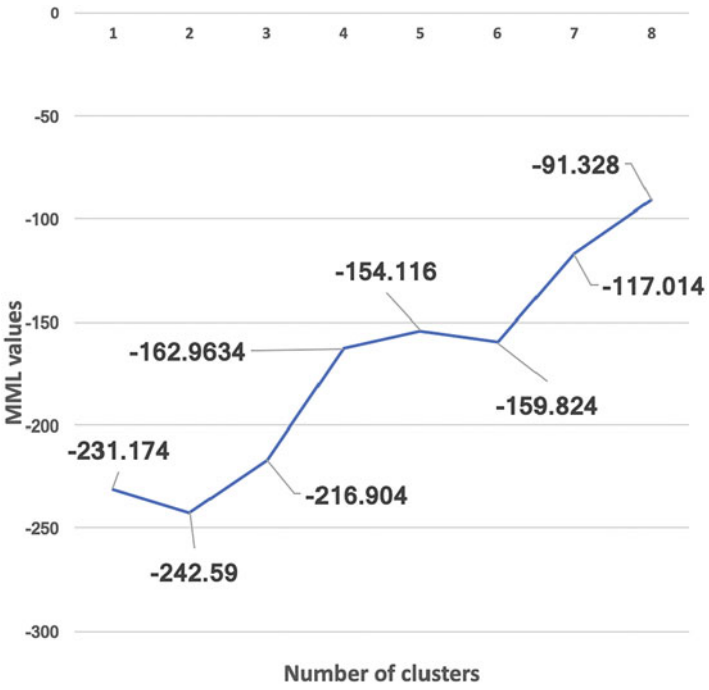



**Fig. 9.18** MML result for heart disease dataset

#### 9.5.3.3 Hepatitis Dataset

This data set includes 155 instances and 20 attributes including the class. The attributes are age, presence of steroid, antivirals administered, fatigue, malaise, anorexia, large liver, firm liver, spleen palpability, presence of spiders, presence of ascites, presence of varices, bilirubin level, alkaline phosphate level, SGOT level, albumin level, protein level and histology result [34]. The binary label indicates if the patient is alive or not. Our mixture model has an accuracy of 91.97% while the accuracy of GMM is 85.12%. The result of MML is presented in Fig. 9.19.

#### 9.5.3.4 Lymphography Dataset

This lymphography data set was obtained from the university medical centre, Institute of Oncology, Ljubljana, Yugoslavia including 148 instances and 19 attributes containing the label to lymphatic diseases [34]. The attributes are lymphatics (normal, arched, deformed, displaced), block of afferent, block of lymph c (superior and inferior flaps), block of lymph s (lazy incision), bypass, extravasates (force out of lymph), regeneration, early uptake, lymph nodes diminish, lymph nodes

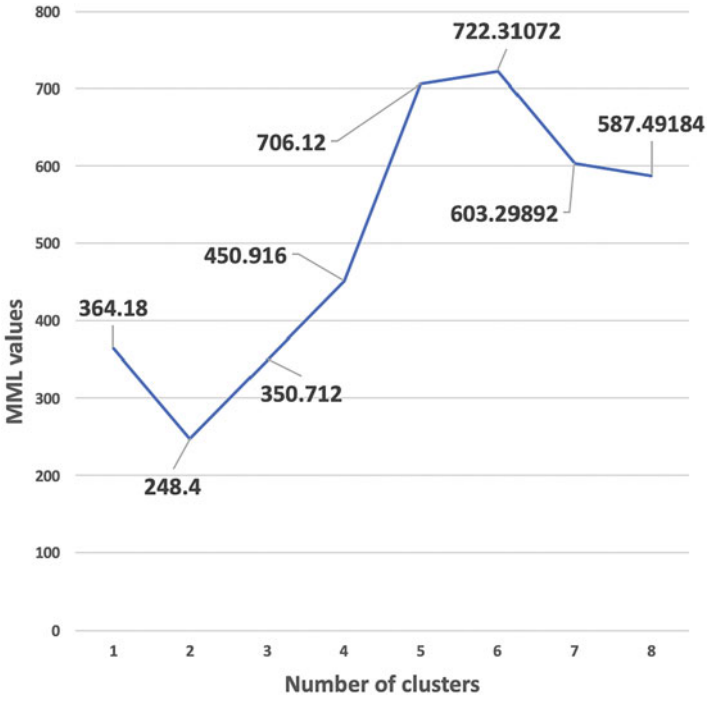



**Fig. 9.19** MML result for hepatitis dataset

enlargement, changes in lymph (bean, oval, round), defect in node (lacunar, lacunar marginal, lacunar central, or no defect), changes in structure (no change, grainy, drop-like, coarse, diluted, reticular, stripped, faint), special forms (no chalices, vesicles), dislocation, exclusion of nodes, and number of nodes. The target class has four values to show normal, metastases, malignant lymph and fibrosis. MBMM and GMM have accuracies of 95.06% and 92.64 %, respectively. MML is validated as shown in Fig. 9.20.

#### 9.5.3.5 Sentiments Analysis

There has been a recent tremendous attention in the automatic identification and extraction of feelings, views, and ideas in text. The crucial importance of information analysis for different sections of society such as scientific, commercial, financial, and political domains motivates the researchers to propose capable models which automatically track attitudes and feelings. Moreover, innovations in technology and digital revolution over the past two decades resulted in generating huge quantities of data. One of the attention-grabbing fields of research is online sources mining such as articles in the news, on-line forums, and websites to identify opinions and

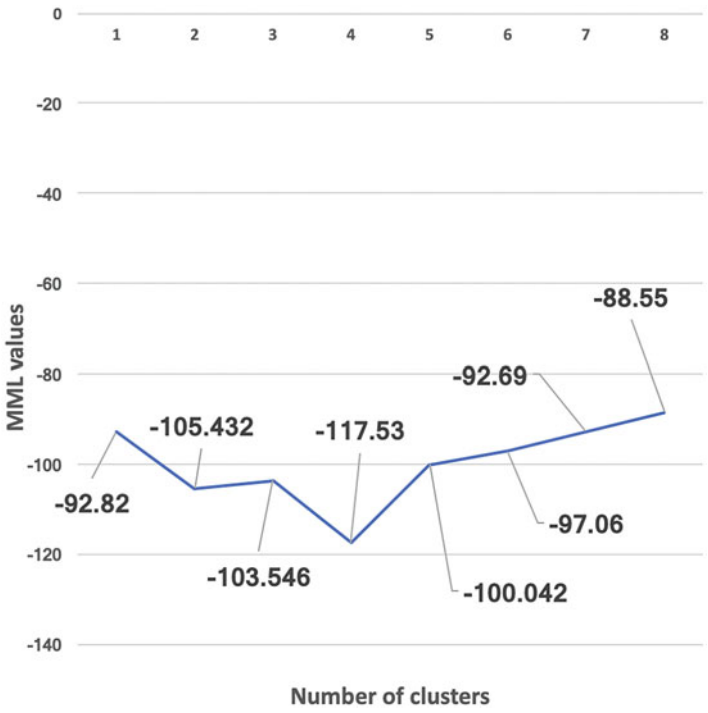



**Fig. 9.20** MML result for lymphography dataset

emotions from text. One of the applications that could benefit from this technology is explicitly recognizing and differentiating of the objective and subjective opinions and perspectives to lessen distractions from opinionated, speculative, and evaluative language as subjective sentences represent sentiment and belief in contrast with objective contents which represent factual information. We evaluate our model on this fascinating real-world application. Using a dataset from UCI repository, this work is based on analyzing 1000 sports articles which were labeled using Amazon Mechanical Turk as objective or subjective and has 60 attributes including total words count, total number of words in the article, semantic objective score, first sentence class, last sentence, text complexity score, and frequencies of words with an objective SENTIWORDNET score, words with a subjective SENTIWORDNET score, coordinating conjunctions, numerals and cardinals, determiners, existential there, foreign words, subordinating preposition or conjunction, ordinal adjectives or numerals, comparative adjectives, superlative adjectives, list item markers, modal auxiliaries, singular common nouns, singular proper nouns, plural proper nouns, plural common nouns, pre-determiners, genitive markers, personal pronouns, possessive pronouns, adverbs, comparative adverbs, superlative adverbs, particles, symbols, 'to' as preposition or infinitive marker, interjections, base form verbs, past tense verbs, present participle or gerund verbs, past participle verbs, present

tense verbs with plural 3rd person subjects, present tense verbs with singular 3rd person subjects, WH-determiners, WH-pronouns, possessive WH-pronouns, WH-adverb, infinitive verbs, quotation pairs in the entire article, questions marks in the entire article, exclamation marks in the entire article, full stops, commas, semicolons, colons, ellipsis, first person pronouns (personal and possessive), second person pronouns (personal and possessive), third person pronouns (personal and possessive), comparative and superlative adjectives and adverbs, past tense verbs with first and second person pronouns, imperative verbs, present tense verbs with third person pronouns and present tense verbs with first and second person [34]. The label indicates subjectivity and objectivity of each article. MBMM has an accuracy of 89.29% in comparison with GMM with 80.16% accuracy. MML result is displayed in Fig. 9.21.

#### 9.5.3.6 Credit Approval

Machine learning has been used in banking to automate daily processes such as fraud detection and risk assessment to make decision about credit cards and loans for years. Thanks to rapid increases in data generation and computing power,

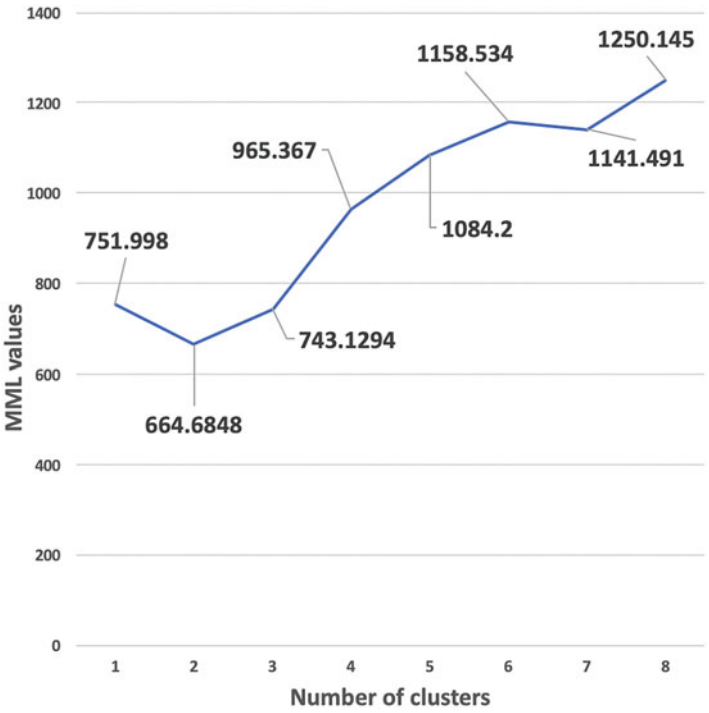



**Fig. 9.21** MML result for sentiments analysis dataset

financial institutions beneficiate from statistical algorithms to discover a better way to segment their customers and model credit applications. Using two datasets from UCI repository, we analyze the performance of our proposed methods in credit approval [34]. The first one is German credit dataset including 1000 instances in which the customers are described by a set of attributes and labeled as good or bad credit risks. The attributes are numerical, qualitative, and categorical. This file has been edited and several indicator variables added to make it suitable for algorithms which cannot cope with categorical variables. Several attributes that are ordered categorical have been coded as integers. The features contain status of existing checking account, duration in month, credit history, purpose, credit amount, savings account or bonds, present employment, installment rate in percentage of disposable income, personal status and sex, other debtors or guarantors, present residence, property, age in years, other installment plans, housing, number of existing credits at this bank, job, number of people being liable to provide maintenance for, telephone and foreign worker. The dataset has a binary label to indicate if the customer is good or bad. MBMM performs better with an accuracy of 94.12% in comparison with GMM which is 86.41% accurate. Figure 9.22 demonstrates the MML outcomes.

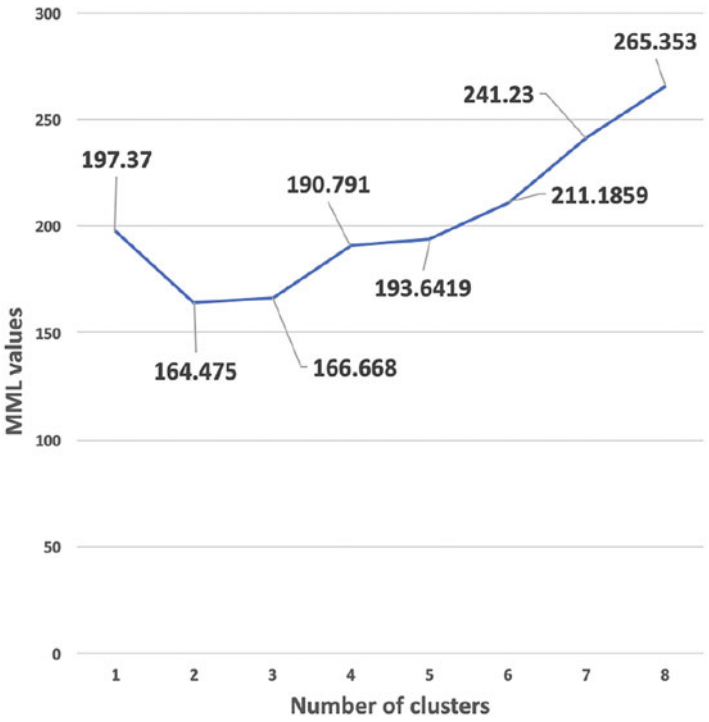



**Fig. 9.22** MML result for credit approval German dataset

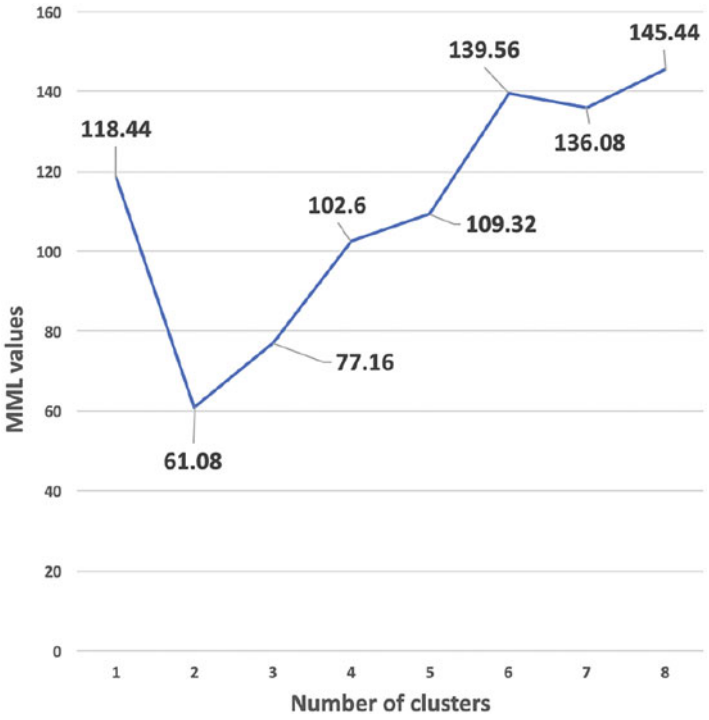



**Fig. 9.23** MML result for credit approval Australian dataset

The second dataset is Australian credit approval including 690 observations and 14 attributes containing six numerical and eight categorical features. All of attributes are changes to nominal values for the convenience of statistical algorithms. The label is binary in this case also. The accuracy of our model is higher than GMM with an accuracy of 92.58% versus 84.33%. MML analysis is displayed in Fig. 9.23.

## 9.6 Conclusion

This work is motivated principally by significant role of unsupervised methods in machine learning specifically clustering. We have presented two algorithms based on finite bivariate and multivariate Beta mixture models. GMM has been widely used in statistical modeling. However, when it comes to dealing with asymmetric data, other distributions demonstrate more flexibility for appropriate data modelling as compared with the Gaussian distribution. This characteristic of such distributions motivated us and we introduced the two new mixture models based on their ability to tackle the modeling of non-gaussian data. To address the challenge of

parameter estimation, we explored deterministic approaches such as maximum likelihood using the expectation maximization algorithm framework and Newton Raphson method to learn our framework and estimate the updated parameters of our mixture models. Given that in real-world applications, having an idea of the number of clusters inherent in the dataset is critical, a model selection technique, namely the minimum message length was implemented to determine the number of clusters which describes the model complexity. The validity of our proposed method in terms of parameter estimation and model selection is demonstrated by experimental results. The usefulness and strength of our method is presented by testing it on real and pre-labeled datasets and tackling challenging real-world applications such as image segmentation, sentiment analysis, and credit approval which have been receiving considerable attention because of their critical role in science and technology. For each application, we compared the performance of our approaches with Gaussian mixture models in describing real-world data. The test results significantly showed greater accuracy and outperformance of BBMM and MBMM. In other words, we can say that our model produces enhanced clustering results largely due to its flexibility. Future works could be devoted to proposing Bayesian inference techniques to learn the developed mixture models.

## Appendix

*Proof of Eq.* (9.12)*:*

$$
\begin{aligned}
\mathcal{L}(\Theta, Z, \mathcal{X}) &= \sum_{j=1}^{M}\sum_{n=1}^{N} \hat{Z}_{nj}\Big(\log p_j + \log p(\mathbf{X}_n|\mathbf{a}_j)\Big) \\
&= \sum_{j=1}^{M}\sum_{n=1}^{N} \hat{Z}_{nj}\Bigg(\log p_j + \log\Big(\frac{\prod_{i=1}^{d} X_{ni}^{(a_{ji}-1)}}{\prod_{i=1}^{d}(1-X_{ni})^{(a_{ji}+1)}} \\
&\qquad \times \Big[1+\sum_{i=1}^{d}\frac{X_{ni}}{(1-X_{ni})}\Big]^{-a_j} \times \frac{\Gamma(\sum_{i=1}^{d} a_{ji})}{\prod_{i=1}^{d}\Gamma(a_{ji})}\Big)\Bigg) \\
&= \sum_{j=1}^{M}\sum_{n=1}^{N} \hat{Z}_{nj}\Bigg(\log p_j + \log\Big(\prod_{i=1}^{d} X_{ni}^{(a_{ji}-1)}\Big) \\
&\qquad - \log\Big(\prod_{i=1}^{d}(1-X_{ni})^{(a_{ji}+1)}\Big) + \log\big(\Gamma(a_j)\big) \\
&\qquad - \log\prod_{i=1}^{d}\Gamma(a_{ji}) + \log\Big[1+\sum_{i=1}^{d}\frac{X_{ni}}{(1-X_{ni})}\Big]^{-a_j}\Bigg)
\end{aligned}
$$

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

## References


1. Diaz-Rozo, J., Bielza, C., Larrañaga, P.: Clustering of data streams with dynamic gaussian mixture models: an IoT application in industrial processes. IEEE Internet Things J. **5**(5), 3533 (2018)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
3. Olkin, I., Liu, R.: A bivariate beta distribution. Statist. Probab. Lett. **62**(4), 407–412 (2003)
4. Bouguila, N.: Clustering of count data using generalized Dirichlet multinomial distributions. IEEE Trans. Knowl. Data Eng. **20**(4), 462–474 (2008)
5. Bouguila, N., ElGuebaly, W.: Integrating spatial and color information in images using a statistical framework. Expert Syst. Appl. **37**(2), 1542–1549 (2010)
6. Bouguila, N., Ziou, D.: A Dirichlet process mixture of generalized Dirichlet distributions for proportional data modeling. IEEE Trans. Neural Netw. **21**(1), 107–122 (2010)
7. Bouguila, N.: Spatial color image databases summarization. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 953–956 (2007)
8. Bouguila, N., Amayri, O.: A discrete mixture-based kernel for SVMs: application to spam and image categorization. Inf. Process. Manag. **45**, 631–642 (2009)
9. Bouguila, N., Ziou, D.: Dirichlet-based probability model applied to human skin detection [image skin detection]. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 521–524 (2004)
10. Bouguila, N., Ziou, D.: Improving content based image retrieval systems using finite multinomial Dirichlet mixture. In: 14th IEEE Signal Processing Society Workshop Machine Learning for Signal Processing, pp. 23–32 (2004)
11. Bouguila, N., Ziou, D.: A powerful finite mixture model based on the generalized Dirichlet distribution: unsupervised learning and applications. In: 17th International Conference on Pattern Recognition, pp. 280–283 (2004)
12. Bouguila, N., Ziou, D.: Using unsupervised learning of a finite Dirichlet mixture model to improve pattern recognition applications. Pattern Recogn. Lett. **26**(12), 1916–1925 (2005)
13. Bouguila, N., Ziou, D.: A Dirichlet process mixture of Dirichlet distributions for classification and prediction. In: IEEE Workshop on Machine Learning for Signal Processing, pp. 297–302 (2008)

14. Bouguila, N., Ziou, D.: A countably infinite mixture model for clustering and feature selection. Knowl. Inf. Syst. **33**(2), 351–370 (2012)
15. Bouguila, N., Ziou, D., Hammoud, R.I.: On Bayesian analysis of a finite generalized Dirichlet mixture via a Metropolis-within-Gibbs sampling. Pattern. Anal. Applic. **12**(2) (2009)
16. Olkin, I., Trikalinos, T.A.: Constructions for a bivariate beta distribution. Statist. Probab. Lett. **96**, 54–60 (2015)
17. McLachlan, G.J.: Mixture models in statistics. In: International Encyclopedia of the Social and Behavioral Sciences, pp. 624–628 (2015)
18. Ganesaligman, S.: Classification and mixture approaches to clustering via maximum likelihood. Appl. Stat. **38**(3), 455–466 (1989)
19. McLachlan, G.J., Krishnan, T.: The EM Algorithm and Extensions. Wiley-Interscience, Hoboken (2008)
20. Bouguila, N., Ziou, D.: Unsupervised selection of a finite Dirichlet mixture model: an MML-based approach. IEEE Trans. Knowl. Data Eng. **18**(8), 993 (2006)
21. Bouguila, N., Ziou, D.: On fitting finite Dirichlet mixture using ECM and MML. In: Third International Conference on Advances in Pattern Recognition, vol. 3686, pp. 172–182 (2005)
22. Bouguila, N., Ziou, D.: MML-based approach for finite Dirichlet mixture estimation and selection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), pp. 42–51 (2005)
23. Bouguila, N., Ziou, D.: Online clustering via finite mixtures of Dirichlet and minimum message length. Eng. Appl. Artif. Intell. **19**, 371–379 (2006)
24. Bouguila, N., Ziou, D.: Unsupervised selection of a finite Dirichlet mixture model: an MML-based approach. IEEE Trans. Knowl. Data Eng. **18**(8), 993–1009 (2006)
25. Bouguila, N., Ziou, D.: High-dimensional unsupervised selection and estimation of a finite generalized Dirichlet mixture model based on minimum message length. IEEE Trans. Pattern Anal. Mach. Intell. **29**(10), 1716 (2007)
26. Wallace, C.S., Dowe, D.L.: MML clustering of multistate, Poisson, von Mises circular and gaussian distributions. Stat. Comput. **10**(1), 73–83 (2000)
27. Baxter, R.A.: Minimum Message Length Inference: Theory and Applications. Monash University, Clayton (1996)
28. Baxter, R.A., Oliver, J.J.: Finding overlapping components with MML. Stat. Comput. **3**(1), 5–16 (2000)
29. Sefidpour, A., Bouguila, N.: Spatial color image segmentation based on finite non-Gaussian mixture models. Expert Syst. Appl. **39**(10), 8993–9001 (2012)
30. Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **24**(3), 381–396 (2002)
31. Bouguila, N., Ziou, D.: High-dimensional unsupervised selection and estimation of a finite generalized Dirichlet mixture model based on minimum message length. IEEE Trans. Pattern Anal. Mach. Intell. **29**(10), 1716 (2007)
32. Agusta, Y., Dowe, D.L.: Unsupervised learning of gamma mixture models using minimum message length. In: Hamza, M.H. (ed.) Proceeding Third ASTED Conference Artificial Intelligence and Applications, pp. 457–462 (2003)
33. Jefferys, W.H., Berger, J.O.: Ockham's razor and Bayesian analysis. Am. Sci. **80**(1), 64–72 (1992)
34. UCI Repository Data Set (1999). https://archive.ics.uci.edu/ml/machine-learningdatabases. Accessed 2 August 1999
35. The Berkeley Segmentation Dataset and Benchmark Dataset [Online]. https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/
36. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (1985)
37. Strehl, A., Joydeep, G.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)
38. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison. In: Proceedings of the 26th Annual International Conference on Machine Learning-ICML (2009)

39. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J. Mach. Learn. Res. **11**, 2837–2854 (2010)
40. Yang, Z., Algesheimer, R., Tessone, C.J.: A comparative analysis of community detection algorithms on artificial networks. Sci. Rep. **6** (2016)
41. Rosenberg, A., Hirschberg, J.: V-Measure a conditional entropy-based external cluster evaluation measure. In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 410–420 (2007)
42. Becker, H.: Identification and Characterization of Events in Social Media. PhD Thesis (2011)
43. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Commun. Stat. Theory Methods **3**, 1–27 (1974)
44. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley Longman Publishing Co., Boston (2005)
45. Jaccard, P.: Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull. Soc. Vaud. Sci. Nat. **37**, 547–579 (1901)
46. Jaccard, P.: The Distribution of the flora in the alpine zone. New Phytol. **11**, 37–50 (1912)

# Chapter 10
# Finite Inverted Beta-Liouville Mixture Models with Variational Component Splitting


**Kamal Maanicshah, Muhammad Azam, Hieu Nguyen, Nizar Bouguila, and Wentao Fan**

**Abstract** Use of mixture models to statistically approximate data has been an interesting topic of research in unsupervised learning methods. Mixture models based on exponential family of distributions have gained popularity in recent years. In this chapter, we introduce a finite mixture model based on Inverted Beta-Liouville distribution which has a higher degree of freedom to provide a better fit for the data. We use a variational learning framework to estimate the parameters which decreases the computational complexity of the model. We handle the problem of model selection with a component splitting approach which is an added advantage as it is done within the variational framework. We evaluate our model against some challenging applications like image clustering, speech clustering, spam image detection, and software defect detection.

## 10.1 Introduction

Data categorization has become an important part of data analysis in recent years as data in all formats have been thriving with the increase in cloud based networks. Given a set of data it is of prime importance to learn the patterns within the data which can have valuable information that help in making important decisions. Clus-

K. Maanicshah (✉) · H. Nguyen · N. Bouguila
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: k_mathin@encs.concordia.ca; hi_guy@encs.concordia.ca; nizar.bouguila@concordia.ca

M. Azam
Department of Electrical and Computer Engineering (ECE), Concordia University, Montreal, QC, Canada
e-mail: mu_azam@encs.concordia.ca

W. Fan
Department of Computer Science and Technology, Huaqiao University, Xiamen, China
e-mail: fwt@hqu.edu.cn



N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6_10

tering of data hence is quintessential in data analysis and inference [20, 30]. Using mixture models for clustering and unsupervised learning finds wide applications in industry. Data are assumed to be described by a mixture of components derived from a particular distribution. The objective is to estimate the parameters of these components which would provide a proper fit to the data. This is the basic idea of mixture models [2, 7]. Gaussian mixture models (GMM) have disseminated the industry and have profound applications in a number of data analysis tasks [31, 35]. Despite widespread use, it is a notable fact that GMMs are not the perfect solution for all types of data. For example, Dirichlet family of distributions performs better with proportional data [4, 7]. Recent investigations on inverted Beta-Liouville based mixture models have shown their effectiveness [1, 15]. In this chapter we consider finite mixtures of inverted Beta-Liouville distributions.

We use a variational learning framework in our model which is a better choice when compared to the traditional methods based on maximum likelihood (ML) and Bayesian estimation techniques. This is because ML based methods do not provide sometimes good approximation to the data. In contrast to ML based methods, approaches based on Bayesian techniques such as Markov chain Monte Carlo (MCMC) tend to be more accurate but they are computationally expensive and moreover convergence is not guaranteed. Variational learning reduces the complexity of the Bayesian method and also overcomes the drawbacks of ML based models. Also, variational methods guarantee convergence. In addition to this, we imbue a component splitting approach within the variational framework as proposed in [10] for model selection. The effectiveness of this approach has been reported in [16] and [6]. According to this idea, we start with two clusters and then go on splitting each cluster into two based on a split criterion. The splitting continues until all clusters fail the split test. Since this algorithm works within the variational framework it improves the efficiency and flexibility of our model. There are many instances in the industry where there is not much data to explain a particular category. Our experiments suggest that our model can tackle this kind of imbalance in data effectively.

The rest of the chapter is classified as follows: Sect. 10.2 describes the mathematical model; Sect. 10.3 explains the variational learning approach; Sect. 10.4 reports the results obtained by using our model. The chapter concludes with Sect. 10.5.

## 10.2 The Statistical Model

In this section we introduce the inverted Beta-Liouville mixture model in Sect. 10.2.1 and we elaborate on the model selection approach using component splitting in Sect. 10.2.2.

### 10.2.1 Finite Inverted Beta-Liouville Mixture Models

Consider a $D$-dimensional vector $\mathbf{X}_i = \left(X_1, X_2, \ldots, X_D\right)$ drawn from a set of $N$ independent and identically distributed data samples $\mathcal{X} = \left(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\right)$ generated from an inverted Beta-Liouville (IBL) distribution [24]. Then, the probability density function $p\left(\mathbf{X}_i \mid \alpha_1, \ldots, \alpha_D, \alpha, \beta, \lambda\right)$ is given by:

$$p\left(\mathbf{X}_i \mid \alpha_{i1}, \ldots, \alpha_{iD}, \alpha, \beta, \lambda\right) = \frac{\Gamma\left(\sum_{l=1}^{D} \alpha_l\right)\Gamma\left(\alpha+\beta\right)}{\Gamma\left(\alpha\right)\Gamma\left(\beta\right)} \prod_{l=1}^{D} \frac{X_{il}^{\alpha_l - 1}}{\Gamma\left(\alpha_l\right)} \times \lambda^{\beta}\left(\sum_{l=1}^{D} X_{il}\right)^{\alpha - \sum_{l=1}^{D} \alpha_l} \left(\lambda + \sum_{l=1}^{D} X_{il}\right)^{-(\alpha+\beta)} \tag{10.1}$$

with the conditions $X_{il} > 0$ for $l = 1, .., D$, $\alpha > 0$, $\beta > 0$ and $\lambda > 0$. The mean, variance, and covariance of IBL distribution are given by:

$$E(X_{il}) = \frac{\lambda\alpha}{\beta - 1} \frac{\alpha_l}{\sum_{l=1}^{D} \alpha_l} \tag{10.2}$$

$$Var(X_{il}) = \frac{\lambda^2\alpha(\alpha+1)}{(\beta-1)(\beta-2)} \frac{\alpha_l(\alpha+1)}{\sum_{l=1}^{D} \alpha_l(\sum_{l=1}^{D} \alpha_l + 1)} \frac{\lambda^2\alpha^2}{(\beta-1)^2} \frac{\alpha_l^4}{(\sum_{l=1}^{D} \alpha_l)^4} \tag{10.3}$$

$$Cov(X_{im}, X_{in}) = \frac{\alpha_m\alpha_n}{\sum_{l=1}^{D} \alpha_l} \left[ \frac{\lambda^2\alpha(\alpha+1)}{(\beta-1)(\beta-2)(\sum_{l=1}^{D} \alpha_l + 1)} - \frac{\lambda^2\alpha^2}{(\beta-1)^2(\sum_{l=1}^{D} \alpha_l)} \right] \tag{10.4}$$

If we assume that each sample $X_i$ is picked from a mixture of IBL distributions, then the mixture model is represented as:

$$p\left(\mathcal{X} \mid \boldsymbol{\pi}, \Theta\right) = \sum_{i=1}^{N} \sum_{j=1}^{M} \pi_j \, p\left(\mathbf{X}_i \mid \theta_j\right) \tag{10.5}$$

where $M$ is the number of components in the mixture model and $\Theta = (\theta_1, \theta_2, \ldots, \theta_M)$. $p\left(\mathbf{X}_i \mid \theta_j\right)$ denotes the conditional probability of the data sample with respect to each component, $\theta_j = (\alpha_{j1}, \ldots, \alpha_{jD}, \alpha_j, \beta_j, \lambda_j)$ represents the parameter with respect to the component $j$ and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_M)$ is the set of mixing parameters and follows the conditions $\sum_{j=1}^{M} \pi_j = 1$ and $0 \leq \pi_j \leq 1$. We now introduce an indicator matrix $\mathcal{Z} = (\mathbf{Z}_1, \ldots, \mathbf{Z}_N)$ which indicates to which component each data sample is assigned to. Here $\mathbf{Z}_i = (Z_{i1}, \ldots, Z_{iM})$. $\mathbf{Z}_i$ is

a binary vector that satisfies the conditions $Z_{ij} \epsilon \{0, 1\}$ and $\sum_{j=1}^{M} Z_{ij} = 1$ and is defined by:

$$Z_{ij} = \begin{cases} 1, & \text{if } \mathbf{X}_i \epsilon j \\ 0, & \text{otherwise} \end{cases} \tag{10.6}$$

The conditional distribution of $\mathcal{Z}$ can thus be defined as:

$$p(\mathcal{Z} \mid \boldsymbol{\pi}) = \prod_{i=1}^{N} \prod_{j=1}^{M} \pi_j^{Z_{ij}} \tag{10.7}$$

Based on this equation we can write the conditional distribution of a data set $\mathcal{X}$ with respect to the clusters as:

$$p(\mathcal{X} \mid \mathcal{Z}, \Theta) = \sum_{i=1}^{N} \sum_{j=1}^{M} p(\mathbf{X}_i \mid \theta_j)^{Z_{ij}} \tag{10.8}$$

As we know that all the parameters are positive it would be good choice to model them using Gamma priors. Hence the priors are defined by:

$$p(\alpha_{jl}) = \mathcal{G}(\alpha_{jl} \mid u_{jl}, v_{jl}) = \frac{v_{jl}^{u_{jl}}}{\Gamma(u_{jl})} \alpha_{jl}^{u_{jl}-1} e^{-v_{jl}\alpha_{jl}} \tag{10.9}$$

$$p(\alpha_j) = \mathcal{G}(\alpha_j \mid p_j, q_j) = \frac{q_j^{p_j}}{\Gamma(p_j)} \alpha_j^{p_j-1} e^{-q_j\alpha_j} \tag{10.10}$$

$$p(\beta_j) = \mathcal{G}(\beta_j \mid g_j, h_j) = \frac{h_j^{g_j}}{\Gamma(g_j)} \beta_j^{g_j-1} e^{-h_j\beta_j} \tag{10.11}$$

$$p(\lambda_j) = \mathcal{G}(\lambda_j \mid s_j, t_j) = \frac{t_j^{s_j}}{\Gamma(s_j)} \lambda_j^{s_j-1} e^{-t_j\lambda_j} \tag{10.12}$$

where $\mathcal{G}(\cdot)$ represents a Gamma distribution and all the hyperparameters in the above priors are positive.

### *10.2.2 Component Splitting for Model Selection*

In our model we use a local model selection approach as proposed in [10]. This approach has also been successfully deployed in [6, 14, 16]. We propose to follow

this design for IBL models. The idea is to split the components in the mixture into two different sets; one called the *free components* and the other called *fixed components*. We constrain the model selection process to the free components without disturbing the fixed components. If we assume the fixed components to be a perfect approximation of the data, let's say $M - s$ components, then we have to approximate the mixing weights of the $s$ free components. Based on this concept we can rewrite the prior of $\mathcal{Z}$ from Eq. (10.7) as:

$$p(\mathcal{Z} \mid \boldsymbol{\pi}, \boldsymbol{\pi}^*) = \prod_{i=1}^{N} \left[ \prod_{j=1}^{s} \pi_j^{Z_{ij}} \prod_{j=s+1}^{M} \pi_j^{*Z_{ij}} \right] \tag{10.13}$$

where $\{\pi_j\}$ indicates the mixing coefficients of the free components and $\{\pi_j^*\}$ indicates the mixing coefficients of fixed components. It is to be noted that these mixing coefficients are always positive and follow the constraint:

$$\sum_{j=1}^{s} \pi_j + \sum_{j=s+1}^{M} \pi_j^* = 1 \tag{10.14}$$

According to this condition, though we only estimate the parameters of the free components $\{\pi_j\}$ we have to update the fixed parameters $\{\pi_j^*\}$ as well. Hence, $\{\pi_j^*\}$ is more like a random variable. So, we choose a prior distribution for the fixed components that depends on the free components. We choose the prior to be a non-standard Dirichlet distribution as it has been found to be optimal in [10]. The conditional probability of the fixed coefficients given the free coefficients can be written as:

$$p(\boldsymbol{\pi}^* \mid \boldsymbol{\pi}) = \left(1 - \sum_{k=1}^{s} \pi_k\right)^{-M+s} \frac{\Gamma\left(\sum_{j=s+1}^{M} c_j\right)}{\prod_{j=s+1}^{M} \Gamma(c_j)} \prod_{j=s+1}^{M} \left(\frac{\pi_j^*}{1 - \sum_{k=1}^{s} \pi_k}\right)^{c_j - 1} \tag{10.15}$$

The graphical representation of our model is shown in Fig. 10.1. Based on all the information we have so far, we can write the joint distribution for our model as:

$$p(\mathcal{X}, \mathcal{Z}, \Theta, \boldsymbol{\pi}^* \mid \boldsymbol{\pi}) = p(\mathcal{X} \mid \mathcal{Z}, \Theta) p(\mathcal{Z} \mid \boldsymbol{\pi}, \boldsymbol{\pi}^*) p(\boldsymbol{\pi}^* \mid \boldsymbol{\pi}) p(\alpha) p(\boldsymbol{\alpha}) p(\boldsymbol{\beta}) p(\boldsymbol{\lambda}) \tag{10.16}$$

$$= \prod_{i=1}^{N} \prod_{j=1}^{M} \left[ \frac{\Gamma\left(\sum_{l=1}^{D} \alpha_{jl}\right) \Gamma(\alpha_j + \beta_j)}{\Gamma(\alpha_j) \Gamma(\beta_j)} \prod_{l=1}^{D} \frac{X_{il}^{\alpha_{jl}-1}}{\Gamma(\alpha_{jl})} \right.$$

$$\left. \times \lambda_j^{\beta_j} \left(\sum_{l=1}^{D} X_{il}\right)^{\alpha_j - \sum_{l=1}^{D} \alpha_{jl}} \left(\lambda_j + \sum_{l=1}^{D} X_{il}\right)^{-(\alpha_j + \beta_j)} \right]^{Z_{ij}}$$

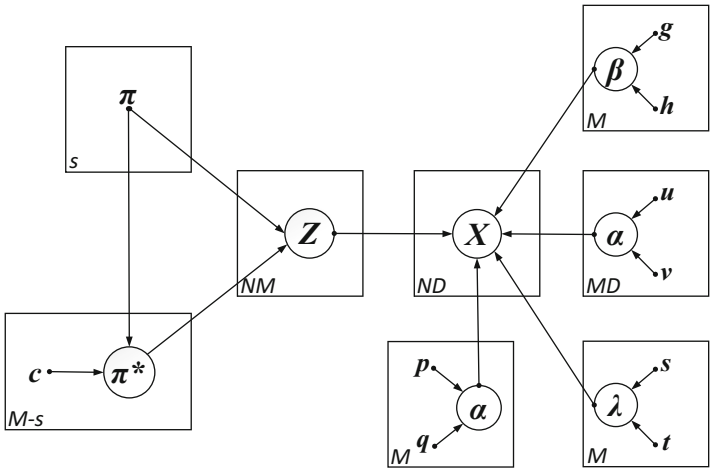



**Fig. 10.1** Graphical representation of IBL mixture model with component splitting. The circles indicate the random variables and model parameters, and plates point out the repetitions with the number in the lower left corners indicating the number of repetitions. The arcs specify the conditional dependencies of the variables

$$
\begin{aligned}
&\times \prod_{i=1}^{N}\left[\prod_{j=1}^{s}\pi_j^{Z_{ij}}\prod_{j=s+1}^{M}\pi_j^{*Z_{ij}}\right]\times\left(1-\sum_{k=1}^{s}\pi_k\right)^{-M+s}\\
&\times\frac{\Gamma\left(\sum_{j=s+1}^{M}c_j\right)}{\prod_{j=s+1}^{M}\Gamma\left(c_j\right)}\prod_{j=s+1}^{M}\left(\frac{\pi_j^*}{1-\sum_{k=1}^{s}\pi_k}\right)^{c_j-1}\\
&\times\prod_{j=1}^{M}\prod_{l=1}^{D}\left[\frac{v_{jl}^{u_{jl}}}{\Gamma\left(u_{jl}\right)}\alpha_{jl}^{u_{jl}-1}e^{-v_{jl}\alpha_{jl}}\times\frac{q_j^{p_j}}{\Gamma\left(p_j\right)}\alpha_j^{p_j-1}e^{-q_j\alpha_j}\right.\\
&\left.\times\frac{h_j^{g_j}}{\Gamma\left(g_j\right)}\beta_j^{g_j-1}e^{-h_j\beta_j}\times\frac{t_j^{s_j}}{\Gamma\left(s_j\right)}\lambda_j^{s_j-1}e^{-t_j\lambda_j}\right] \qquad (10.17)
\end{aligned}
$$

## 10.3 Variational Inference

In this section, we explain the variational learning approach in Sect. 10.3.1 and the component splitting approach with the variational algorithm is illustrated in Sect. 10.3.2.

### 10.3.1 Variational Learning

Though the Bayesian approach gives an accurate estimation for the parameters the complication lies in the calculation of the true posterior distribution $p(\Theta \mid \mathcal{X}, \boldsymbol{\pi})$. Calculating the integrals of the posterior may be intractable [5]. A wise solution for this problem would be to approximate the true posterior distribution rather than compute it. The accuracy by this method might not be as good as Bayesian estimation but it greatly reduces the computation cost and gives fairly comparable results. So, we approximate a new distribution $Q(\Theta)$ to be closer to the true posterior distribution $p(\Theta \mid \mathcal{X}, \boldsymbol{\pi})$. We do this by calculating the Kullback-Leibler (KL) divergence between the two distributions. The KL divergence estimates how much the two distributions are different from each other. The KL divergence between $Q(\Theta)$ and $p(\Theta \mid \mathcal{X}, \boldsymbol{\pi})$ is given by:

$$KL(Q \mid\mid P) = -\int Q(\Theta) \ln \left( \frac{p(\Theta \mid \mathcal{X}, \boldsymbol{\pi})}{Q(\Theta)} \right) d\Theta \tag{10.18}$$

which can be rewritten as:

$$KL(Q \mid\mid P) = \ln\ p(\mathcal{X} \mid \boldsymbol{\pi}) - \mathcal{L}(Q) \tag{10.19}$$

where $\mathcal{L}(Q)$ is the lower bound defined by:

$$\mathcal{L}(Q) = \int Q(\Theta) \ln \left( \frac{p(\mathcal{X}, \Theta \mid \boldsymbol{\pi})}{Q(\Theta)} \right) d\Theta \tag{10.20}$$

We know that $KL(Q \mid\mid P) = 0$ when both the distributions are similar. Based on Eq. (10.19) we can say this criteria can be achieved when $\mathcal{L}(Q)$ is maximized. We cannot calculate the lower bound for $Q(\Theta)$ altogether. Since all the parameters are assumed to be identically independent as well, we can use mean field theory [26] to factorize $Q(\Theta)$ as:

$$Q(\Theta) = Q(\mathcal{Z})Q(\alpha)Q(\boldsymbol{\alpha})Q(\boldsymbol{\beta})Q(\boldsymbol{\lambda})Q(\boldsymbol{\pi}^*) \tag{10.21}$$

Now, we find the variational solution for each of the parameters. For example, if we consider some parameter $Q_k(\Theta_k)$ the variational solution is given by:

$$Q_k(\Theta_k) = \frac{exp\langle \ln\ p(\mathcal{X}, \Theta) \rangle_{\neq k}}{\int exp\langle \ln\ p(\mathcal{X}, \Theta) \rangle_{\neq k} d\Theta} \tag{10.22}$$

where $\langle \cdot \rangle_{\neq k}$ is the expectation corresponding to all the parameters other than $\Theta_k$. The variational approximation involves initiating the variational solutions in the beginning and iteratively updating the solutions based on Eq. (10.22). Since the

lower bound is convex apropos to each of the parameters, convergence is always guaranteed. However, this is not usually the case when it comes to purely Bayesian approaches. We can derive the variational solutions for our model as shown in appendix as:

$$Q(\mathcal{Z}) = \prod_{i=1}^{N}\Bigg[\prod_{j=1}^{s} r_{ij}^{Z_{ij}} \prod_{j=s+1}^{M} r_{ij}^{*Z_{ij}}\Bigg] \tag{10.23}$$

$$Q(\boldsymbol{\pi}^*) = \Bigg(1 - \sum_{k=1}^{s}\pi_k\Bigg)^{-M+s} \frac{\Gamma\big(\sum_{j=s+1}^{M} c_j^*\big)}{\prod_{j=s+1}^{M}\Gamma\big(c_j^*\big)} \prod_{j=s+1}^{M}\Bigg(\frac{\pi_j^*}{1-\sum_{k=1}^{s}\pi_k}\Bigg)^{c_j^*-1} \tag{10.24}$$

$$Q(\alpha) = \prod_{j=1}^{M}\prod_{l=1}^{D}\mathcal{G}\big(\alpha_{jl} \mid u_{jl}^*, v_{jl}^*\big), \qquad Q(\boldsymbol{\alpha}) = \prod_{j=1}^{M}\mathcal{G}\big(\alpha_j \mid p_j^*, q_j^*\big) \tag{10.25}$$

$$Q(\boldsymbol{\beta}) = \prod_{j=1}^{M}\mathcal{G}\big(\beta_j \mid g_j^*, h_j^*\big), \qquad Q(\boldsymbol{\lambda}) = \prod_{j=1}^{M}\mathcal{G}\big(\lambda_j \mid s_j^*, t_j^*\big) \tag{10.26}$$

where:

$$r_{ij} = \frac{\tilde{r}_{ij}}{\sum_{j=1}^{s}\tilde{r}_{ij} + \sum_{j=s+1}^{M}\tilde{r}_{ij}^*}, \qquad r_{ij}^* = \frac{\tilde{r}_{ij}^*}{\sum_{j=1}^{s}\tilde{r}_{ij} + \sum_{j=s+1}^{M}\tilde{r}_{ij}^*} \tag{10.27}$$

$$\begin{aligned}\tilde{r}_{ij} = \exp\Bigg\{ &\ln\pi_j + R_j + S_j + \Big(\overline{\alpha}_j - \sum_{l=1}^{D}\overline{\alpha}_{jl}\Big)\ln\Big(\sum_{l=1}^{D} X_{il}\Big) + \overline{\beta}_j\langle\ln\lambda_j\rangle \\ &+ \sum_{l=1}^{D}\Big[\big(\overline{\alpha}_{jd}-1\big)\ln X_{id}\Big] - \big(\overline{\alpha}+\overline{\beta}\big)T_{ij}\Bigg\}\end{aligned} \tag{10.28}$$

$$\begin{aligned}\tilde{r}_{ij}^* = \exp\Bigg\{ &\langle\ln\pi_j^*\rangle + R_j + S_j + \Big(\overline{\alpha}_j - \sum_{l=1}^{D}\overline{\alpha}_{jl}\Big)\ln\Big(\sum_{l=1}^{D} X_{il}\Big) + \overline{\beta}_j\langle\ln\lambda_j\rangle \\ &+ \sum_{l=1}^{D}\Big[\big(\overline{\alpha}_{jd}-1\big)\ln X_{id}\Big] - \big(\overline{\alpha}+\overline{\beta}\big)T_{ij}\Bigg\}\end{aligned} \tag{10.29}$$

$$R_j = \ln \frac{\Gamma\left(\sum_{l=1}^{D} \overline{\alpha}_{jl}\right)}{\prod_{l=1}^{D} \Gamma\left(\overline{\alpha}_{jl}\right)} + \sum_{l=1}^{D} \overline{\alpha}_{jl} \left[ \psi\left( \sum_{l=1}^{D} \overline{\alpha}_{jl} \right) - \psi\left(\overline{\alpha}_{jl}\right) \right] \left[ \left\langle \ln \alpha_{jl} \right\rangle - \ln \overline{\alpha}_{jl} \right]$$
$$+ \frac{1}{2} \sum_{l=1}^{D} \overline{\alpha}_{jl}^2 \left[ \psi'\left( \sum_{l=1}^{D} \overline{\alpha}_{jl} \right) - \psi'\left(\overline{\alpha}_{jl}\right) \right] - \left\langle \left( \ln \alpha_{jl} - \ln \overline{\alpha}_{jl} \right)^2 \right\rangle$$
$$+ \frac{1}{2} \sum_{a=1}^{D} \sum_{b=1}^{D} \overline{\alpha}_{ja}\, \overline{\alpha}_{jb} \left[ \psi'\left( \sum_{l=1}^{D} \overline{\alpha}_{jl} \right) \left( \left\langle \ln \alpha_{ja} \right\rangle - \ln \overline{\alpha}_{ja} \right) \right.$$
$$\left. \times \left( \left\langle \ln \alpha_{jb} \right\rangle - \ln \overline{\alpha}_{jb} \right) \right] \tag{10.30}$$

$$S = \ln \frac{\Gamma(\overline{\boldsymbol{\alpha}} + \overline{\boldsymbol{\beta}})}{\Gamma(\overline{\boldsymbol{\alpha}})\Gamma(\overline{\boldsymbol{\beta}})} + \overline{\boldsymbol{\alpha}} \left[ \psi(\overline{\boldsymbol{\alpha}} + \overline{\boldsymbol{\beta}}) - \psi(\overline{\boldsymbol{\alpha}}) \right] \left( \left\langle \ln \boldsymbol{\alpha} \right\rangle - \ln \overline{\boldsymbol{\alpha}} \right)$$
$$+ \overline{\boldsymbol{\beta}} \left[ \psi(\overline{\boldsymbol{\alpha}} + \overline{\boldsymbol{\beta}}) - \psi(\overline{\boldsymbol{\beta}}) \right] \left( \left\langle \ln \boldsymbol{\beta} \right\rangle - \ln \overline{\boldsymbol{\beta}} \right)$$
$$+ 0.5 \overline{\boldsymbol{\alpha}}^2 \left[ \psi'(\overline{\boldsymbol{\alpha}} + \overline{\boldsymbol{\beta}}) - \psi'(\overline{\boldsymbol{\alpha}}) \right] \left\langle (\ln \boldsymbol{\alpha} - \ln \overline{\boldsymbol{\alpha}})^2 \right\rangle$$
$$+ 0.5 \overline{\boldsymbol{\beta}}^2 \left[ \psi'(\overline{\boldsymbol{\alpha}} + \overline{\boldsymbol{\beta}}) - \psi'(\overline{\boldsymbol{\beta}}) \right] \left\langle (\ln \boldsymbol{\beta} - \ln \overline{\boldsymbol{\beta}})^2 \right\rangle$$
$$+ \overline{\boldsymbol{\alpha}}\, \overline{\boldsymbol{\beta}} \psi'(\overline{\boldsymbol{\alpha}} + \overline{\boldsymbol{\beta}}) \left( \left\langle \ln \boldsymbol{\alpha} \right\rangle - \ln \overline{\boldsymbol{\alpha}} \right) \left( \left\langle \ln \boldsymbol{\beta} \right\rangle - \ln \overline{\boldsymbol{\beta}} \right) \tag{10.31}$$

$$T_{ij} = \ln \left[ \overline{\lambda}_j + \sum_{l=1}^{D} X_{il} \right] + \frac{\overline{\lambda}_j}{\overline{\lambda}_j + \sum_{l=1}^{D} X_{il}} \left[ \left\langle \ln \lambda_j \right\rangle - \ln \overline{\lambda}_j \right] \tag{10.32}$$

$$c_j^* = \sum_{i=1}^{N} r_{ij}^* + c_j \tag{10.33}$$

$$u_{jl}^* = u_{jl} + \sum_{i=1}^{N} \langle Z_{ij} \rangle \overline{\alpha}_{jl} \left[ \psi\left( \sum_{l=1}^{D} \overline{\alpha}_{jl} \right) - \psi\left(\overline{\alpha}_{jl}\right) \right. \tag{10.34}$$
$$\left. + \psi'\left( \sum_{l=1}^{D} \overline{\alpha}_{jl} \right) \sum_{d \neq l}^{D} \left( \langle \ln \alpha_{jl} \rangle - \ln \overline{\alpha}_{jl} \right) \overline{\alpha}_{jl} \right]$$

$$v_{jl}^* = v_{jl} - \sum_{i=1}^{N} \langle Z_{ij} \rangle \left[ \ln X_{il} - \ln \left( \sum_{l=1}^{D} X_{il} \right) \right] \tag{10.35}$$

$$p_j^* = p_j + \sum_{l=1}^{D} \langle Z_{ij} \rangle \left[ \psi\left(\overline{\boldsymbol{\alpha}}_j + \overline{\boldsymbol{\beta}}_j\right) - \psi\left(\overline{\boldsymbol{\alpha}}_j\right) + \overline{\boldsymbol{\beta}}_j \psi'\left(\overline{\boldsymbol{\alpha}}_j + \overline{\boldsymbol{\beta}}_j\right) \left( \langle \ln \beta_j \rangle - \overline{\boldsymbol{\beta}}_j \right) \right] \overline{\boldsymbol{\alpha}}_j \tag{10.36}$$

$$q_j^* = q_j - \sum_{i=1}^{N} \langle Z_{ij} \rangle \ln \left( \sum_{l=1}^{D} X_{id} \right) + \sum_{i=1}^{N} \langle Z_{ij} \rangle T_{ij} \tag{10.37}$$

$$g_j^* = g_j + \sum_{l=1}^{D} \langle Z_{ij} \rangle \left[ \psi\left(\overline{\alpha}_j + \overline{\beta}_j\right) - \psi\left(\overline{\beta}_j\right) + \overline{\alpha}_j \psi'\left(\overline{\alpha}_j + \overline{\beta}_j\right) \left( \langle \ln \alpha_j \rangle - \overline{\alpha}_j \right) \right] \overline{\beta}_j$$

$$h_j^* = h_j + \sum_{i=1}^{N} \langle Z_{ij} \rangle \left[ T_{ij} - \langle \ln \lambda_j \rangle \right] \tag{10.38}$$

$$s_j^* = s_j + \sum_{i=1}^{N} \langle Z_{ij} \rangle \overline{\beta}_j \tag{10.39}$$

$$t_j^* = t_j + \sum_{i=1}^{N} \langle Z_{ij} \rangle \frac{\overline{\alpha}_j + \overline{\beta}_j}{\overline{\lambda}_j + \sum_{l=1}^{D} X_{il}} \tag{10.40}$$

The first and second derivatives of the Gamma function are given by the digamma and trigamma functions, $\psi(\cdot)$ and $\psi'(\cdot)$ respectively. The values of the expectations mentioned in the above equations are given by:

$$\langle Z_{ij} \rangle = \begin{cases} r_{ij}, & \text{for} j = 1, \ldots, s \\ r_{ij}^*, & \text{otherwise} \end{cases} \tag{10.41}$$

$$\overline{\alpha}_{jl} = \langle \alpha_{jl} \rangle = \frac{u_{jl}}{v_{jl}}, \quad \overline{\alpha}_j = \langle \alpha_j \rangle = \frac{p_j}{q_j}, \quad \overline{\beta}_j = \langle \beta_j \rangle = \frac{g_j}{h_j}, \quad \overline{\lambda}_j = \langle \lambda_j \rangle = \frac{s_j}{t_j} \tag{10.42}$$

$$\langle \ln \alpha_{jl} \rangle = \psi(u_{jl}^*) - \ln v_{jl}^*, \qquad \langle \ln \alpha_j \rangle = \psi(p_j^*) - \ln q_j^*, \tag{10.43}$$

$$\langle \ln \beta_j \rangle = \psi(g_j^*) - \ln h_j^*, \qquad \langle \ln \lambda_j \rangle = \psi(s_j^*) - \ln t_j^* \tag{10.44}$$

$$\left\langle \left( \ln \alpha_{jl} - \ln \overline{\alpha}_{jl} \right)^2 \right\rangle = \left[ \psi\left(u_{jl}^*\right) - \ln u_{jl}^* \right]^2 + \psi'\left(u_{jl}^*\right) \tag{10.45}$$

$$\left\langle \left( \ln \alpha_j - \ln \overline{\alpha}_j \right)^2 \right\rangle = \left[ \psi\left(p_j^*\right) - \ln p_j^* \right]^2 + \psi'\left(p_j^*\right) \tag{10.46}$$

$$\left\langle \left( \ln \beta_j - \ln \overline{\beta}_j \right)^2 \right\rangle = \left[ \psi\left(g_j^*\right) - \ln g_j^* \right]^2 + \psi'\left(g_j^*\right) \tag{10.47}$$

$$\langle \pi_j^* \rangle = \left(1 - \sum_{k=1}^{s} \pi_k\right) \frac{\sum_{i=1}^{N} r_{ij}^* + c_j}{\sum_{i=1}^{N} \sum_{k=s+1}^{M} r_{ik}^* + c_k} \tag{10.48}$$

$$\left\langle \ln \pi_j^* \right\rangle = \ln\left(1 - \sum_{k=1}^{s} \pi_k\right) + \psi\left(\sum_{i=1}^{N} r_{ij}^* + c_j\right) - \psi\left(\sum_{i=1}^{N} \sum_{k=s+1}^{M} r_{ik}^* + c_k\right) \tag{10.49}$$

As mentioned before, since $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^*$ are bound together by the constraint that they sum up to 1, we have to find a proper update equation of $\pi_j$ with respect to $\pi_j^*$. We do this by differentiating the lower bound with respect to $\pi_j$ and equating it to 0. Hence the update equation can be written as:

$$\pi_j = \left(1 - \sum_{k=s+1}^{M} \langle \pi_k^* \rangle\right) \frac{\sum_{i=1}^{N} r_{ij}}{\sum_{i=1}^{N} \sum_{k=1}^{s} r_{ik}} \tag{10.50}$$

Based on the above update equations, we can calculate the lower bound by:

$$\mathcal{L}(Q) = \langle \ln p(\mathcal{X} \mid \mathcal{Z}, \Theta) \rangle + \langle \ln p(\mathcal{Z} \mid \boldsymbol{\pi}, \boldsymbol{\pi}^*) \rangle + \langle \ln p(\boldsymbol{\pi}^* \mid \boldsymbol{\pi}) \rangle \tag{10.51}$$

$$- \langle \ln p(\Theta) \rangle - \langle \ln Q(\mathcal{Z}) \rangle - \langle \ln Q(\Theta) \rangle \tag{10.52}$$

Our algorithm to maximize the lower bound and estimate the number of components in the mixture simultaneously follows in the next subsection.

### *10.3.2 Component Splitting Algorithm*

Generally, in the variational approach, we update the variational solutions described in the previous subsection iteratively until there is no notable change seen between two consecutive iterations. However, in [10] a split and merge approach is adopted. According to this approach we start the algorithm with two clusters initially. This criteria is checked first by running the algorithm without local model selection. We now split one of the mixture components into two and label them as *free component* in our procedure. The remaining components are the *fixed components*. Now we run our variational algorithm estimating the parameters of the free components without estimating the fixed components. In this procedure there are three different possibilities: (1) the two new components may fit the data appropriately and hence are retained. In this case, the split is a success and the algorithm is rerun for the new set of components with the added component. (2) Only one of the components might have a significant mixing coefficient and the other might be infinitesimal. This means that the split is a failure and the algorithm moves to the next component for the splitting. (3) In this case the estimate of the mixing coefficient for both

the free components goes infinitesimal. This is due to the presence of outliers in the data. We bar this split from happening as it might end up in an infinite loop. These components can be later removed as they are just unnecessary outliers. The algorithm terminates when all the components in the current set fail the split test. The overall algorithm can be epitomized as:

1. Initialization
   - Initialize number of components $M$ to 2
   - Initialize values for $\mathbf{u}$, $\mathbf{p}$, $\mathbf{g}$, $\mathbf{s}$ and $\mathbf{c}$ with 1 and $\boldsymbol{\nu}$, $\mathbf{q}$, $\mathbf{h}$ and $\mathbf{t}$ with 0.01.
2. Start the variational inference without the local model selection
3. If only one component remains, the algorithm ends
4. Sort all the elements in $M$ in descending order by their mixing coefficients.
5. For each element $j$ in $M$:
   - Split $j$ into $j_j$ and $j_2$ as the free components
   - set $\pi_{j_1} = \pi_{j_2} = \pi_j/2$, $u_{jl_1} = u^*_{jl}$, $u_{jl_2} = u^*_{jl}$, $v_{jl_1} = v^*_{jl}$, $v_{jl_2} = v^*_{jl}$, $p_{jl_1} = p^*_{jl}$, $p_{jl_2} = p^*_{jl}$, $q_{jl_1} = q^*_{jl}$, $q_{jl_2} = q^*_{jl}$, $g_{jl_1} = g^*_{jl}$, $g_{jl_2} = g^*_{jl}$, $h_{jl_1} = h^*_{jl}$, $h_{jl_2} = h^*_{jl}$, $s_{jl_1} = s^*_{jl}$, $s_{jl_2} = s^*_{jl}$, $t_{jl_1} = t^*_{jl}$ and $t_{jl_2} = t^*_{jl}$.
   - $c^*_j = \sum_{i=1}^{N} r^*_{ij}$ for each $j$ in the fixed components
   - Apply variational inference with component splitting by updating $Q(\mathcal{Z})$, $Q(\boldsymbol{\pi}^*)$, $Q(\alpha)$, $Q(\boldsymbol{\alpha})$, $Q(\boldsymbol{\beta})$, $Q(\boldsymbol{\lambda})$ until convergence
   - Use (10.50) to calculate the mixing coefficients of free components
   - Split test fails if only one remaining component left. Move to next component
   - If both components are redundant, split test fails. Move on to next component
   - If both components remains, then $M = M + 1$
6. Repeat steps 4, 5 until the splitting test fails for all the components.

## 10.4 Experimental Results

We present the experimental results we have obtained with our model in this section. We compare our variational IBL mixture model (IBLMM) with Gaussian mixture models with maximum likelihood estimation (GMM) and variational Gaussian mixture models (varGMM) since these are the standard nowadays. We evaluate our model against four challenging applications: object categorization, speech categorization, spam image categorization, and software defect categorization. We try varying combinations involving imbalanced data to check the robustness of our model even when little data is available. We start with more or less equally weighted data sets to highly imbalanced data. The results are as follows:

### *10.4.1 Speech Categorization*

With voice recognition based automation and control taking over in recent time, efficient categorization of speech signals becomes an important task. To evaluate our model, we took a simple task of clustering between male and female speakers in the TSP speech data set [22]. The TSP data set consists of speech utterances of 10 speakers where five are male and five are female. There are 60 speech utterances for each of the speakers. We take 500 samples from each category for our experiment. The pre-processing step for speech data involves removal of non-speech parts like momentary pauses as a first step. This is done by voice activity detection (VAD) which removes the empty signals so that our model doesn't get trained on unnecessary pause signals. We now extract Mel Frequency Cepstral Coefficients (MFCC) which has been widely used for speech recognition tasks [29, 33]. The MFCC feature descriptors are 39 dimensional. Each speech utterance is sampled with a frame rate of 25 ms with a window shift of 10 ms. By this method a number of feature descriptors can be obtained from a single speech utterance file. We use bag of words feature model to create a histogram of the extracted MFCC features. This data serves as input to our model. The confusion matrix for our model is shown in Fig. 10.2. Table 10.1 shows the accuracy of IBLMM compared to GMM and varGMM. It shows that IBLMM improves the accuracy of GMM and varGMM

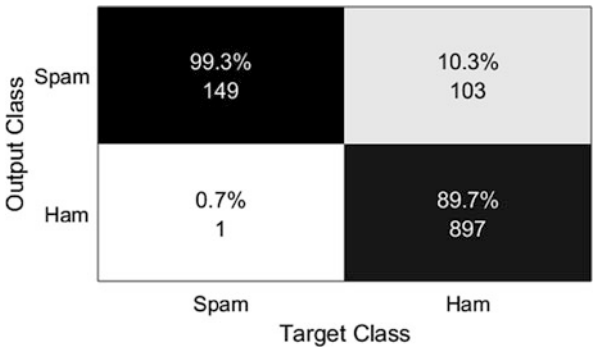



**Fig. 10.2** Confusion matrix of TSP speech data set with varIBLMM

**Table 10.1** Accuracy of different models for TSP speech data set

| Method | Accuracy(%) |
|---|---|
| varIBLMM | 86.6 |
| varGMM | 85.9 |
| GMM | 85.2 |

### *10.4.2 Image Categorization*

Pattern recognition from images is an important part of applications related to computer vision [8, 9, 23, 32]. It plays a major role in image retrieval, automated machinery, robot navigation, etc. For us to apply our model for image clustering, we have to extract feature descriptors from the images. Some of the commonly used methods for feature extraction are: Scale Invariant feature Transform (SIFT) [25], Histogram of Gaussians (HoG) [12], Speeded-Up Robust Features (SURF) [3], etc. Once the features are extracted we have to represent each image in terms of these features. The best way to do that would be to use the bag of visual words representation [11, 27, 34]. The idea for the bag of visual words approach is to cluster all the feature descriptors extracted from all the images using k-means creating a histogram of unique features for each of the image. These data will act as input to our model.

#### 10.4.2.1 Images Clustering

For our first experiment we use the Ghim dataset[1] to evaluate the efficiency of our model. The Ghim dataset has 20 categories with 500 images in each class. Each of the images is 400×300 or 300×400. All images are in JPEG format. We use only four classes for ease of representation. Sample image from each of the four classes is shown in Fig. 10.3 we choose 500 images from the fireworks class, 150 images from cars, 250 images from Chinese buildings, and 275 from dragon flies contributing 1175 images on the whole. It should be noted that the data points belonging to each class are varied over the four classes. We extract SIFT features from these images and use it to create bag of visual words features. Figure 10.4 shows the confusion matrix obtained by using IBLMM on this data. The comparison of accuracy with GMM and varGMM model is shown in Table 10.2. It is clearly seen that the accuracy with our model is higher than the other two.


**Fig. 10.3** Sample images from Ghim dataset. (**a**) Fireworks. (**b**) Cars. (**c**) Chinese buildings. (**d**) Dragon flies

[1] http://www.ci.gxnu.edu.cn/cbir/Dataset.aspx.

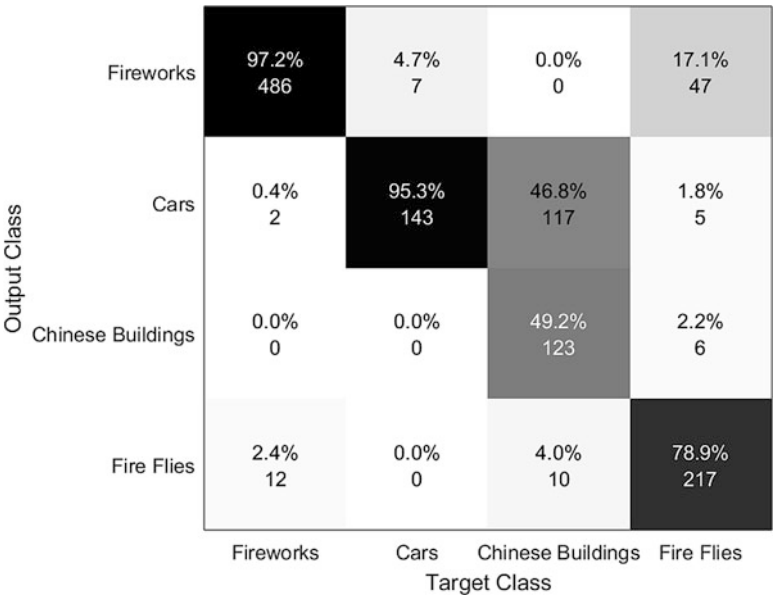



**Fig. 10.4** Confusion matrix of Ghim data set with varIBLMM

**Table 10.2** Accuracy of different models for Ghim dataset

| Method | Accuracy(%) |
|---|---|
| varIBLMM | 82.46 |
| varGMM | 74.21 |
| GMM | 74.04 |

#### 10.4.2.2 Spam Images Clustering

Usage of email services has become a quotidian task of everyday life nowadays. This also leaves us as a target to multiple ad agencies and fraudsters who send repeated ads and fake ones to trick us to reveal our personal information. Spam mails have also become a source of threats over the recent years. Hence it is very important to isolate the spam mails from the legitimate ones. However, it is also a very challenging task as the amount of spam data available is less when compared to the real ones in real- world applications due to the presence of repeated images. Due to this reason detecting Spam images could be a good application to test the robustness our model. So we choose the spam data set created in [13] which consisted of three sets of images. One is the ham data which contains normal images obtained from personal mails of people and considered useful. The other two sets contain spam images from spam archive created in [18] and a set of spam images taken from personal spam emails. In addition to this we also use images from the Princeton spam benchmark data set.[2] All the images used are taken from

[2] http://www.cs.princeton.edu/cass/spam/.

real emails and hence is a good representation of the real-world scenario. However, all the three spam data sets contained a number of duplicate images. We took 150 varied spam images between the three spam data sets and 1000 images from the ham data. Sample images form the spam and ham sets are shown in Figs. 10.5 and 10.6, respectively. We can see that the spam data accounts for only 15% of the total data. We then extract SIFT features from these images and then create visual bag of words feature histogram from it. The confusion matrix for this data with our model is shown in Fig. 10.7. In applications based on security it is important that the false negative rate (FNR) is low because even if one malicious image is allowed in the


**Fig. 10.5** Sample images from the spam collection

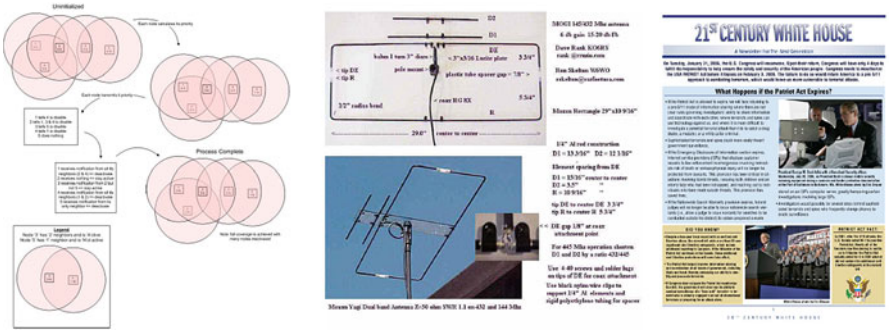



**Fig. 10.6** Sample images from the ham collection

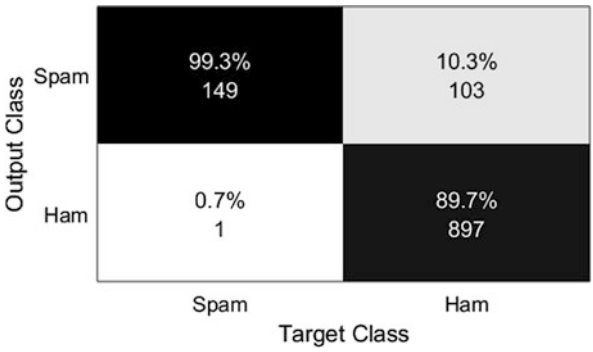



**Fig. 10.7** Confusion matrix of spam image data set with varIBLMM

network it might result in compromising the entire network in the worst case. On the other hand, it is also important that the important images that are intended for the user are delivered as well; hence the false positive rates (FPR) should be low. Due to this reason both FPR and FNR have to be low for a good spam categorization model. We enforce the following performance measures to evaluate our model:

$$Precision = \frac{TP}{TP + FP} \tag{10.53}$$

$$Recall = \frac{TP}{TP + FN} \tag{10.54}$$

$$FalsePositiveRate(FPR) = \frac{FP}{FP + TN} \tag{10.55}$$

$$FalseNegativeRate(FNR) = \frac{FN}{FN + TP} \tag{10.56}$$

where true positives (TP) is the number of spam images correctly predicted as spam; false positives (FP) is the number of non-spam images predicted as spam; true negatives (TN) is the number of non-spam images correctly predicted as not spam and false negative (FN) is the number of spam images that have been classified as not spam. Table 10.3 shows the comparison of different performance measures for IBLMM, GMM, and varGMM, respectively. The FNR and FPR values are both low compared to GMM and varGMM which highlights the capability of our model to cluster imbalanced data sets.

### *10.4.3 Software Defect Categorization*

Identification of software defects is an important part of software testing. Using machine learning techniques helps to identify defects in a short time and helps reduce the manual workforce for testing [17, 19, 21]. We validate our model against five data sets from the Promise software engineering repository [28], namely CM 1, JM1, KC1, KC2, and PC1. CM1, JM1, and PC1 are written in C and KC1 and KC2 are written in C++. CM1 is a software written for a NASA spacecraft instrument, JM1 is a real-time predictive ground system, KC1 and KC2 are storage

**Table 10.3** Performance measures of different models for spam image data set

| Method | Accuracy(%) | Precision | Recall | FPR | FNR |
|---|---|---|---|---|---|
| varIBLMM | 90.96 | 59.1 | 99.33 | 0.10 | 0.006 |
| varGMM | 81.22 | 40.8 | 98.00 | 0.21 | 0.020 |
| GMM | 79.73 | 39.1 | 98.66 | 0.23 | 0.013 |

**Table 10.4** Results on defect detection using different models

| Data Set | Model (%) | Accuracy | Precision | Recall | FNR |
|---|---|---|---|---|---|
| CM1 | varIBL | 67.87 | 17.50 | 61.22 | 0.39 |
| | varGMM | 72.69 | 1.13 | 2.04 | 0.98 |
| | GMM | 71.29 | 1.04 | 2.04 | 0.98 |
| JM1 | varIBL | 66.09 | 0.29 | 52.16 | 0.48 |
| | varGMM | 74.10 | 0.23 | 15.59 | 0.84 |
| | GMM | 74.10 | 0.23 | 15.59 | 0.84 |
| KC1 | varIBL | 69.41 | 30.28 | 75.15 | 0.25 |
| | varGMM | 73.06 | 32.81 | 70.85 | 0.29 |
| | GMM | 72.68 | 32.39 | 70.56 | 0.29 |
| KC2 | varIBL | 77.78 | 47.40 | 79.43 | 0.21 |
| | varGMM | 49.04 | 4.04 | 6.50 | 0.93 |
| | GMM | 74.90 | 7.14 | 1.87 | 0.98 |
| PC1 | varIBL | 68.80 | 11.68 | 53.24 | 0.47 |
| | varGMM | 89.35 | 2.32 | 1.3 | 0.99 |
| | GMM | 89.35 | 2.32 | 1.3 | 0.99 |

management systems for processing ground data, and PC1 is a flight software for earth orbiting satellites. McCabe and Halstead features are considered to describe the source code of these software to create these data sets. The performance metrics used are the same as in previous subsection. In the case of software defects we are more concerned about the false negatives and hence FNR is the most important measure. From Table 10.4 we are able to see that the False negative rate of IBLMM is very much higher than GMM and varGMM for all the data sets. The ratio between the defect and the non-defect class was around 1:10 in some cases and we found in our experiments that varGMM and GMM are unable to distinguish the data into two classes in these scenarios.

## 10.5 Conclusion

We have proposed an efficient mixture model for clustering based on Inverted Beta Liouville mixtures. The variational framework combined with component splitting approach is found to be effective in model selection. The robustness of our model is evident from the experiments which involved data sets with varied weights. The first experiment with equal weight exhibited good results. The second experiment for object image clustering showed the effectiveness of our model in terms of mixed weights and also proved the efficiency of model selection. With spam image clustering we were able to achieve 90% accuracy against GMM and varGMM which had only around 80% accuracy along with low FPR and FNR. In the last experiment, our model proved to be better than the other two models in identifying the defect class. It is to be noted that some data sets in this experiment had less than 10%

of total data for the defect class. As the results are encouraging we can introduce feature selection within the variational framework along with model selection to improve the efficiency of our model.

## Appendix: Proof of Eqs. (10.23), (10.24), (10.25) and (10.26)

From Eq. 10.16 we can write the logarithm of the joint as:

$$\begin{aligned}
\ln p(\mathcal{X}, \mathcal{Z}) = \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{ij}\Bigg[ & \ln \frac{\Gamma\left(\sum_{l=1}^{D}\alpha_{jl}\right)}{\prod_{l=1}^{D}\Gamma\left(\alpha_{jl}\right)} + \ln \frac{\Gamma(\alpha_j+\beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)} + \sum_{l=1}^{D}\left(\alpha_{jl}-1\right)\ln X_{il} \\
& + \beta_j \ln \lambda_j + \left(\alpha_j - \sum_{i=1}^{D}\alpha_{jl}\right)\ln\left(\sum_{i=1}^{D} X_{il}\right) \\
& - \left(\alpha_j+\beta_j\right)\ln\left(\lambda_j + \sum_{i=1}^{D} X_{il}\right)\Bigg] \\
& + \sum_{i=1}^{N}\left[\sum_{j=1}^{s} Z_{ij}\ln\pi_j + \sum_{j=s+1}^{M} Z_{ij}\ln\pi_j^*\right] - \left(M-s\right)\ln\left[1-\sum_{k=1}^{s}\pi_k\right] \\
& + \ln\frac{\Gamma\left(\sum_{j=s+1}^{M} c_j\right)}{\prod_{j=s+1}^{M}\Gamma\left(c_j\right)} + \sum_{j=s+1}^{M}\left(c_j-1\right)\ln\left[\pi_j^* - \left(1-\sum_{k=1}^{s}\pi_k\right)\right] \\
& + \sum_{j=1}^{M}\sum_{l=1}^{D} u_{jl}\ln\nu_{jl} - \ln\Gamma\left(u_{jl}\right) + \left(u_{jl}-1\right)\ln\alpha_{jl} - \nu_{jl}\alpha_{jl} \\
& + \sum_{j=1}^{M} p_j \ln q_j - \ln\Gamma\left(p_j\right) + \left(p_j-1\right)\ln\alpha_j - q_j\alpha_j \\
& + \sum_{j=1}^{M} g_j \ln h_j - \ln\Gamma\left(g_j\right) + \left(g_j-1\right)\ln\beta_j - h_j\beta_j \\
& + \sum_{j=1}^{M} s_j \ln t_j - \ln\Gamma\left(s_j\right) + \left(s_j-1\right)\ln\lambda_j - t_j\lambda_j
\end{aligned} \tag{10.57}$$

To derive the variational solutions of each parameter, we consider the logarithm with respect to each of the parameter assuming the rest of the parameters to be constant. This is explained in the following subsections.

### *Variational Solution for Q(Z) Eq. (10.23)*

The logarithm with respect to $Q(Z_i)$ on the joint is given by:

$$\begin{aligned}\ln Q(Z_i) = &\sum_{j=1}^{M} Z_{ij}\Bigg[R_j + S_j + \sum_{l=1}^{D}(\alpha_{jl}-1)\ln X_{il} + \beta_j \ln \lambda_j \\ &+ \Big(\alpha_j - \sum_{i=1}^{D}\alpha_{jl}\Big)\ln\Big(\sum_{i=1}^{D} X_{il}\Big) - (\alpha_j+\beta_j)T_{ij}\Bigg] \\ &+ \Bigg[\sum_{j=1}^{s} Z_{ij}\ln\pi_j + \sum_{j=s+1}^{M} Z_{ij}\ln\pi_j^*\Bigg] \qquad (10.58)\\ = &\sum_{j=1}^{s}\Bigg[\ln\pi_j + R_j + S_j + \sum_{l=1}^{D}(\alpha_{jl}-1)\ln X_{il} + \beta_j\ln\lambda_j \\ &+ \Big(\alpha_j - \sum_{i=1}^{D}\alpha_{jl}\Big)\ln\Big(\sum_{i=1}^{D} X_{il}\Big) - (\alpha_j+\beta_j)T_{ij}\Bigg] \\ &\sum_{j=s+1}^{M}\Bigg[\langle\ln\pi_j^*\rangle + R_j + S_j + \sum_{l=1}^{D}(\alpha_{jl}-1)\ln X_{il} + \beta_j\ln\lambda_j \\ &+ \Big(\alpha_j - \sum_{i=1}^{D}\alpha_{jl}\Big)\ln\Big(\sum_{i=1}^{D} X_{il}\Big) - (\alpha_j+\beta_j)T_{ij}\Bigg] \qquad (10.59)\end{aligned}$$

where

$$R_j = \left\langle \ln \frac{\Gamma\left(\sum_{l=1}^{D}\alpha_{jl}\right)}{\prod_{l=1}^{D}\Gamma(\alpha_{jl})}\right\rangle,\ S_j = \left\langle \ln\frac{\Gamma(\alpha_j+\beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)}\right\rangle,\ T_{ij} = \left\langle \ln\Big(\lambda_j + \sum_{i=1}^{D} X_{il}\Big)\right\rangle \qquad (10.60)$$

$R_j$, $S_j$, and $T_{ij}$ are intractable in the above equations. Due to this reason we use second order Taylor series approximation for $R_j$ and $S_j$ and first order Taylor series approximation for $T_{ij}$. the equations are given in Eqs. (10.30), (10.31), and (10.32), respectively. It is a notable fact that (10.58) is of the form:

$$\ln Q(\mathcal{Z}) = \sum_{i=1}^{N}\Bigg[\sum_{j=1}^{s} Z_{ij}\ln\tilde{r}_{ij} + \sum_{j=s+1}^{M} Z_{ij}\ln\tilde{r}_{ij}^*\Bigg] + \text{const} \qquad (10.61)$$

given

$$\ln \tilde{r}_{ij} = \ln \pi_j + R_j + S_j + \left(\overline{\alpha}_j - \sum_{l=1}^{D} \overline{\alpha}_{jl}\right) \ln\left(\sum_{l=1}^{D} X_{il}\right) + \overline{\beta}_j \langle \ln \lambda_j \rangle + \sum_{l=1}^{D} \left[\left(\overline{\alpha}_{jd} - 1\right) \ln X_{id}\right] - \left(\overline{\alpha} + \overline{\beta}\right) T_{ij} \tag{10.62}$$

$$\ln \tilde{r}_{ij}^{*} = \langle \ln \pi_j^{*} \rangle + R_j + S_j + \left(\overline{\alpha}_j - \sum_{l=1}^{D} \overline{\alpha}_{jl}\right) \ln\left(\sum_{l=1}^{D} X_{il}\right) + \overline{\beta}_j \langle \ln \lambda_j \rangle + \sum_{l=1}^{D} \left[\left(\overline{\alpha}_{jd} - 1\right) \ln X_{id}\right] - \left(\overline{\alpha} + \overline{\beta}\right) T_{ij} \tag{10.63}$$

By taking the exponentiation of Eq. (10.58) we can write:

$$Q(\mathcal{Z}) \propto \prod_{i=1}^{N} \left[ \prod_{j=1}^{s} \tilde{r}_{ij}^{Z_{ij}} \prod_{j=s+1}^{M} \tilde{r}_{ij}^{*Z_{ij}} \right] \tag{10.64}$$

Normalizing this equation we can write the variational solution of $Q(\mathcal{Z})$ as

$$Q(\mathcal{Z}) \propto \prod_{i=1}^{N} \left[ \prod_{j=1}^{s} r_{ij}^{Z_{ij}} \prod_{j=s+1}^{M} r_{ij}^{*Z_{ij}} \right] \tag{10.65}$$

where $r_{ij}$ and $r_{ij}^{*}$ can be obtained from Eqs. (10.28) and (10.29). Also, we can say that $\langle Z_{ij} \rangle = r_{ij}$ for $j = 1, \ldots, s$ and $\langle Z_{ij}^{*} \rangle = r_{ij}^{*}$ for $j = s+1, \ldots, M$

### *Proof of Eq. (10.24): Variational Solution of $Q(\boldsymbol{\pi}^{*})$*

Similarly, the logarithm of the variational solution $Q(\boldsymbol{\pi}^{*})$ is given as

$$\begin{aligned} \ln Q(\pi_j^{*}) &= \langle \ln p(\mathcal{X}, \Theta) \rangle_{\Theta \neq \pi_j^{*}} \\ &= \sum_{i=1}^{N} \langle Z_{ij} \rangle \ln \pi_j^{*} + (c_j - 1) \ln \pi_j^{*} + \text{const} \\ &= \ln \pi_j^{*} \left[ \sum_{i=1}^{N} \langle Z_{ij} \rangle + c_j - 1 \right] + \text{const} \end{aligned} \tag{10.66}$$

This equation shows that it has the same logarithmic form as that of Eq. (10.15). So we can write the variational solution of $Q(\boldsymbol{\pi}^*)$ as

$$Q(\boldsymbol{\pi}^*) = \left(1 - \sum_{k=1}^{s} \pi_k\right)^{-M+s} \frac{\Gamma\left(\sum_{j=s+1}^{M} c_j^*\right)}{\prod_{j=s+1}^{M} \Gamma(c_j^*)} \prod_{j=s+1}^{M} \left(\frac{\pi_j^*}{1 - \sum_{k=1}^{s} \pi_k}\right)^{c_j^*-1} \tag{10.67}$$

where

$$c_j^* = \sum_{i=1}^{N} \langle Z_{ij} \rangle + c_j \tag{10.68}$$

$\langle Z_{ij} \rangle = r_{ij}^*$ in the above equation.

## *Proof of Eq. (10.25): Variational Solution of $Q(\boldsymbol{\alpha})$*

As in the other two cases the logarithm of the variational solution $Q(\alpha_{jl})$ is given by

$$\begin{aligned} \ln Q(\alpha_{jl}) =& \langle \ln p(\mathcal{X}, \Theta) \rangle_{\Theta \neq \alpha_{jl}} \\ =& \sum_{i=1}^{N} \langle Z_{ij} \rangle \left[ \mathcal{J}(\alpha_{jl}) + \alpha_{jl} \ln X_{il} - \alpha_{jl} \ln \left( \sum_{i=1}^{D} X_{il} \right) \right] \\ &+ (u_{jl} - 1) \ln \alpha_{jl} - v_{jl} \alpha_{jl} + \text{const} \end{aligned} \tag{10.69}$$

where

$$\mathcal{J}(\alpha_{jl}) = \left\langle \ln \frac{\Gamma\left(\alpha_{jl} + \sum_{s \neq l}^{D+1} \alpha_{js}\right)}{\Gamma(\alpha_{jl}) \prod_{s \neq l}^{D+1} \Gamma(\alpha_{js})} \right\rangle_{\Theta \neq \alpha_{jl}} \tag{10.70}$$

Similar to what we encountered in the case of $R_j$ the equation for $\mathcal{J}(\alpha_{jl})$ is also intractable. We solve this problem finding the lower bound for the equation by calculating the first-order Taylor expansion with respect to $\overline{\alpha}_{jl}$. The calculated lower bound is given by

$$\mathcal{L}(\alpha_{jl}) \geq \overline{\alpha}_{jl} \ln \alpha_{jl} \left[ \psi\left( \sum_{l=1}^{D+1} \overline{\alpha}_{jl} \right) - \psi(\overline{\alpha}_{jl}) + \sum_{s \neq l}^{D+1} \overline{\alpha}_{js} \right.$$

$$\times \psi'\Big(\sum_{l=1}^{D+1}\overline{\alpha}_{jl}\Big)\big(\langle \ln \alpha_{js}\rangle - \ln \overline{\alpha}_{js}\big)\Big] + \text{const} \tag{10.71}$$

This approximation is also found to be a strict lower bound of $\mathcal{L}(\alpha_{jl})$. Substituting this equation for lower bound in Eq. (10.69)

$$\begin{aligned}\ln Q(\alpha_{jl}) = \sum_{i=1}^{N}\langle Z_{ij}\rangle\overline{\alpha}_{jl}\ln\alpha_{jl}\Big[\psi\Big(\sum_{l=1}^{D}\overline{\alpha}_{jl}\Big) - \psi(\overline{\alpha}_{jl}) \\ + \psi'\Big(\sum_{l=1}^{D}\overline{\alpha}_{jl}\Big)\sum_{d\neq l}^{D}\big(\langle\ln\alpha_{jl}\rangle - \ln\overline{\alpha}_{jl}\big)\overline{\alpha}_{jl}\Big] \\ + \sum_{i=1}^{N}\alpha_{jl}\langle Z_{ij}\rangle\Big[\ln X_{il} - \ln\Big(\sum_{l=1}^{D}X_{il}\Big)\Big] + \text{const}\end{aligned} \tag{10.72}$$

This equation can be rewritten as

$$\ln Q(\alpha_{jl}) = \ln\alpha_{jl}\big(u_{jl} + \varphi_{jl} - 1\big) - \alpha_{jl}\big(v_{jl} - \vartheta_{jl}\big) + \text{const} \tag{10.73}$$

where

$$\begin{aligned}\varphi_{jl} = \sum_{i=1}^{N}\langle Z_{ij}\rangle\overline{\alpha}_{jl}\Big[\psi\Big(\sum_{l=1}^{D}\overline{\alpha}_{jl}\Big) - \psi(\overline{\alpha}_{jl}) \\ + \psi'\Big(\sum_{l=1}^{D}\overline{\alpha}_{jl}\Big)\sum_{d\neq l}^{D}\big(\langle\ln\alpha_{jl}\rangle - \ln\overline{\alpha}_{jl}\big)\overline{\alpha}_{jl}\Big]\psi'\Big(\sum_{l=1}^{D+1}\overline{\alpha}_{jl}\Big)\big(\langle\ln\alpha_{js}\rangle - \ln\overline{\alpha}_{js}\big)\Big]\end{aligned} \tag{10.74}$$

$$\vartheta_{jl} = \sum_{i=1}^{N}\langle Z_{ij}\rangle\Big[\ln X_{il} - \ln\Big(\sum_{l=1}^{D}X_{il}\Big)\Big] \tag{10.75}$$

Eq. (10.73) is the logarithmic form of a Gamma distribution. If we exponentiate both the sides, we get

$$Q(\alpha_{jl}) \propto \alpha_{jl}^{u_{jl}+\varphi_{jl}-1}e^{-\left(v_{jl}-\vartheta_{jl}\right)\alpha_{jl}} \tag{10.76}$$

This leaves us with the optimal solution for the hyper-parameters $u_{jl}$ and $v_{jl}$ given by

$$u_{jl}^{*} = u_{jl} + \varphi_{jl}, \quad v_{jl}^{*} = v_{jl} - \vartheta_{jl} \tag{10.77}$$

By following the same procedure we can get the variational solutions for $Q(\boldsymbol{\alpha})$, $Q(\boldsymbol{\beta})$, and $Q(\boldsymbol{\lambda})$.

## References


1. Bakhtiari, A.S., Bouguila, N.: A latent Beta-Liouville allocation model. Expert Syst. Appl. **45**, 260–272 (2016)
2. Banfield, J.D., Raftery, A.E.: Model-based gaussian and non-gaussian clustering. Biometrics **49**(3), 803–821 (1993)
3. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (surf). Comput. Vis. Image Underst. **110**(3), 346–359 (2008). Similarity Matching in Computer Vision and Multimedia
4. Bdiri, T., Bouguila, N.: Positive vectors clustering using inverted Dirichlet finite mixture models. Expert Syst. Appl. **39**(2), 1869–1882 (2012)
5. Belloni, A., Chernozhukov, V.: On the computational complexity of MCMC-based estimators in large samples. Ann. Statist. **37**(4), 2011–2055 (2009)
6. Bouguila, N.: A variational component splitting approach for finite generalized Dirichlet mixture models. In: 2012 International Conference on Communications and Information Technology (ICCIT), pp. 53–57 (2012)
7. Bouguila, N., Ziou, D., Vaillancourt, J.: Novel mixtures based on the Dirichlet distribution: Application to data and image classification. In: Perner, P., Rosenfeld, A. (eds.) Machine Learning and Data Mining in Pattern Recognition, pp. 172–181. Springer, Heidelberg (2003)
8. Chen, Y., Wang, J.Z., Krovetz, R.: An unsupervised learning approach to content-based image retrieval. In: Proceeding Seventh International Symposium Signal Processing and its Applications, vol. 1, pp. 197–200 (2003). https://doi.org/10.1109/ISSPA.2003.1224674
9. Chen, Y., Wang, J.Z., Krovetz, R.: Clue: cluster-based retrieval of images by unsupervised learning. IEEE Trans. Image Process. **14**(8), 1187–1201 (2005). https://doi.org/10.1109/TIP.2005.849770
10. Constantinopoulos, C., Likas, A.: Unsupervised learning of gaussian mixtures based on variational component splitting. IEEE Trans. Neural Netw. **18**(3), 745–755 (2007)
11. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, pp. 1–22 (2004)
12. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceeding IEEE Computer Society Conference Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 886–893 (2005)
13. Dredze, M., Gevaryahu, R., Elias-Bachrach, A.: Learning fast classifiers for image spam. In: CEAS 2007 - The Fourth Conference on Email and Anti-Spam, 2–3 August 2007, Mountain View, California, USA (2007)
14. Fan, W., Bouguila, N.: Variational learning of finite Beta-Liouville mixture models using component splitting. In: The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2013)
15. Fan, W., Bouguila, N.: Model-based clustering based on variational learning of hierarchical infinite beta-liouville mixture models. Neural. Process. Lett. **44**(2), 431–449 (2016)
16. Fan, W., Bouguila, N., Ziou, D.: Variational learning of finite Dirichlet mixture models using component splitting. Neurocomputing **129**, 3–16 (2014)
17. Felix, E.A., Lee, S.P.: Integrated approach to software defect prediction. IEEE Access **5**, 21524–21547 (2017)

18. Fumera, G., Pillai, I., Roli, F.: Spam filtering based on the analysis of text information embedded into images. J. Mach. Learn. Res. **7**, 2699–2720 (2006)
19. Islam, R., Sakib, K.: A package based clustering for enhancing software defect prediction accuracy. In: 2014 17th International Conference on Computer and Information Technology (ICCIT). IEEE, Piscataway, pp. 81–86 (2014)
20. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Comput. Surv. **31**(3), 264–323 (1999)
21. Jing, X.Y., Zhang, Z.W., Ying, S., Wang, F., Zhu, Y.P.: Software defect prediction based on collaborative representation classification. In: Companion Proceedings of the 36th International Conference on Software Engineering, pp. 632–633. ACM, New York (2014). ICSE Companion 2014
22. Kabal, P.: TSP speech database. Tech. rep., Department of Electrical & Computer Engineering. McGill University, Montreal (2002)
23. Liu, D., Chen, T.: Unsupervised image categorization and object localization using topic models and correspondences between images. In: Proceeding IEEE 11th International Conference Computer Vision, pp. 1–7 (2007). https://doi.org/10.1109/ICCV.2007.4408852
24. Loh, W.Y.: Symmetric multivariate and related distributions. Technometrics **34**, 235–236 (2012)
25. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91 (2004)
26. Opper, M., Saad, D.: Tutorial on Variational Approximation Methods. MITP, Cambridge (2001)
27. Ravinder, M., Venugopal, T.: Content-based cricket video shot classification using bag-of-visual-features. In: Artificial Intelligence and Evolutionary Computations in Engineering Systems (2016)
28. Sayyad Shirabad, J., Menzies, T.: The PROMISE Repository of Software Engineering Databases. School of Information Technology and Engineering. University of Ottawa, Canada (2005)
29. Tyagi, V., Wellekens, C.: On desensitizing the mel-cepstrum to spurious spectral components for robust speech recognition. In: Proceedings (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005., vol. 1, pp. I/529–I/532 (2005)
30. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Trans. Neural Netw. **16**(3), 645–678 (2005)
31. Yu, J.: Fault detection using principal components-based gaussian mixture model for semiconductor manufacturing processes. IEEE Trans. Semicond. Manuf. **24**(3), 432–444 (2011)
32. Zakariya, S.M., Ali, R., Ahmad, N.: Combining visual features of an image at different precision value of unsupervised content based image retrieval. In: Proceeding IEEE International Conference Computational Intelligence and Computing Research, pp. 1–4 (2010)
33. Zheng, F., Zhang, G., Song, Z.: Comparison of different implementations of MFCC. J. Comput. Sci. Technol. **16**(6), 582–589 (2001)
34. Zhu, Q., Zhong, Y., Zhao, B., Xia, G., Zhang, L.: Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery. IEEE Geosci. Remote Sens. Lett. **13**(6), 747–751 (2016)
35. Zhu, J., Ge, Z., Song, Z.: Variational Bayesian gaussian mixture regression for soft sensing key variables in non-gaussian industrial processes. IEEE Trans. Control Syst. Technol. **25**(3), 1092–1099 (2017)

# Chapter 11
# Online Variational Learning for Medical Image Data Clustering


**Meeta Kalra, Michael Osadebey, Nizar Bouguila, Marius Pedersen, and Wentao Fan**

**Abstract** Data mining is an extensive area of research involving pattern discovery and feature extraction which is applied in various critical domains. In clinical aspect, data mining has emerged to assist the clinicians in early detection, diagnosis, and prevention of diseases. Advances in computational methods have led to implementation of machine learning in multi-modal clinical image analysis. One recent method is online learning where data become available in a sequential order, thus sequentially updating the best predictor for the future data at each step, as opposed to batch learning techniques which generate the best predictor by learning the entire data set at once.

In this chapter, we have examined and analysed multi-modal medical images by developing an unsupervised machine learning algorithm based on online variational inference for finite inverted Dirichlet mixture model. Our prime focus was to validate the developed approach on medical images. We do so by implementing the algorithm on both synthetic and real data sets. We test the algorithm's ability to detect challenging real world diseases, namely brain tumour, lung tuberculosis, and melanomic skin lesion.

M. Kalra (✉) · N. Bouguila
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: m_lra@encs.concordia.ca; nizar.bouguila@concordia.ca

M. Osadebey · M. Pedersen
Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway
e-mail: michael.osadebey@ntnu.no; marius.pedersen@ntnu.no

W. Fan
Department of Computer Science and Technology, Huaqiao University, Xiamen, China
e-mail: fwt@hqu.edu.cn



N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6_11

## 11.1 Introduction

In modern era, imaging is increasingly implemented in medical diagnosis and scientific research. Thus, leading to advances in technical and diagnostic improvements in the field of medical imaging [1]. This has resulted in the emergence of medical data mining which is an increasingly notable research domain [2]. In medicine, imaging is a non-invasive biomedical technique applied in clinical contexts to identify and diagnose diseases [3, 4]. Depending on the medical imaging technique used, medical imaging can give insights into two categories of biomedical analysis: structural or functional [5]. For example, MRI can be used to give structural information of the tumour mass but can also be used to monitor blood flow into the tumour, thus giving functional insights [6]. Although most medical images represent anatomical structures of the body, application of data mining on them can give valuable insights on the physiology and diagnosis for computer-aided diagnosis [7–9]. However, extraction and analysis of pertinent information from the often noisy medical images is becoming a more and more pressing issue [4, 10]. The growth of computational methods and image processing has found its way in clinical image processing and decision making [11]. These computational methods lead to different interpretations, medical usefulness and highlight various characteristics of the diagnosis by implementation of statistical models.

In this chapter, we describe how statistical approach has helped resolve the problem of noise and increase the amount of information that can be extracted from an image so as to support the clinician in making critical decisions faster and with confidence [12]. The differences between Monte Carlo Markov chain (MCMC), variational learning, and maximum likelihood estimation (MLE) methods have been succinctly described in [13]. For instance, variational learning and MLE methods have been described to be more efficient than MCMC. Out of these methods, maximum likelihood estimation (MLE) has been the most well described and well known in probabilistic models. It has been extensively applied for estimation of parameters in modern statistics. In this method, the Expectation Maximization (EM) algorithm has been generally adopted to be the standard methodology to learn finite mixture models. One problem which EM faces is the over-fitting and being unable to determine the model complexity [14]. However, the disadvantage can be offset by the adoption of Bayesian framework. The Bayesian approach is very comprehensive since the posterior distribution covers the uncertainty of the process. In essence, the Bayesian framework goes hand-in-hand with an approximation scheme. Robert and Cabella [15] describe the utilization of MCMC techniques as the most significant sampling methods which enabled the application of Bayesian techniques in wide aspects of studies. However, the critical challenge of MCMC is limitation to small-scale applications due to the need of high computational resources to solve it. In addition, convergence diagnosis approaches are not yet well developed for MCMC methods. Thus, variational inference method was developed to overcome the limitations of MCMC.

Variational inference, also known as variational Bayes, is a deterministic approximation method, where the model's posterior distribution is approximated using analytical procedures [16]. It has generated a lot of interest in finite mixture models through the provision of high generalization schemes and high computation tractability. Model selection and parameter estimation can be performed simultaneously through the use of variational inference.

Online mixture learning algorithms have been described to be more efficient in the modeling of data streams, as compared to batch algorithms. Examples include online Gaussian mixture models (GMM) considered for instance in [17]. The main short-coming which has been witnessed in this method is the unrealistic Gaussian assumption which is not catered for in real life. Most of the recent past research works have demonstrated that simulations with other methodological approaches can be better than the GMM when dealing with non-Gaussian data. A notable example is the Dirichlet mixture which is a better alternative when dealing with proportional data in several applications. The recent recommendation by a number of researchers is that other distributions should be considered and adopted for better results. For example, Bouguila and Ziou [18] developed an online learning approach in which the MML criterion was utilized and incorporated. An online variational inference algorithm has been developed in [19], also. In this chapter, we propose a more elaborated way in which model selection and online learning are examined simultaneously.

The rest of this chapter is organized as follows: Sect. 11.2 describes significant approaches of using data mining in the healthcare domain. Section 11.3 describes the motivation behind medical image segmentation. Sections 11.4 and 11.5 would describe the way to estimate the model complexity of the finite inverted Dirichlet mixture model and the parameters involved simultaneously in order to achieve an online variational inference algorithm. Section 11.6 of the chapter describes the accuracy and efficiency of the proposed approach on synthetic data sets and challenging real world medical applications which is followed by conclusion in Sect. 11.7.

## 11.2 Data Mining and Its Use in Healthcare

In simple terms, data mining approach uses computational models to extract useful information from the data. Particularly, in healthcare, the data generated are rich and multi-directional, e.g., electronic medical records data, medical image data, proteomic and genomic data, to name a few [20]. Despite the abundance of data, computer-aided decision support is at its nascency. In this aspect, data mining is implemented to assist clinicians in the early detection, diagnosis, and prevention of diseases. This is achieved by establishing models on medical data sets. These models learn from the data and help predict disease prognosis and progression.

Basically, data mining models are grouped into two categories; descriptive and predictive models [21]. As the name suggests, descriptive models define the associations that are represented in the data by pattern discovery [22]. In contrast, predictive models are applied to predict a future behaviour or trend as opposed to giving information of the existing behaviour [23]. Depending on the type of medical data, a descriptive or predictive model is chosen for. The important data mining tasks applied in healthcare to associate to patients raw data and extract it to validate conclusions on the diagnosis and treatment regimens [24, 25] are:

### *11.2.1 Classification*

Classification techniques are largely based on statistical models. As per the name, classification refers to the concept of assigning data into target classes. Data are grouped into testing and training sets whenever classification is being implemented. Training data are used by the classifiers in coming up with conclusive attributes of the data before they are put in classes whereas the testing data sets are used to determine the correctness or accuracy of the classifier.
In hospitals or clinics, classification can be applied to determine risk pattern of each patient depending on the data that are stored about the patient [26]. Since these classifiers are rule based, they are implemented to classify the patient into low or high risk populations for a certain diagnosis or disease [27]. In this approach, the patient cases are known, thus classification can be described as supervised learning. A practical application of classification is that the hospitals and diagnosing units determine the cost of treating the patients in the classes of low risk or high risk diseases [28].

### *11.2.2 Trend Analysis*

Trend analysis is a purely statistical approach where data are temporally examined. These data sets can be obtained through continuous recording of data of a specific patient. The statistical approach to this is called time series data analysis [19]. In this approach, data sets are assigned a “time” attribute such that time dependent properties of the data sets can be deduced and analysed. This analysis is important as time patterns and irregularities are critical concerns for the emergence of various diseases. For example, patients often experience immense pain during and after operation and require anesthesia. In normal recovery, the requirement of pain analgesic changes over time. Thus, the analysis of dose delivery information of the analgesic over time on a patient can help predict the variance of a patient pain relief condition [29]. Another application of trend analysis is to follow the population trends of patient populations undergoing a certain treatment for hospital

visits, medical costs, and lengths of stay of patients [30, 31]. Thus, incurring a trend in the aspects of treatment cost and effectiveness.

### *11.2.3 Clustering*

Simply put, clustering of data is the placing of similar data together in a cluster and dissimilar ones in the others. While clustering can be confused with classification, there is a notable difference among the two. Clustering is an unsupervised learning technique whereas classification is a supervised learning one. Importantly, in clustering the data information about classes is not known. Clustering also does not necessitate the subtle information for the partitioning of the data [32]. A major challenge in this method is that clusters have to be identified first. Typical examples of its application are genomic sequence analysis and genetic expression data analysis [33].

### *11.2.4 Regression*

In regression, data items are analyzed with the motivation of establishing a relationship in the known dependent variable and unknown and independent estimated variable. Statistically, regression is the most effective tool for predicting future patterns [34]. In biomedical research regression correlation coefficients are frequently used to establish a cause and effect relationship. For example, to determine if the patient has high blood pressure and the relationship of the risk of high blood pressure to the weight and age of the patient [35].

### *11.2.5 Association*

Association is the criterion in which the data are examined for the similarities or bonding in which they can be attributed. In examining the data, association rule is very effective. It reveals the correlations and relationships in which the objects are portrayed. Association rules are critical factors in medical marketing, advertising, and commodities management [13]. In essence, association rules make it possible for grouping items as per their attributes, then generating rules which can be used conclusively for the data sets. An accurate example is the ranking of hospitals where data mining techniques facilitate the placing of different hospitals according to their performance and other attributes by creating the necessary association on information from various hospitals and then ranking them [14, 36].

### *11.2.6 Summarization*

Using summarization, data can be examined and abstracted to smaller groups or sets of data. The smaller group of data gives the overall description or attributes of the generalized data. The data which are being abstracted can be examined in different ways or perspectives depending on the scope. For instance, this is effectively applied on electronic medical records where the data of the patient population are analysed, categorized based on the data and the insurance providers [16, 37]. By mining the data this way, patterns and regularities of a data set are easily recognized.

## 11.3 Motivation for Medical Image Segmentation

Image segmentation has attracted immense amount of interest in the medical domain because it can computationally discover the morphology inconsistencies embedded in an image of given organ or a tissue or a cell without relying on a set of predetermined labels. Conventional unsupervised image segmentation methods such as k-means, k-medoid, and c-means are typically known for the problem of cluster centre initialization and determination of optimum number of clusters. Similar challenges are experienced in probabilistic-based methods such as Gaussian mixture model (GMM).

These methods are not very suitable for real world classification problems due to the existence of the many correlated sub-tasks. Recently, online mixture Gaussian models and their extensions have been developed and also applied in many aspects. However, the Gaussian's consideration is rarely met and seems to be unrealistic in many real life applications.

The number of clusters that describe the data without the effect of over-fitting or under-fitting is one of the most challenging problems faced in finite mixture modeling. This problem is traditionally solved by using the maximum likelihood method with model selection criteria i.e., MDL, BIC, etc. However, using this approach requires evaluation of the given selection criterion for several number of components that is computationally demanding.

The two main challenging problems faced when dealing with finite mixtures are the determination of the number of the mixture components and the estimation of the mixture's parameters. With regard to parameters estimation, two families of approaches could be considered, namely frequentist and Bayesian techniques. Maximum likelihood (ML) which is the most popular among frequentist estimation techniques for mixture learning has several shortcomings since it can easily get caught in saddle points or local maxima and it depends on the initially set parameters.

In this chapter, we propose an online variational inference framework for finite inverted Dirichlet mixture model and demonstrate its application to medical image segmentation to assist medical practitioners in healthcare sector.

## 11.4 Model Specification

### *11.4.1 Finite Inverted Dirichlet Mixture Model*

The main reason for using finite inverted Dirichlet method is basically to have a flexible distribution for our mixture model. Unlike the Gaussian distribution, it is reasonably flexible and has the property to perform in both symmetric and asymmetric modes. A graphical model for finite inverted Dirichlet mixture model is shown in Fig. 11.1. Consider a positive $D$-dimensional vector that is sampled from a finite inverted Dirichlet mixture model with $M$ components. Hence, the finite mixture of inverted Dirichlet distributions can be defined as:

$$p(\mathbf{X}_i \mid \boldsymbol{\pi}, \boldsymbol{\alpha}) = \sum_{j=1}^{M} \pi_j ID(\mathbf{X}_i \mid \boldsymbol{\alpha}_j) \tag{11.1}$$

where $\alpha = (\alpha_1, \ldots, \alpha_M)$ and $\pi = (\pi_1, \ldots, \pi_M)$ denote the mixing coefficients along with the constraints that they are positive and sum to one. Also, the term $ID(\mathbf{X_i}|\boldsymbol{\alpha_j})$ hereby represents the $j$th inverted Dirichlet distribution with the parameter $(\boldsymbol{\alpha_j})$ which is defined as [32]:

$$ID(\mathbf{X}_i \mid \boldsymbol{\alpha}_j) = \frac{\Gamma\left(\sum_{l=1}^{D+1} \alpha_{jl}\right)}{\prod_{l=1}^{D+1} \Gamma(\alpha_{jl})} \prod_{l=1}^{D} X_{il}^{\alpha_{jl}-1} \left(1 + \sum_{l=1}^{D} X_{il}\right)^{-\sum_{l=1}^{D+1} \alpha_{jl}} \tag{11.2}$$

where, $X_{il}$ is positive for $l = 1, \ldots, D$ and $\boldsymbol{\alpha}_j = (\alpha_{j1}, \alpha_{j2}, \ldots..\alpha_{jD+1}), \alpha_{jl} > 0$ for $l = 1, \ldots, D+1$. Mean, variance, and co-variance of the inverted Dirichlet distribution are hereby given as under:

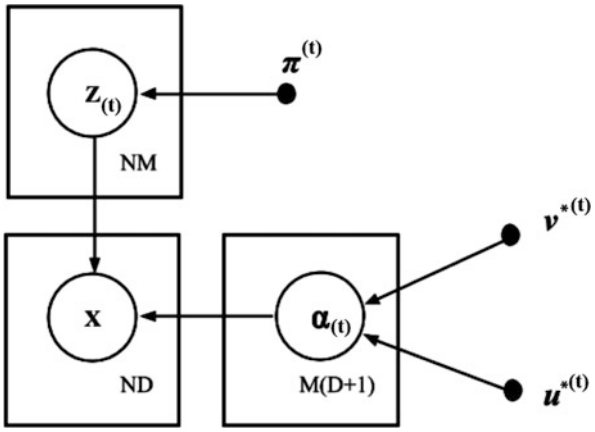



**Fig. 11.1** Graphical model representation for finite inverted Dirichlet mixture. Symbols in the circle denote the random variables; otherwise, they denote the model parameters

$$\mathbb{E}\left[X_l\right] = \frac{\alpha_l}{\alpha_{D+1} - 1} \tag{11.3}$$

$$var\left(X_l\right) = \frac{\alpha_l\left(\alpha_j + \alpha_{D+1} - 1\right)}{\left(\alpha_{D+1} - 1\right)^2\left(\alpha_{D+1} - 2\right)} \tag{11.4}$$

$$cov(X_a, X_b) = \frac{\alpha_a \alpha_b}{(\alpha_{D+1} - 1)^2(\alpha_{D+1} - 2))} \tag{11.5}$$

We introduce an M-dimensional binary random vector $\mathbf{Z_i} = \{Z_{i1}, \ldots., Z_{iM}\}$ called the latent variable which is hidden for each of the observed vector $X_i$ in order to calculate the maximum likelihood where $Z_{ij}$ is 1. Furthermore, conditional distribution of the Z given the mixing coefficients is as under:

$$p\left(\mathcal{Z} \mid \boldsymbol{\pi}\right) = \prod_{i=1}^{N}\prod_{j=1}^{M} \pi_j^{Z_{ij}} \tag{11.6}$$

Therefore, the conditional distribution of the data set $\mathcal{X}$ can be written as:

$$p\left(\mathcal{X} \mid \mathcal{Z}, \boldsymbol{\alpha}\right) = \prod_{i=1}^{N}\prod_{j=1}^{M} I\,D\left(\mathbf{X}_i \mid \boldsymbol{\alpha}_j\right)^{Z_{ij}} \tag{11.7}$$

Assuming that the parameters of the inverted Dirichlet are statistically independent, and for every parameter $\alpha_{jl}$, the gamma distribution that is adopted to approximate the conjugate prior is given as below:

$$p\left(\alpha_{jl}\right) = \mathcal{G}\left(\alpha_{jl} \mid u_{jl}, v_{jl}\right) = \frac{v_{jl}^{u_{jl}}}{\Gamma\left(u_{jl}\right)}\alpha_{jl}^{u_{jl}-1}e^{-v_{jl}\alpha_{jl}} \tag{11.8}$$

Here, $\{u_{jl}\}$ and $\{v_{jl}\}$ are hyper-parameters which have constraint such that $u_{jl} > 0$ and $v_{jl} > 0$. Now considering $\boldsymbol{\alpha}$ we can write,

$$p\left(\boldsymbol{\alpha}\right) = \prod_{j=1}^{M}\prod_{l=1}^{D} p\left(\alpha_{jl}\right) \tag{11.9}$$

The joint distribution of all the random variables can be written as:

$$\begin{aligned} p\left(\mathcal{X}, \mathcal{Z}, \boldsymbol{\alpha} \mid \boldsymbol{\pi}\right) &= p\left(\mathcal{X} \mid \mathcal{Z}, \boldsymbol{\alpha}\right)p\left(Z \mid \boldsymbol{\pi}\right)p\left(\boldsymbol{\alpha}\right) \\ &= \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\pi_j \frac{\Gamma\left(\sum_{l=1}^{D+1}\alpha_{jl}\right)}{\prod_{l=1}^{D+1}\Gamma\left(\alpha_{jl}\right)}\prod_{l=1}^{D} X_{il}^{\alpha_{jl}-1}\right. \end{aligned}$$

$$\times \left(1 + \sum_{l=1}^{D+1} X_{il}\right)^{-\sum_{l=1}^{D+1} \alpha_{jl}} \Bigg]^{Z_{ij}} \tag{11.10}$$

$$\times \prod_{j=1}^{M} \prod_{l=1}^{D+1} \frac{v_{jl}^{u_{jl}}}{\Gamma\left(u_{jl}\right)} \alpha_{jl}^{u_{jl}-1} e^{-v_{jl}\alpha_{jl}} \tag{11.11}$$

## 11.5 Online Variational Learning for Finite Inverted Dirichlet Mixture Model

### *11.5.1 Variational Inference*

Variational inference is used to formulate the computation of conditional probability in terms of an optimization problem which is basically deterministic approximation. The main objective of variational inference is to perform an approximation of conditional density of latent variables based on the observed variables. The best choice to find this approximation is by doing optimization. In essence, we make use of a family of densities over the latent variables, which are parameterized by free "variational parameters". Therefore, the task of the optimization is to find the member from this density family, i.e., the setting of the parameters that lie close to the conditional of interest using KL divergence [38]. In order to estimate the parameters of the finite inverted Dirichlet mixture model correctly and to select the appropriate number of components for the model, we adopted an online variational approach [39]. For simplifying the notation, we define $\Theta = \{\mathcal{Z}, \boldsymbol{\alpha}\}$. The main purpose of variational learning is to find an approximation $Q(\Theta)$, that approximates $p(\Theta|\mathcal{X}, \boldsymbol{\pi})$. To do this, we find the Kullback–Leibler (KL) divergence which is the distance between the distribution $Q(\Theta)$ and posterior distribution $p(\Theta \mid \mathcal{X}, \boldsymbol{\pi})$ given by,

$$KL(Q \mid\mid P) = -\int Q(\Theta) \ln\left(\frac{p(\Theta \mid \mathcal{X}, \boldsymbol{\pi})}{Q(\Theta)}\right) d\Theta \tag{11.12}$$

Modifying this equation we can write

$$KL(Q \mid\mid P) = \ln\ p(\mathcal{X} \mid \boldsymbol{\pi}) - \mathcal{L}(Q) \tag{11.13}$$

where, $L(Q)$ is called the variational lower bound, defined as:

$$\mathcal{L}(Q) = \int Q(\Theta) \ln\left(\frac{p(\mathcal{X}, \Theta \mid \boldsymbol{\pi})}{Q(\Theta)}\right) d\Theta \tag{11.14}$$

The KL divergence being a similarity measure follows the conditions $KL(Q \mid\mid P) \geq 0$ and $KL(Q \mid\mid P) = 0$ when $Q(\Theta) = p(\Theta \mid \mathcal{X})$. From (11.13) we can say $\mathcal{L}(Q)$ is the lower bound of $p(\mathcal{X} \mid \boldsymbol{\pi})$. We maximize the lower bound which means we are minimizing the KL divergence and hence approximating the true posterior distribution. However, the true posterior distribution cannot be used directly for variational inference as it is computationally intractable. Therefore, for this reason we use the method of mean-field approximation for our algorithm [40–42] by which we factorize $Q(\Theta)$ into disjoint tractable distributions as below:

$$Q(\Theta) = Q(\mathcal{Z})Q(\boldsymbol{\alpha}) \tag{11.15}$$

To maximize the lower bound $L(Q)$, we are supposed to make a variational optimization of $L(Q)$ with respect to each factor. The variational solution for a specific parameter $Q_k(\Theta_k)$ is:

$$Q_k(\Theta_k) = \frac{exp\langle \ln\ p(\mathcal{X}, \Theta)\rangle_{\neq k}}{\int exp\langle \ln\ p(\mathcal{X}, \Theta)\rangle_{\neq k} d\Theta} \tag{11.16}$$

where $\langle . \rangle_{\neq k}$ is the expectation with respect to all the parameters other than $\Theta_k$.

We hereby can obtain the following optimal variational solutions for the finite inverted Dirichlet mixture model (derived in Appendix section Proof of Eq. (11.17): Variational Solution of $Q(\mathcal{Z})$ and in Appendix section Proof of Eqs. (11.18), (11.22) and (11.23))

$$Q(\mathcal{Z}) = \prod_{i=1}^{N}\prod_{j=1}^{M} r_{ij}^{Z_{ij}} \tag{11.17}$$

$$Q(\boldsymbol{\alpha}) = \prod_{j=1}^{M}\prod_{l=1}^{D+1} G(\alpha_{jl}|u_{jl}^{*}, v_{jl}^{*}) \tag{11.18}$$

where,

$$r_{ij} = \frac{\rho_{ij}}{\sum_{j=1}^{M}\rho_{ij}} \tag{11.19}$$

$$\rho_{ij} = exp\left\{ \ln \pi_j + \tilde{R}_j + \sum_{l=1}^{D}\left(\overline{\alpha}_{jl} - 1\right)\ln X_{il} - \sum_{l=1}^{D+1}\overline{\alpha}_{jl}\ \ln\left(1 + \sum_{l=1}^{D} X_{il}\right)\right\} \tag{11.20}$$

$$\tilde{R}_j = \ln \frac{\Gamma\left(\sum_{l=1}^{D+1}\overline{\alpha}_{jl}\right)}{\prod_{l=1}^{D+1}\Gamma\left(\overline{\alpha}_{jl}\right)}$$

$$+\sum_{l=1}^{D+1}\overline{\alpha}_{jl}\left[\psi\left(\sum_{l=1}^{D+1}\overline{\alpha}_{jl}\right)-\psi\left(\overline{\alpha}_{jl}\right)\right]\left[\left\langle\ln\alpha_{jl}\right\rangle-\ln\overline{\alpha}_{jl}\right]$$

$$+\frac{1}{2}\sum_{l=1}^{D+1}\overline{\alpha}_{jl}^{2}\left[\psi'\left(\sum_{l=1}^{D+1}\overline{\alpha}_{jl}\right)-\psi'\left(\overline{\alpha}_{jl}\right)\right]-\left\langle\left(\ln\alpha_{jl}-\ln\overline{\alpha}_{jl}\right)^{2}\right\rangle$$

$$+\frac{1}{2}\sum_{a=1}^{D+1}\sum_{b=1}^{D+1}\overline{\alpha}_{ja}\,\overline{\alpha}_{jb}\left[\psi'\left(\sum_{l=1}^{D+1}\overline{\alpha}_{jl}\right)\left(\left\langle\ln\overline{\alpha}_{ja}\right\rangle-\ln\overline{\alpha}_{ja}\right)\right.$$

$$\left.\times\left(\left\langle\ln\overline{\alpha}_{jb}\right\rangle-\ln\overline{\alpha}_{jb}\right)\right] \tag{11.21}$$

The estimation equations for $u_{jl}^{*}$ and $v_{jl}^{*}$ are given by (derived in Appendix section Proof of Eqs. (11.18), (11.22) and (11.23))

$$u_{jl}^{*}=u_{jl}+\sum_{i=1}^{N}\left\langle Z_{ij}\right\rangle\overline{\alpha}_{jl}\left[\psi\left(\sum_{s=1}^{D+1}\overline{\alpha}_{js}\right)-\psi\left(\overline{\alpha}_{jl}\right)\right.$$

$$\left.+\sum_{s\neq l}^{D+1}\psi'\left(\sum_{s=1}^{D+1}\overline{\alpha}_{js}\right)\times\overline{\alpha}_{js}\left(\left\langle\ln\alpha_{js}\right\rangle-\ln\overline{\alpha}_{js}\right)\right] \tag{11.22}$$

$$v_{jl}^{*}=v_{jl}-\sum_{i=1}^{N}\left\langle Z_{ij}\right\rangle\left[\ln X_{il}-\ln\left(1+\sum_{l=1}^{D+1}X_{il}\right)\right] \tag{11.23}$$

$\psi(\cdot)$ and $\psi'(\cdot)$ in the above equations represent the digamma and trigamma functions. The expectation of values mentioned in the equations above is given by the equations below,

$$\left\langle Z_{ij}\right\rangle=r_{ij} \tag{11.24}$$

$$\overline{\alpha}_{jl}=\left\langle\alpha_{jl}\right\rangle=\frac{u_{jl}^{*}}{v_{jl}^{*}},\quad\left\langle\ln\alpha_{jl}\right\rangle=\psi\left(u_{jl}^{*}\right)-\ln v_{jl}^{*} \tag{11.25}$$

$$\left\langle\left(\ln\alpha_{jl}-\ln\overline{\alpha}_{jl}\right)^{2}\right\rangle=\left[\psi\left(u_{jl}^{*}\right)-\ln u_{jl}^{*}\right]^{2}+\psi'\left(u_{jl}^{*}\right) \tag{11.26}$$

We therefore maximize the variational lower bound $L(Q)$ to estimate the coefficient $\boldsymbol{\pi}$ which is treated as parameter for mixture model. The derivative of this lower bound with respect to $\boldsymbol{\pi}$ (derived in Appendix section Proof of Eq. (11.27)) is given as under:

$$\pi_j = \frac{1}{N}\sum_{i=1}^{N} r_{ij} \tag{11.27}$$

Therefore, for the variational learning of inverted Dirichlet mixture model, the value of the lower bound is calculated as:

$$\begin{aligned} L(Q) &= \sum_{z}\int Q(Z,\boldsymbol{\alpha})\ln\{\frac{p(\chi, Z, \boldsymbol{\alpha}|\boldsymbol{\pi})}{Q(Z,\boldsymbol{\alpha})}\}d\alpha \\ &= < \ln p(\chi|Z,\boldsymbol{\alpha}) > + < \ln p(Z|\boldsymbol{\pi}) > + < \ln p(\boldsymbol{\alpha}) > \\ &\quad - < \ln Q(Z) > - < \ln Q(\boldsymbol{\alpha}) > \end{aligned} \tag{11.28}$$

### *11.5.2 Online Variational Inference*

In this section, we present an online variational inference algorithm for finite inverted Dirichlet mixture models. In this algorithm we treat variational inference as a natural gradient which is the inverse of the Riemannian metric multiplied by the gradient [43]. We do this as it helps to achieve optimal convergence which allows to have faster online inference.

Online learning is when the data become available in a sequence and later the previous data are used as a reference to update the best predictor for the new incoming data at each step since the data are continuously arriving in online fashion. It is different from batch learning variational technique, in which we know the best predictor by working on the entire data set at the same time. Online learning is being commonly used in many areas where it is completely infeasible to train the entire data set at once since the data set is too large to be trained altogether. Online learning is also extensively useful in areas such as stock price prediction where it is important to adapt to the new patterns in the data or even when the data itself are generated as a function of time. In such a case when the data are continuously arriving in an online fashion, we have to estimate the variational lower bound to a fixed amount of data which is $N$. Considering this, the value expected from the model evidence $p(X)$ for a data with finite size can be derived as [44]:

$$\left\langle \ln p(X)\right\rangle_{\phi} = \int \phi(X)\ln\left(\int p(X|\Theta)p(\Theta)d(\Theta)\right)dx \tag{11.29}$$

where $\phi(X)$ represents the probability distribution which is unknown for the data observed. Thus, the corresponding expected variational lower bound can be computed using [44]:

$$\begin{aligned}\langle \mathcal{L}(Q)\rangle_{\phi} &= \left\langle \sum_{\mathcal{Z}} \int Q(\alpha) Q(\mathcal{Z}) \ln\left[\frac{p(X,\mathcal{Z}|\alpha)p(\alpha)}{Q(\alpha)Q(\mathcal{Z})}\right] d\alpha \right\rangle_{\phi} \\ &= N \int Q(\alpha) d\alpha \left\langle \sum_{Z} Q(Z) \ln\left[\frac{p(X,Z|\alpha)}{Q(Z)}\right]\right\rangle_{\phi} \\ &\quad + \int Q(\alpha) \ln\left[\frac{p(\alpha)}{Q(\alpha)}\right] d\alpha \end{aligned} \tag{11.30}$$

We consider $t$ as the actual amount of data observed, thus for the observed data, the current lower bound can be estimated by [44]

$$\begin{aligned}\mathcal{L}^{(t)}(Q) =& \frac{N}{t} \sum_{i=1}^{t} \int Q(\boldsymbol{\alpha}) \mathrm{d}\boldsymbol{\alpha} \sum_{\mathbf{Z}_i} Q(\mathbf{Z}_i) \ln\left[\frac{p(\mathbf{X}_i, \mathbf{Z}_i|\boldsymbol{\alpha})}{Q(\mathbf{Z}_i)}\right] \\ &+ \int Q(\boldsymbol{\alpha}) \ln\left[\frac{p(\boldsymbol{\alpha})}{Q(\boldsymbol{\alpha})}\right] \mathrm{d}\boldsymbol{\alpha} \end{aligned} \tag{11.31}$$

We realize that while $N$ remains fixed, $t$ increases over time. The main reason for this is the fact that the principal objective of the proposed online algorithm is the expected log evidence computed for a fixed amount of data. Even if there is an increase in the observed data, the algorithm basically computes the same quantity. Now relating this to the context, the former observed data are then used to improve the quality of estimation of the expected variational lower bound in Eq. (11.30). This inherently approximates the resulting log evidence as it does not have any previous knowledge of the former observed data.

With respect to the expectation values we saw in previous section, Eqs. (11.25) and (11.26) for $i = 1, 2, \ldots, N$ and $l = 1, 2, \ldots, D+1$ get modified to the below equations as the data are getting updated in online fashion.

$$\overline{\alpha}_{jl} = \langle \alpha_{jl} \rangle = \frac{u_{jl}^{(t-1)}}{v_{jl}^{(t-1)}}, \quad \langle \ln \alpha_{jl} \rangle = \psi\left(u_{jl}^{(t-1)}\right) - \ln v_{jl}^{(t-1)} \tag{11.32}$$

$$\left\langle \left(\ln \alpha_{jl} - \ln \overline{\alpha}_{jl}\right)^2 \right\rangle = \left[\psi\left(u_{jl}^{(t-1)}\right) - \ln u_{jl}^{(t-1)}\right]^2 + \psi'\left(u_{jl}^{(t-1)}\right) \tag{11.33}$$

The fundamental concept of this online algorithm is to enable successful maximization of the present variational lower bound in Eq. (11.31). Assuming that the observed data set exists in the form $\{X_1, \ldots X_{t-1}\}$. For every new observation $X_t$, we mainly perform maximization of the present $\mathcal{L}^{(t)}(Q)$ with respect to $Q(\mathcal{Z}_t)$ while $Q(\alpha)$ is set to $Q^{(t-1)}(\alpha)$ and $\pi_j$ is set to $\pi_j^{(t-1)}$. Hence, the variational solution can be computed using:

$$Q(\mathcal{Z}_t) = \prod_{j=1}^{M} r_{tj}^{Z_{tj}} \tag{11.34}$$

where we substitute Eq. (11.19) and $\rho_{ij}$ becomes

$$\rho_{ij} = exp\left\{ \ln \pi_j^{(t-1)} + \tilde{R}_j + \sum_{l=1}^{D} \left(\overline{\alpha}_{jl} - 1\right) \ln X_{il} - \sum_{l=1}^{D+1} \overline{\alpha}_{jl} \ln\left(1 + \sum_{l=1}^{D} X_{il}\right) \right\} \tag{11.35}$$

where $R_j$ is given by

$$\begin{aligned} \tilde{R}_j = & \ln \frac{\Gamma\left(\sum_{l=1}^{D+1} \overline{\alpha}_{jl}\right)}{\prod_{l=1}^{D+1} \Gamma\left(\overline{\alpha}_{jl}\right)} \\ & + \sum_{l=1}^{D+1} \overline{\alpha}_{jl} \left[ \psi\left( \sum_{l=1}^{D+1} \overline{\alpha}_{jl} \right) - \psi\left(\overline{\alpha}_{jl}\right) \right] \left[ \left\langle \ln \alpha_{jl} \right\rangle - \ln \overline{\alpha}_{jl} \right] \\ & + \frac{1}{2} \sum_{l=1}^{D+1} \overline{\alpha}_{jl}^2 \left[ \psi'\left( \sum_{l=1}^{D+1} \overline{\alpha}_{jl} \right) - \psi'\left(\overline{\alpha}_{jl}\right) \right] - \left\langle \left( \ln \alpha_{jl} - \ln \overline{\alpha}_{jl} \right)^2 \right\rangle \\ & + \frac{1}{2} \sum_{a=1}^{D+1} \sum_{b=1}^{D+1} \overline{\alpha}_{ja} \overline{\alpha}_{jb} \left[ \psi'\left( \sum_{l=1}^{D+1} \overline{\alpha}_{jl} \right) \left( \left\langle \ln \overline{\alpha}_{ja} \right\rangle - \ln \overline{\alpha}_{ja} \right) \right. \\ & \left. \times \left( \left\langle \ln \overline{\alpha}_{jb} \right\rangle - \ln \overline{\alpha}_{jb} \right) \right] \end{aligned} \tag{11.36}$$

Later, we maximize the lower bound $\mathcal{L}^{(t)}(Q)$ with respect to $Q^{(t)}(\alpha)$ and $\pi_j^{(t)}$ while $Q(\mathcal{Z}_t)$ is fixed. As mentioned before, here we consider variational inference as a natural gradient method. Therefore, the coefficient matrix for the posterior parameter distribution gets canceled since the natural gradient of a parameter is obtained by multiplying the gradient by the inverse of Riemannian metric. Therefore, the natural gradients for $\Delta u_{js}$, $\Delta v_{js}$ for $j = 1, 2, \ldots, M$, and $s = 1, 2, \ldots, D+1$ are

$$\begin{aligned} \Delta u_{js} = & \overline{\alpha}_{js} \left[ \psi\left( \sum_{l=1}^{D+1} \overline{\alpha}_{jl} \right) - \psi\left(\overline{\alpha}_{js}\right) \right. \\ & \left. + \psi'\left( \sum_{l=1}^{D+1} \overline{\alpha}_{jl} \right) \times \overline{\alpha}_{jl} \left( \left\langle \ln \alpha_{jl} \right\rangle - \ln \overline{\alpha}_{jl} \right) \right] \sum_{i=1}^{N} r_{ij} \end{aligned} \tag{11.37}$$

$$\Delta v_{js} = -\sum_{i=1}^{N} r_{ij}[\ln X_{is} - \ln(1 + \sum_{l=1}^{D} X_{il})] \tag{11.38}$$

Thus, the variational solution to $Q^{(t)}(\alpha)$ is given by

$$Q^{(t)}(\alpha) = \prod_{j=1}^{M} \prod_{l=1}^{D+1} G(\alpha_{jl}^{*} | u_{jl}^{*}, v_{jl}^{*}) \tag{11.39}$$

Therefore, we update the hyper-parameters and optimal variational parameters as

$$u_{jl}^{(t)} = u_{jl}^{(t-1)} + \rho_t \Delta u_{jl} \tag{11.40}$$

$$v_{jl}^{(t)} = v_{jl}^{(t-1)} + \rho_t \Delta v_{jl} \tag{11.41}$$

where $\rho_t$ is learning rate in which $\epsilon \in (0,1)$ and $\eta_o \geq 0$ are defined as

$$\rho_t = (\eta_o + t)^{-\epsilon} \tag{11.42}$$

The function of the learning rate here is adopted from [45] and is used to forget the earlier inaccurate estimation effects that contributed to the lower bound and expedite the convergence of the learning process. Online learning embraces the fact that learning environments can (and do) change from second to second. The mixing coefficient $\pi_{jl}^{(t)}$ is given by

$$\pi_{jl}^{(t)} = \pi_{jl}^{(t-1)} + \rho_t \Delta \pi_{jl} \tag{11.43}$$

where $\Delta \pi_j$ is

$$\Delta \pi_j = \frac{1}{N} \sum_{i=1}^{N} r_{ij} - \pi_j^{(t-1)} \tag{11.44}$$

The variational lower bound in case of online variational inference does not always increase whereas in batch variational it does because in case of online learning a new contribution is always added to the lower bound for each new observation. It is very important to choose the hyper-parameters and the learning rate accurately since it might affect the convergence of the model.

**Algorithm 1** Online Variational learning of the finite inverted Dirichlet mixture model

1. Choose the initial number of components $M$.
2. Initialize the value of hyper-parameters values for $u_{jl}$ and $v_{jl}$.
3. Using K-means algorithm, initialize the value of $r_{ij}$.
4. **for** $t = 1 \rightarrow N$ **do**
   - i The variational E-step:
   - ii Update the variational solutions for $Q(Z_t)$ using $r_{ij}$
   - iii The variational M-step:
   - iv Compute the learning rate $\rho_t = (\eta_o + t)^{-\epsilon}$
   - v Calculate the natural gradients $\Delta u_{js}$, $\Delta v_{js}$ and $\Delta \pi_j$ using (36), (37) and (43) respectively.
   - vi Update the variational solution for $Q^{(t)}(\alpha)$ and the mixing coefficient $\pi_{jl}^{(t)}$ through (38) and (42)
   - vii Repeat the variational E-step and M-step until new data is observed.
5. **end for**

## 11.6 Experimental Results

In order to evaluate the performance of our proposed algorithm we first validate it on synthetic data sets of varied sizes. Once the algorithm is validated, we further apply it on real world medical image data sets which are available with ground truth to perform segmentation and analysis of diseases. In our case, we performed medical image segmentation on three data sets of different diseases and different medical image testing techniques. We applied the algorithm to detect brain tumour, skin lesion, and tuberculosis. Furthermore, we have used three different formats of images to test the applicability of the algorithm on varied output formats, namely MRI scans, normal photographs, and X-ray images.

In order to have an insight on the accuracy of our algorithm we further compared it to the implementation of online variational inference of finite Gaussian mixture model on the data sets. We chose online variational inference of finite Gaussian mixture model as the comparison algorithm since Gaussian mixtures are widely applied in medical applications.

### *11.6.1 Image Segmentation*

Image segmentation is a key challenge in image analysis. In medical imaging, it is a particularly difficult challenge due to high variability in the image data sets. This variability arises due to two reasons. One, each human itself has variability in the anatomy of the organ or tissue. Second, there is an additional technical variability introduced to the images due to the different modalities (e.g., MRI, PET scans, CT scans, etc.) by which the image is created.

Let's say we have an input observed dataframe X which contains N pixels such that X= $\{X_1, \ldots X_N\}$. Each pixel is modeled as a mixture of M inverted Dirichlet distributions:

$$p(\mathbf{X}_i \mid \boldsymbol{\pi}, \boldsymbol{\alpha}) = \sum_{j=1}^{M} \pi_j ID(\mathbf{X}_i \mid \boldsymbol{\alpha}_j) \tag{11.45}$$

where $X_i$ is the pixel intensity value. We normalize the pixel values of an input image to unit sum. In all our experiments, we initialize the number of components $M$ to 15. The parameters of the $\epsilon$ and $\eta_o$ learning rate are set to 0.1 and 64, respectively. The accuracy of the algorithm was verified by comparison with the ground truth that was available for each data set. According to our experiments, a good choice of the initial values of the hyper-parameters $u_{jl}$ and $v_{jl}$ are discovered to be 1 and 0.01, respectively. We can thus detect the optimal number of the components $M$ by eliminating the components with the small mixing coefficients close to 0.

### *11.6.2 Synthetic Data*

The goal of using synthetic data is to investigate the accuracy of the online variational approach for both parameter estimation and model selection. Therefore, we first tested the model accuracy on synthetic data sets. These data sets consisted of different data sizes, namely 300, 400, 600, 800, and 1000. The effectiveness of the algorithm was tested by estimating the mixture parameters. Table 11.1 represents a comparison of the estimation performed by online variational learning of inverted Dirichlet mixture model versus the real parameters. It is noted that our algorithm can determine mixing coefficient parameters ($\hat{\alpha}_{j1}, \hat{\alpha}_{j2}, \hat{\alpha}_{j3}$ and $\hat{\pi}_j$) close to the real data parameters ( $\alpha_{j1}, \alpha_{j2}, \alpha_{j3}$ and $\pi_j$).

There are different ways that can be used for estimation of the number of components. In our case, once the algorithm reached convergence, we removed the components with very small (less than $10^{-5}$) mixing coefficients in each data set.

### *11.6.3 Medical Image Data Sets*

After validating the algorithm on synthetic data sets, we applied it on three biomedical image data sets. These data sets were used to detect three different disease morphologies which were created using three different imaging techniques. These data sets were MRI scan of brain tumours, X-ray scans of lung tuberculosis, and normal png format pictures of skin lesions. We observed that our algorithm could detect the morphological and structural anomalies similar to the ground truth data. We used 25 images in each case and compared the results of our proposed

**Table 11.1** Real and estimated parameters of different data sets

| Data set | $N_j$ | j | $\alpha_{j1}$ | $\alpha_{j2}$ | $\alpha_{j3}$ | $\pi_j$ | $\hat{\alpha}_{j1}$ | $\hat{\alpha}_{j2}$ | $\hat{\alpha}_{j3}$ | $\hat{\pi}_j$ |
|---|---|---|---|---|---|---|---|---|---|---|
| S1 | 100 | 1 | 7 | 25 | 79 | 0.33 | 7.43 | 25.62 | 80.28 | 0.33 |
| $(N = 300)$ | 100 | 2 | 11 | 32 | 63 | 0.34 | 10.41 | 31.81 | 62.97 | 0.34 |
| | 100 | 3 | 22 | 45 | 51 | 0.33 | 22.37 | 43.99 | 51.09 | 0.33 |
| S2 | 200 | 1 | 7 | 25 | 79 | 0.50 | 6.89 | 25.82 | 81.9 | 0.49 |
| $(N = 400)$ | 200 | 2 | 11 | 32 | 63 | 0.50 | 11.52 | 33.9 | 67.36 | 0.51 |
| S3 | 200 | 1 | 7 | 25 | 79 | 0.33 | 7.56 | 25.62 | 79.3 | 0.33 |
| $(N = 600)$ | 200 | 2 | 11 | 32 | 63 | 0.34 | 11.26 | 31.55 | 62.78 | 0.34 |
| | 200 | 3 | 22 | 45 | 51 | 0.33 | 22.42 | 46.66 | 51.55 | 0.33 |
| S4 | 200 | 1 | 7 | 25 | 79 | 0.25 | 7.5 | 25.53 | 82.88 | 0.26 |
| $(N = 800)$ | 200 | 2 | 11 | 32 | 63 | 0.25 | 11.32 | 33.21 | 68.4 | 0.24 |
| | 400 | 3 | 22 | 45 | 51 | 0.50 | 21.2 | 44.55 | 50.87 | 0.5 |
| S5 | 200 | 1 | 7 | 25 | 79 | 0.25 | 6.98 | 24.95 | 78.76 | 0.24 |
| $(N = 800)$ | 200 | 2 | 11 | 32 | 63 | 0.25 | 10.21 | 29.43 | 60.29 | 0.25 |
| | 200 | 3 | 22 | 45 | 51 | 0.25 | 22.11 | 45.11 | 50.85 | 0.27 |
| | 200 | 4 | 28 | 83 | 90 | 0.25 | 28.75 | 85.4 | 93.05 | 0.24 |
| S6 | 200 | 1 | 7 | 25 | 79 | 0.20 | 7.32 | 24.95 | 76.64 | 0.20 |
| $(N = 1000)$ | 200 | 2 | 11 | 32 | 63 | 0.20 | 11.86 | 34.79 | 68.16 | 0.17 |
| | 200 | 3 | 22 | 45 | 51 | 0.20 | 23 | 46.1 | 51.59 | 0.21 |
| | 200 | 4 | 28 | 83 | 90 | 0.20 | 28.71 | 83.81 | 88.55 | 0.22 |
| | 200 | 5 | 40 | 3 | 56 | 0.20 | 37.94 | 2.95 | 55.05 | 0.20 |

$N$ denotes the total number of data points, $N_j$ denotes the number of data points in the cluster $j$. $\alpha_{j1}, \alpha_{j2}, \alpha_{j3}$ and $\pi_j$ are the real parameters and $\hat{\alpha}_{j1}, \hat{\alpha}_{j2}, \hat{\alpha}_{j3}$ and $\hat{\pi}_j$ are the parameters estimated by our proposed algorithm

algorithm with online variational learning for Gaussian mixture model to determine the model performance.

#### 11.6.3.1 Brain Tumour Detection

Gliaomas or brain tumours are the most prominent brain malignancies which exhibit varying degrees of aggressiveness, prognosis, and inherent variability in the MRI image representation. Due to the heterogeneous nature of the brain anatomy, the MRI image segmentation and tumor detection is a highly challenging task [46]. For this, the brain tumour data set was obtained from BRATS2015[1] [47, 48]. The data set consists of four MRI sequence images for each patient. The MRI sequence images were fluid attenuation inversion recovery (FLAIR), T1c, T1p, and T2 which all stand for images which are weighted with respect to the relaxation time of protons in the body tissue during the scanning. FLAIR is widely applied

[1] https://www.smir.ch/BRATS/Start2015.

to detect clinical malformations related to diseases like multiple sclerosis (MS), haemorrhages, meningitis, etc. [49]. In our experiment, we used the available FLAIR images for image segmentation and brain tumour detection.

The resulting accuracy of brain MRI segmentation was measured using Jaccard and Dice metrics. This is illustrated in Fig. 11.2, and the mean and standard deviation are illustrated in Fig. 11.3. The Jaccard and Dice for the BRATS2015[2] data set were significantly greater for our proposed algorithm than the online variational Gaussian mixture model helping us conclude that it can be useful to detect tumours. The mean was 0.5 greater than the compared algorithm, and the standard deviation was comparatively less showing the robustness of our model.

Representative segmentation results after running our proposed algorithm are depicted in Fig. 11.4 where the three clusters generated by the algorithm are depicted against the MRI image. The last image in the panel is the best prediction made by the algorithm, and it is seen that the algorithm is able to identify the brain glioma. Further, the post processing images are depicted in Fig. 11.5 where the last image in the panel is the post processed ground truth image. The predicted image by the algorithm was compared against the ground truth. We are able to visibly see the similarities of the detection by the algorithm versus an expert's opinion.

#### 11.6.3.2 Skin Lesion Diagnosis

Similarly to brain gliomas, skin melanomas are also difficult to detect. Specially because the naked eye is not able to differentiate between the malignant and benign skin melanoma [50]. Therefore, digital imaging and lesion detection with identification can help increase the efficiency in the detection and treatment [51]. Furthermore, since skin is the largest organ of the body and highly visible, taking photos of the melanomas from smart phones would add convenience in the process. However, analysis of smart phone medical images is also a challenging task due to the heterogeneity [52, 53]. For this reason, the data used for assessing the performance of the proposed algorithm were done on the photos of skin lesion obtained from International Skin Imaging Collaboration.[3] The data set consists of images of skin melanoma of patients.

The accuracy of the result obtained from skin image segmentation is measured by Jaccard and Dice metrics as illustrated in Fig. 11.6, and the mean and standard deviation are shown in Fig. 11.7 by comparing the proposed algorithm with online variational finite Gaussian mixture model. The Jaccard index and Dice coefficient for the data set were significantly greater for our proposed algorithm since both the values for each image were above 0.85. The mean was 0.7 greater than the compared algorithm, and the standard deviation was 0.05 less for our algorithm

[2] Same as footnote 1.

[3] https://isic-archive.com/api/v1.

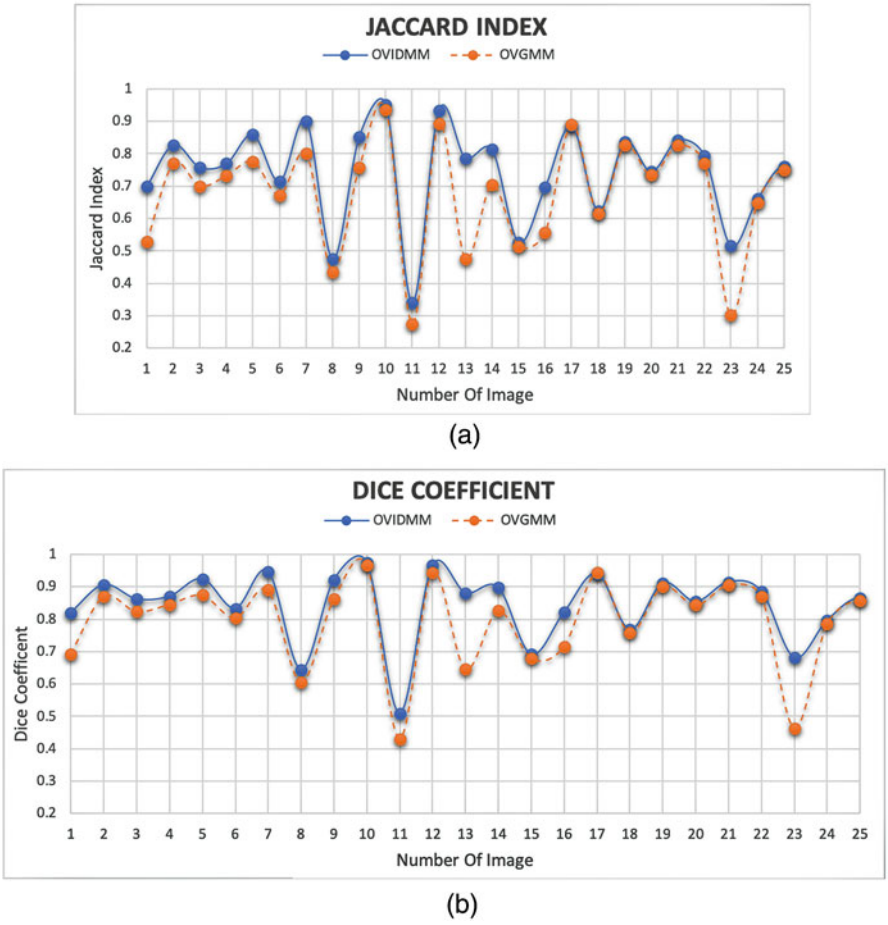



**Fig. 11.2** Results using Jaccard and dice evaluation metrics for brain tumour detection. (**a**) Jaccard metric. (**b**) Dice metric

proving the robustness of our algorithm. This demonstrates the accuracy of our model for predicting skin lesions.

Figure 11.8 shows a representative image of skin melanoma from the ISIC database (left panel, first photo), and the best segmented and detected melanoma by the algorithm can be seen at the end of the panel in the figure. In this case, the algorithm was able to detect 14 clusters. Figure 11.9 displays a representative skin melanoma image achieved after post processing for the ground truth in order to compare it with the algorithm.

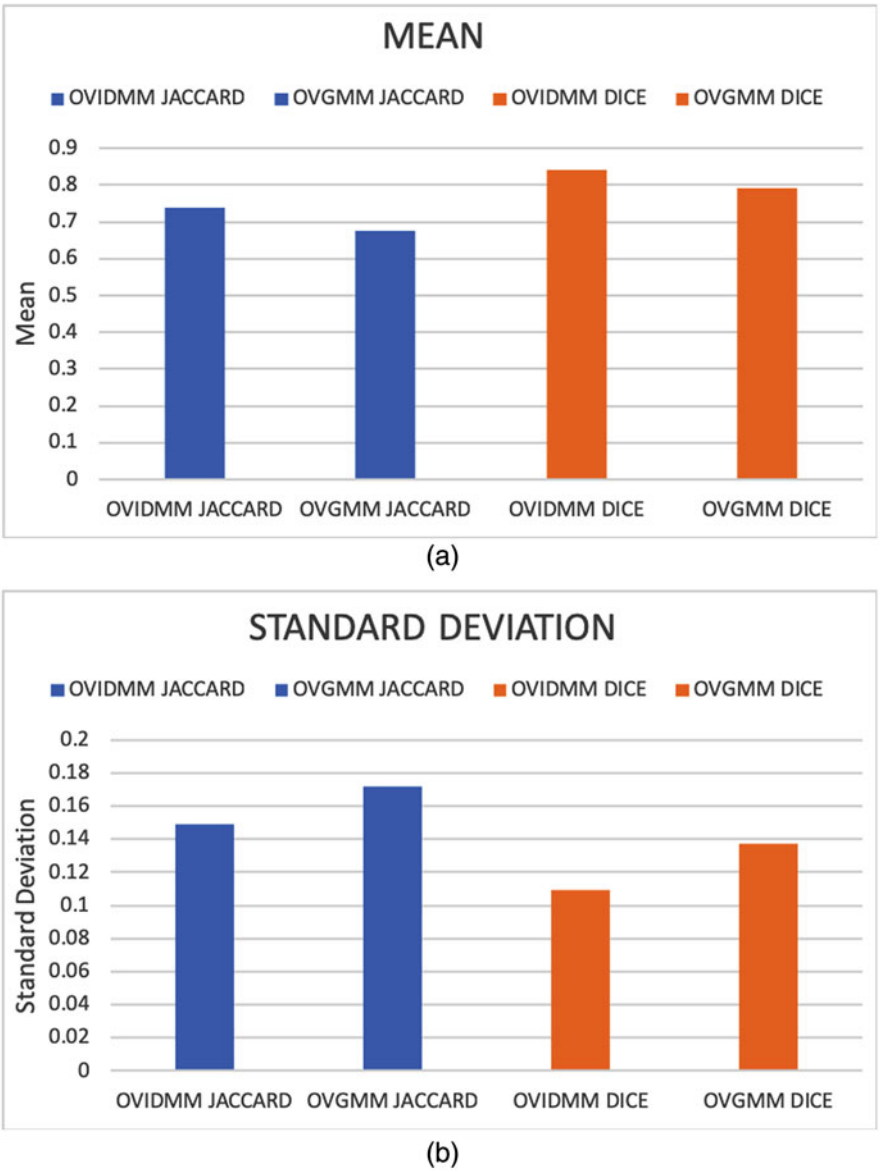



**Fig. 11.3** Mean and standard deviation results for brain tumour detection. (**a**) Mean. (**b**) Standard deviation

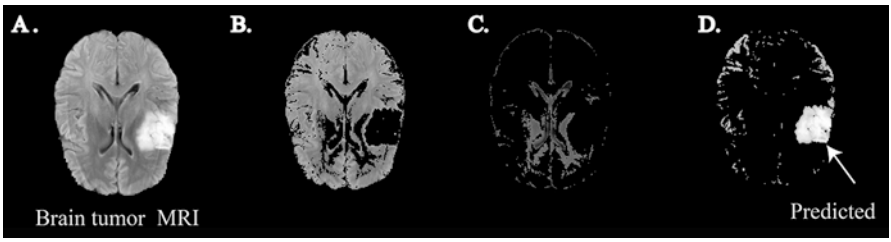



**Fig. 11.4** Best segmented brain MRI images: (**a**) Input image, (**b**) 3rd Cluster, (**c**) 6th Cluster, (**d**) 7th Cluster

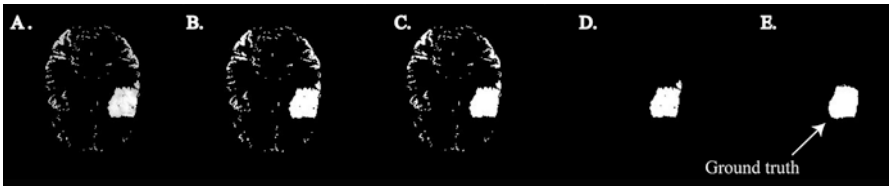



**Fig. 11.5** Segmented brain MRI images after post processing: (**a**) Clustered image, (**b**) Binary image, (**c**) Clustered after filling holes, (**d**) Processed clustered image and (**e**) Ground truth image. The data set was taken from BRATS database [47, 48] where the ground truth data were available

#### 11.6.3.3 Lung Tuberculosis Detection

Tuberculosis is caused by *Mycobacterium tuberculosis* which majorly infects the lung but can spread rapidly through the body [54]. X-ray is currently the most common diagnostic tool used to detect tuberculosis. However, a lot of time the infection goes undetected due to the high intrinsic noise in the X-ray measurements [55]. Besides, in a low resource setup X-ray interpretations are performed by non-experts [56]. Here, a digital analysis of detection can lend to computer-aided decision support. Therefore, the third data set used for this analysis is an X-ray image selected from collection of data compiled by National Library of Medicine in collaboration with the Department of Health and Human Services, Montgomery County, Maryland, USA [57, 58]. The sample set is composed of 58 cases with manifestation of tuberculosis and 80 normal cases. Each image is gray-scale with a spatial resolution of $4020 \times 4892$, or $4892 \times 4020$. We performed our algorithm on 25 images and on cases where tuberculosis was detected. It is to be noted that we compared only the right mask of the lung for the algorithm predictions as the ground truth was available for that.

The accuracy obtained by performing lung segmentation is given by Jaccard and Dice metrics as illustrated in Fig. 11.10, and the mean and standard deviation are shown in Fig. 11.11. The Jaccard and Dice for this data set were significantly higher for our proposed algorithm since each image had a considerable difference in the value from the online variational learning of finite Gaussian mixture model. The

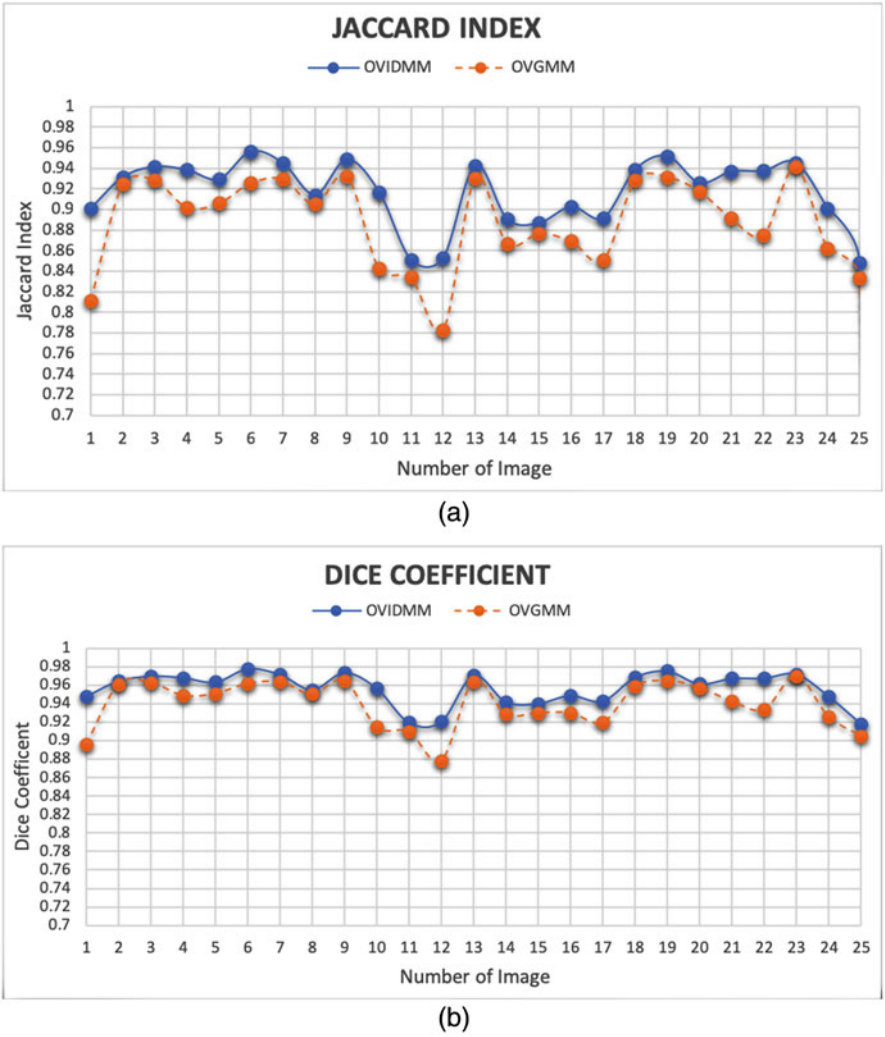
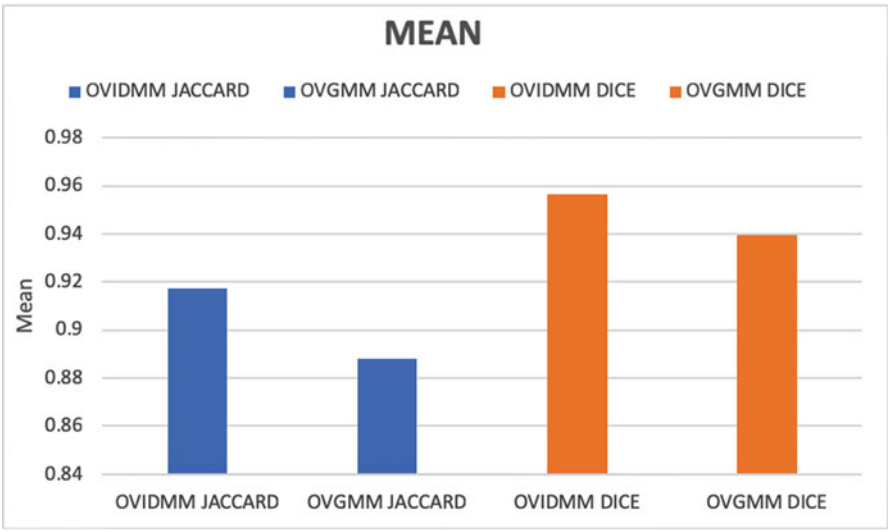


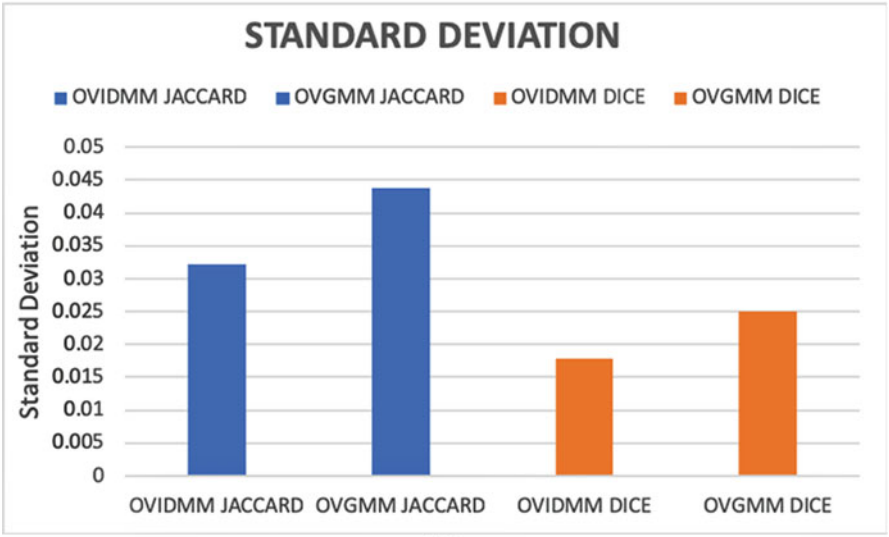

**Fig. 11.6** Results using Jaccard and Dice evaluation metrics for skin lesion diagnosis. (**a**) Jaccard metric. (**b**) Dice metric

mean was 0.11 greater than the compared algorithm, and the standard deviation was comparatively less for our algorithm showing the strength of our model.

Figure 11.12 is a representative image of the algorithm prediction where the top four clusters are depicted. In the panel the first image is that of the X-ray and the last image is of the best predicted tuberculosis image by the algorithm. There were 14 clusters generated by the algorithm which are not shown here. Figure 11.13 depicts the images of the same lung X-ray segmentation after post processing on




(a)




(b)

**Fig. 11.7** Mean and standard deviation results for skin lesion diagnosis. (**a**) Mean. (**b**) Standard deviation

segmentation. It can be clearly seen in the last images of Fig. 11.12 (predicted) and Fig. 11.13 (ground truth) that the algorithm is able to capture the similar segment of tuberculosis in the right lung.

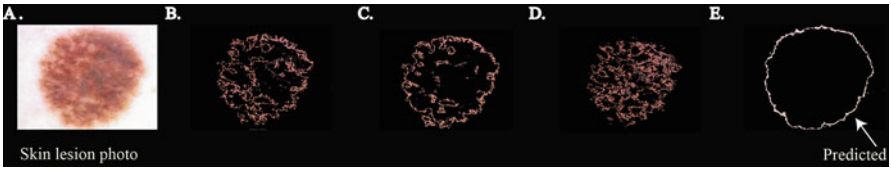



**Fig. 11.8** Best Segmented Skin Lesion Images: (**a**) Input image, (**b**) 0th Cluster, (**c**) 9th Cluster, (**d**) 14th Cluster (**e**) 10th Cluster

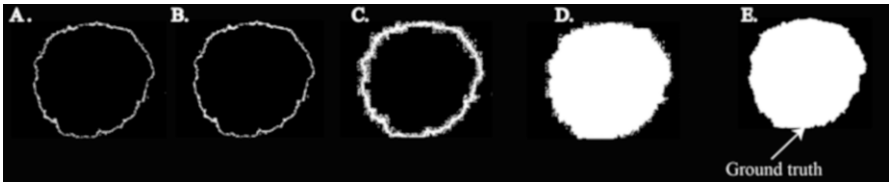



**Fig. 11.9** Segmented Skin lesion images after post processing: (**a**) Clustered image, (**b**) Greyscale image, (**c**) Binary image, (**d**) Binary image after filling holes, and (**e**) Ground truth image. The data set was taken from ICIS database where the ground truth data were available

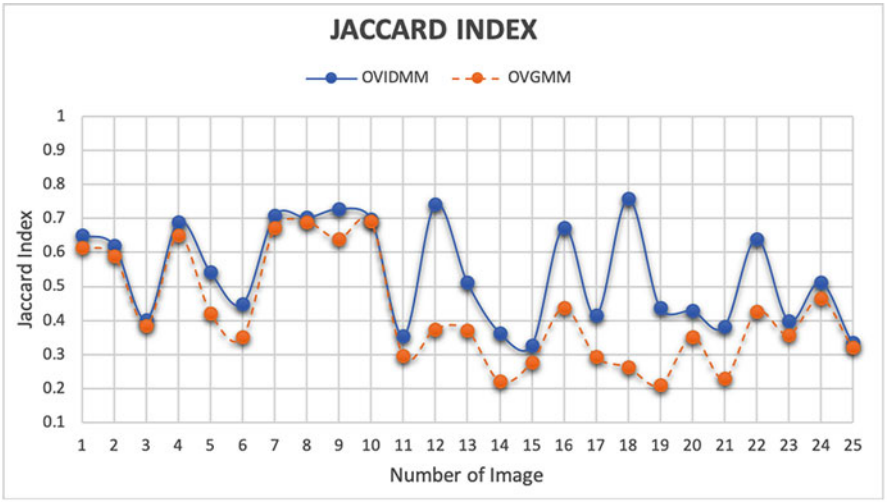
## 11.7 Conclusion

Computational and statistical approaches like the one presented in this chapter hold a significant impact on medical image analysis and interpretation in both clinical applications and scientific research. Recent progress in the development of unsupervised algorithms and their implementation as a method for medical image analysis has enabled efficient discovery and determination of morphological and textural patterns in the images solely from the data provided to the algorithm.

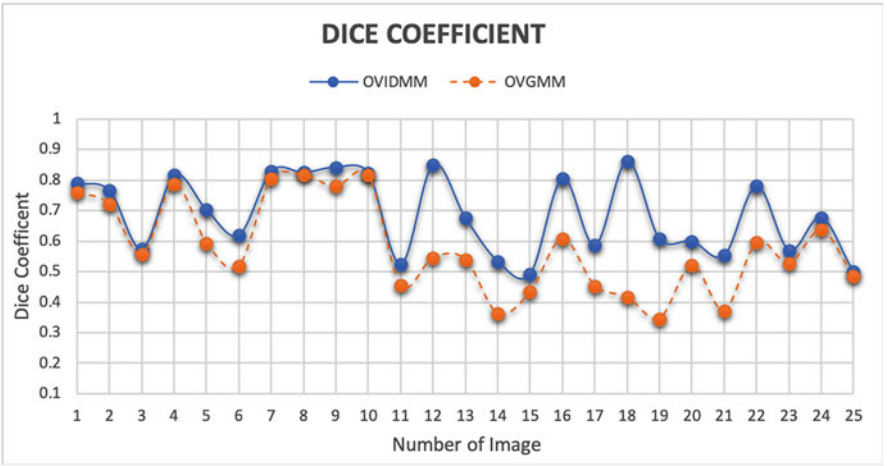
This chapter proposed an online variational algorithm for finite inverted Dirichlet mixture models learning. With the implementation of this method, we depict the advantages of estimating the full posterior distribution of the model parameters in contrast with the maximum likelihood approach.

The effectiveness of the proposed approach has been evaluated on both synthetic data sets and medical image data. In comparison with the online variational finite Gaussian mixture model algorithm, our online inverted Dirichlet mixture model algorithm is much more efficient and effective on multi-modal data sets. The experimental results show the validity of the proposed approach in terms of parameter estimation and model selection on different data sets.

However, there are certain restrictions in the application of this approach on medical image data sets. First, as witnessed in this chapter, the application of this approach would improve its performance by the use of large data sets where the method can find more generalized feature. Second, although the unsupervised feature extraction and representation have enhanced accuracy, it is to be noted that the methodological architecture to achieve this requires domain-specific knowledge due to the scarce availability of ground truth data [59, 60]. Third, due to the vast



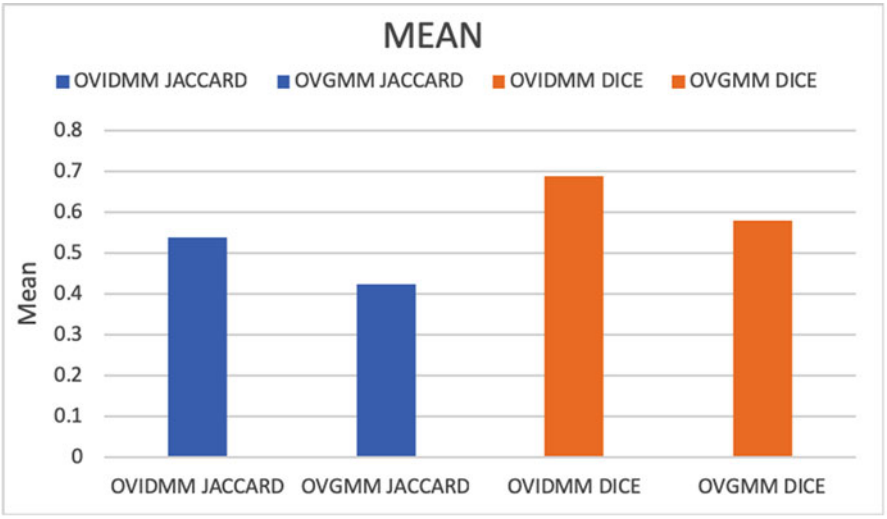

(a)



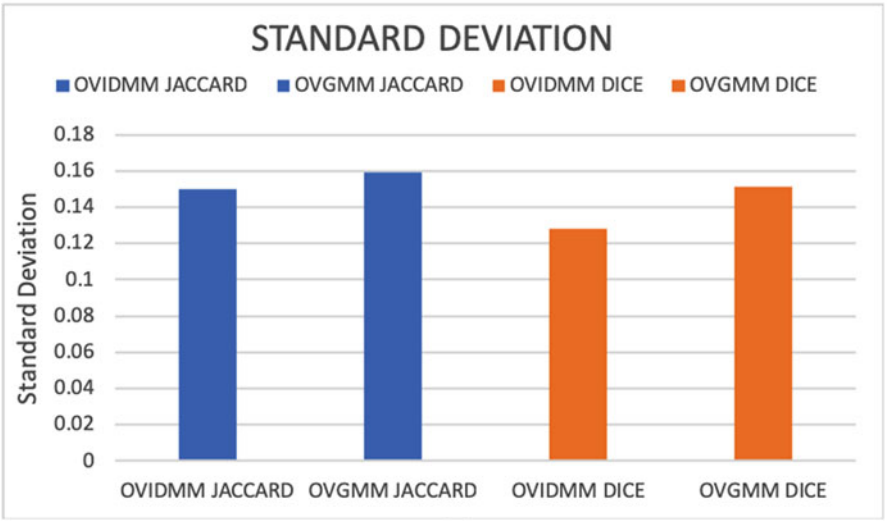

(b)

**Fig. 11.10** Results using Jaccard and Dice evaluation metrics for lung tuberculosis detection. (**a**) Jaccard metric. (**b**) Dice metric

expanse of medical imaging instruments and techniques, it is of high importance to develop algorithms and methods to efficiently acquire the images from these techniques such that they can be further handled effectively by the use of the unsupervised algorithms [61]. Finally, it is to be noted that the use of an analysis where the posterior distribution is of importance can be deemed impractical. This typically happens when the model and its interaction with the parameters are too




(a)




(b)

**Fig. 11.11** Mean and standard deviation results for lung tuberculosis detection. (**a**) Mean. (**b**) Standard deviation

complex for posterior distributions to be calculated accurately. In such cases, there is no option but to revert to methods where point estimates are derived.

The future work can be devoted to include feature selection component with the proposed model along with an extension of online model to generalized inverted Dirichlet mixtures which would help us improve the model learning.

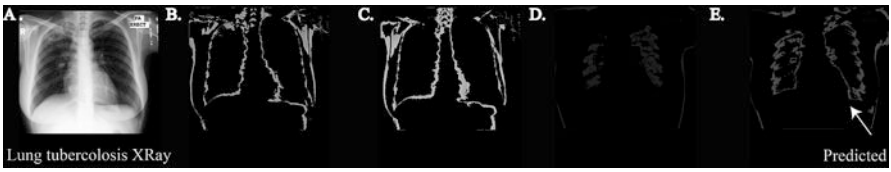



**Fig. 11.12** Best segmented Lung images: (**a**) Input image, (**b**) 10th Cluster, (**c**) 7th Cluster, (**d**) 4th Cluster, (**e**) 0th Cluster

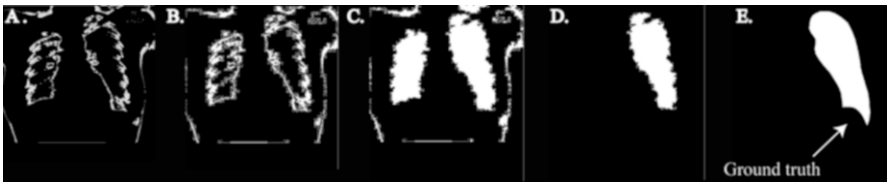



**Fig. 11.13** Lung X-ray after post processing: (**a**) Clustered image, (**b**) Binary image, (**c**) Clustered after filling holes, (**d**) Processed cluster and (**e**) Ground truth image. The data set was taken from Montgomery County—Chest X-ray Database provided by National Library of Medicine where the ground truth data were available

## Appendix

### *Proof of Eq. (11.17): Variational Solution of $Q(\mathcal{Z})$*

For the variational solution $Q_s(\Theta_s)$, the general expression expressed as:

$$\ln Q_s(\Theta_s) = \left\langle \ln p(X, \Theta) \right\rangle_{j \neq s} + const \tag{11.46}$$

where *const* is an additive term representing every term that is independent of $Q_s(\Theta_s)$. Now consider the joint distribution in Eq. (11.10), the variational solution for $Q(\mathcal{Z})$ can be derived as follows:

$$\ln Q(\mathcal{Z}) = \alpha_{ij} \left[ \ln \pi_j + \mathcal{R}_j + \sum_{l=1}^{D+1} (\alpha_{jl} - 1) \ln X_{il} \right] + const \tag{11.47}$$

Where

$$\mathcal{R}_j = \left\langle \ln \frac{\Gamma(\sum_{l=1}^{D+1} \alpha_{jl}}{\prod_{D+1_{l=1}} \Gamma(\alpha_{jl})} \right\rangle_{\alpha_{jl}, \ldots \alpha_{jD+1}} \tag{11.48}$$

and

$$\alpha_{jl} = \langle \alpha_{jl} \rangle = \frac{u_{jl}}{v_{jl}} \tag{11.49}$$

Since we don't have a closed form solution for $\mathcal{R}_j$, therefore it is not possible to directly apply the variational inference. Therefore, in order to provide traceable approximations, the second-order Taylor's expansion is used to approximate the expected values of parameters $\alpha_j$ [14]. Hence, considering the logarithm form of (11.6), Eq. (11.47) can be written as

$$\ln Q(\mathcal{Z}) = \sum_{i=1}^{N} \sum_{j=1}^{M} \mathcal{Z}_{ij} \ln \rho_{ij} + const \tag{11.50}$$

where

$$\ln \rho_{ij} = \ln \pi_j + \mathcal{R}_j + \sum_{l=1}^{D} (\alpha_{jl} - 1) \ln X_{il} \tag{11.51}$$

Since all the term without $\mathcal{Z}_{ij}$ can be added to the constant, it possible to show that

$$Q(\mathcal{Z}) \propto \prod_{i=1}^{N} \prod_{j=1}^{M} \rho_{ij}^{Z_{ij}} \tag{11.52}$$

To find the exact formula for $Q(\mathcal{Z})$, Eq. (11.53) should be normalized and the calculation can be expressed as

$$Q(\mathcal{Z}) = \prod_{i=1}^{N} \prod_{j=1}^{M} r_{ij}^{Z_{ij}} \tag{11.53}$$

where

$$r_{ij} = \frac{\rho_{ij}}{\sum_{j=1}^{M} \rho_{ij}} \tag{11.54}$$

It is noteworthy that $\sum_{j=1}^{M} r_{ij} = 1$, thus the result for $Q(\mathcal{Z})$ is

$$\langle \mathcal{Z}_{ij} \rangle = r_{ij} \tag{11.55}$$

## *Proof of Eqs. (11.18), (11.22) and (11.23)*

Assuming the parameters $\alpha_{jl}$ are independent in a mixture model with $M$ components, we can factorize $Q(\alpha)$ as

$$Q(\alpha) = \prod_{j=1}^{M} \prod_{l=1}^{D+1} Q(\alpha_{jl}) \tag{11.56}$$

We compute the variational solution for the $\mathrm{Q}(\alpha_{jl})$ by using Eq. (11.16) instead of using the gradient method. The logarithm of the variational solution $\mathrm{Q}(\alpha_{jl})$ is given by,

$$\begin{aligned}\ln Q(\alpha_{jl}) =& \langle \ln p(\mathcal{X}, \Theta) \rangle_{\Theta \neq \alpha_{jl}} \\ =& \sum_{i=1}^{N} \langle Z_{ij} \rangle \mathcal{J}(\alpha_{jl}) + \alpha_{jl} \sum_{i=1}^{N} \langle Z_{ij} \rangle \ln X_{il} - \alpha_{jl} \ln \left(1 + \sum_{l=1}^{D+1} X_{il}\right) \\ &+ (u_{jl} - 1) \ln \alpha_{jl} - v_{jl} \alpha_{jl} + \text{const}\end{aligned} \tag{11.57}$$

where,

$$\mathcal{J}(\alpha_{jl}) = \left\langle \ln \frac{\Gamma\left(\alpha_{jl} + \sum_{s \neq l}^{D+1} \alpha_{js}\right)}{\Gamma(\alpha_{jl}) \prod_{s \neq l}^{D+1} \Gamma(\alpha_{js})} \right\rangle_{\Theta \neq \alpha_{jl}} \tag{11.58}$$

Similar to what we encountered in the case of $R_j$, the equation for $\mathcal{J}(\alpha_{jl})$ is also intractable. We solve this problem finding the lower bound for the equation by calculating the first-order Taylor expansion with respect to $\overline{\alpha}_{jl}$. The calculated lower bound is given by [44],

$$\begin{aligned}\mathcal{J}(\alpha_{jl}) \geq \overline{\alpha}_{jl} \ln \alpha_{jl} \Bigg[ & \psi\left(\sum_{l=1}^{D+1} \overline{\alpha}_{jl}\right) - \psi(\overline{\alpha}_{jl}) + \sum_{s \neq l}^{D+1} \overline{\alpha}_{js} \\ & \times \psi'\left(\sum_{l=1}^{D+1} \overline{\alpha}_{jl}\right) \left( \langle \ln \alpha_{js} \rangle - \ln \overline{\alpha}_{js} \right) \Bigg] + \text{const}\end{aligned} \tag{11.59}$$

Substituting this equation for lower bound in Eq. (11.57)

$$\ln Q(\alpha_{jl}) = \sum_{i=1}^{N} \langle Z_{ij} \rangle \overline{\alpha}_{jl} \ln \alpha_{jl} \left[ \psi\left(\sum_{l=1}^{D+1} \overline{\alpha}_{jl}\right) - \psi(\overline{\alpha}_{jl}) \right.$$

$$+\sum_{s\neq l}^{D+1}\overline{\alpha}_{js}\psi'\Big(\sum_{l=1}^{D+1}\overline{\alpha}_{jl}\Big)\big(\langle\ln\alpha_{js}\rangle-\ln\overline{\alpha}_{js}\big)\Big]$$
$$+\alpha_{jl}\sum_{i=1}^{N}\langle Z_{ij}\rangle\ln X_{il}-\alpha_{jl}\ln\Big(1+\sum_{l=1}^{D+1}X_{il}\Big)$$
$$+\big(u_{jl}-1\big)\ln\alpha_{jl}-v_{jl}\alpha_{jl}+\text{const} \tag{11.60}$$

This equation can be rewritten as,

$$\ln Q\big(\alpha_{jl}\big)=\ln\alpha_{jl}\big(u_{jl}+\varphi_{jl}-1\big)-\alpha_{jl}\big(v_{jl}-\vartheta_{jl}\big)+\text{const} \tag{11.61}$$

where,

$$\varphi_{jl}=\sum_{i=1}^{N}\langle Z_{ij}\rangle\overline{\alpha}_{jl}\Big[\psi\Big(\sum_{l=1}^{D+1}\overline{\alpha}_{jl}\Big)-\psi\big(\overline{\alpha}_{jl}\big)$$
$$+\sum_{s\neq l}^{D+1}\overline{\alpha}_{js}\psi'\Big(\sum_{l=1}^{D+1}\overline{\alpha}_{jl}\Big)\big(\langle\ln\alpha_{js}\rangle-\ln\overline{\alpha}_{js}\big)\Big] \tag{11.62}$$

$$\vartheta_{jl}=\sum_{i=1}^{N}\langle Z_{ij}\rangle\Big[\ln X_{il}-\ln\Big(1+\sum_{l=1}^{D}X_{il}\Big)\Big] \tag{11.63}$$

Equation (11.61) is the logarithmic form of a gamma distribution. If we exponentiate both the sides, we get,

$$Q\big(\alpha_{jl}\big)\propto\alpha_{jl}^{u_{jl}+\varphi_{jl}-1}e^{-\big(v_{jl}-\vartheta_{jl}\big)\alpha_{jl}} \tag{11.64}$$

This leaves us with the optimal solution for the hyper-parameters $u_{jl}$ and $v_{jl}$ given by,

$$u_{jl}^{*}=u_{jl}+\varphi_{jl},\quad v_{jl}^{*}=v_{jl}-\vartheta_{jl} \tag{11.65}$$

## *Proof of Eq. (11.27)*

We calculate the mixing coefficients value $\pi$ by maximizing the lower bound w.r.t to $\pi$. It is essential to include Lagrangian term in the lower bound because of the constraint $\sum_{j=1}^{M}\pi_j=1$. Then, solving for the derivative w.r.t $\pi_j$ and setting the result to zero, we have [44]

$$\frac{\partial \mathcal{L}(Q)}{\partial \pi_j} = \frac{\partial \mathcal{L}(Q)}{\partial \pi_j} \sum_{i=1}^{N} \sum_{j=1}^{M} r_{ij} \ln \pi_j + \lambda \left( \sum_{j=1}^{M} \pi_j - 1 \right)$$

$$= \sum_{i=1}^{N} r_{ij} (1/\pi_j) + \lambda = 0 \tag{11.66}$$

$$\Rightarrow \sum_{i=1}^{N} r_{ij} = -\lambda \pi_j \tag{11.67}$$

By taking the sum of both sides of Eq. (11.67) over j, we can obtain $\lambda = -N$. Then substituting the value of $\lambda$ Eq. (11.66), we can obtain

$$\pi_j = \frac{1}{N} \sum_{i=1}^{N} r_{ij} \tag{11.68}$$

## References


1. Agrawal, J.P., Erickson, B.J., Kahn, C.E.: Imaging informatics: 25 years of progress. Yearb. Med. Inform. **Suppl 1**, 23–31 (2016)
2. Sohail, M.N., Jiadong, R., Uba, M.M., Irshad, M.: A comprehensive looks at data mining techniques contributing to medical data growth: A survey of researcher reviews. In: Patnaik, S., Jain, V. (eds.) Recent Developments in Intelligent Computing, Communication and Devices. Springer, Singapore, pp. 21–26 (2019)
3. Ganguly, D., Chakraborty, S., Balitanas, M., Kim, Th.: Medical imaging: A review. In: Kim, Th., Stoica, A., Chang, R.S. (eds.) Security-Enriched Urban Computing and Smart Grid. Springer, Heidelberg, pp. 504–516 (2010)
4. Perera, C.M., Chakrabarti, R.: A review of m-health in medical imaging. Telemed. e-Health **21**(2), 132–137 (2015)
5. Lester, D.S., Olds, J.L.: Biomedical imaging: 2001 and beyond. Anat. Rec. An Offi. Publ. Am. Assoc. Anatomists **265**(2), 35–36 (2001)
6. Van Beek, E.J., Hoffman, E.A.: Functional imaging: CT and MRI. Clin. Chest Med. **29**(1), 195–216 (2008)
7. Doi, K.: Computer-aided diagnosis in medical imaging: historical review, current status and future potential. Comput. Med. Imaging Graph. **31**(4–5), 198–211 (2007)
8. Petrick, N., Sahiner, B., Armato III, S.G., Bert, A., Correale, L., Delsanto, S., Freedman, M.T., Fryd, D., Gur, D., Hadjiiski, L., Huo, Z., Jiang, Y., Morra, L., Paquerault, S., Raykar, V., Samuelson, F., Summers, R.M., Tourassi, G., Yoshida, H., Zheng, B., Zhou, C., Chan, H.P.: Evaluation of computer-aided detection and diagnosis systems. Med. Phys. **40**(8), 087001 (2013)
9. Erickson, B.J., Korfiatis, P., Akkus, Z., Kline, T.L.: Machine learning for medical imaging. Radiographics **37**(2), 505–515 (2017)
10. Guadalupe Sanchez, M., Guadalupe Sánchez, M., Vidal, V., Verdu, G., Verdú, G., Mayo, P., Rodenas, F.: Medical image restoration with different types of noise. In: 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4382–4385 (2012)

11. Sittig, D.F., Wright, A., Osheroff, J.A., Middleton, B., Teich, J.M., Ash, J.S., Campbell, E., Bates, D.W.: Grand challenges in clinical decision support. J. Biomed. Inform. **41**(2), 387–392 (2008)
12. Chen, T.J., Chuang, K.S., Chang, J.H., Shiao, Y.H., Chuang, C.C.: A blurring index for medical images. J. Digit. Imaging **19**(2), 118–125 (2005)
13. Fan, W., Bouguila, N., Ziou, D.: Variational learning for finite Dirichlet mixture models and applications. IEEE Trans. Neural Netw. Learn. Syst. **23**(5), 762–774 (2012)
14. Tirdad, P., Bouguila, N., Ziou, D.: Variational learning of finite inverted Dirichlet mixture models and applications. In: Laalaoui, Y., Bouguila, N. (eds.) Artificial Intelligence Applications in Information and Communication Technologies, vol. 607, pp. 119–145. Springer, Cham (2015)
15. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods (Springer Texts in Statistics). Springer, Heidelberg (2005)
16. Gultepe, E., Makrehchi, M.: Improving clustering performance using independent component analysis and unsupervised feature learning. Hum-centric Comput. Inf. Sci. **8**(1), 148:1–148:19 (2018)
17. Fan, W., Bouguila, N., Ziou, D.: Variational learning of finite Dirichlet mixture models using component splitting. Neurocomputing **129**, 3–16 (2014)
18. Bouguila, N., Ziou, D.: Online clustering via finite mixtures of Dirichlet and minimum message length. Eng. Appl. Artif. Intell. **19**(4), 371–379 (2006)
19. Zakariya, S.M., Ali, R., Ahmad, N.: Combining visual features of an image at different precision value of unsupervised content based image retrieval. In: 2010 IEEE International Conference on Computational Intelligence and Computing Research, pp. 1–4 (2010)
20. Constantinopoulos, C., Likas, A.: Unsupervised learning of Gaussian mixtures based on variational component splitting. IEEE Trans. Neural Netw. **18**(3), 745–755 (2007)
21. Williams, G.: Descriptive and predictive analytics. In: Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery, pp. 171–177. Springer, New York (2011)
22. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. Data Min. Knowl. Disc. **15**(1), 55–86 (2007)
23. Bellazzi, R., Zupan, B.: Predictive data mining in clinical medicine: current issues and guidelines. Int. J. Med. Inform. **77**(2), 81–97 (2008)
24. Swan, M.: Emerging patient-driven health care models: an examination of health social networks, consumer personalized medicine and quantified self-tracking. Int. J. Environ. Res. Public Health **6**(2), 492–525 (2009)
25. Iavindrasana, J., Cohen, G., Depeursinge, A., Müller, H., Meyer, R., Geissbuhler, A. Clinical data mining: a review. Yearb. Med. Inform. 121–133 (2018)
26. Chechulin, Y., Nazerian, A., Rais, S., Malikov, K.: Predicting patients with high risk of becoming high-cost healthcare users in Ontario (Canada). Healthc. Policy **9**, 68–79 (2014)
27. Ramezankhani, A., Kabir, A., Pournik, O., Azizi, F., Hadaegh, F.: Classification-based data mining for identification of risk patterns associated with hypertension in middle eastern population: A 12-year longitudinal study. Medicine (Baltimore) **95**(35), e4143 (2016)
28. Parva, E., Boostani, R., Ghahramani, Z., Paydar, S.: The necessity of data mining in clinical emergency medicine; a narrative review of the current literature. Bull. Emerg. Trauma. **5**(2), 90–95 (2017)
29. Kuo, I.T., Chang, K.Y., Juan, D.F., Hsu, S.J., Chan, C.T., Tsou, M.Y.: Time-dependent analysis of dosage delivery information for patient-controlled analgesia services. PLoS One **13**(3), 1–13 (2018)
30. Lee, M.J., Chen, C.J., Lee, K.T., Shi, H.Y.: Trend analysis and outcome prediction in mechanically ventilated patients: A nationwide population-based study in Taiwan. PLoS One **10**(4), 1–13 (2015)
31. Baek, H., Cho, M., Kim, S., Hwang, H., Song, M., Yoo, S.: Analysis of length of hospital stay using electronic health records: A statistical and data mining approach. PLoS One **13**(4), 1–16 (2018)
32. Tiao, G.G., Cuttman, I.: The inverted Dirichlet distribution with applications. J. Am. Stat. Assoc. **60**(311), 793–805 (1965)

33. Xu, R., Wunsch, D.C.: Clustering algorithms in biomedical research: A review. IEEE Rev. Biomed. Eng. **3**, 120–154 (2010)
34. Wang, H.X., Luo, B., Zhang, Q.B., Wei, S.: Estimation for the number of components in a mixture model using stepwise split-and-merge EM algorithm. Pattern Recogn. Lett. **25**(16), 1799–1809 (2004)
35. Schneider, A., Hommel, G., Blettner, M.: Linear regression analysis: part 14 of a series on evaluation of scientific publications. Dtsch. Arztebl. Int. **44**, 776–82 (2010)
36. Kovalchuk, S.V., Funkner, A.A., Metsker, O.G., Yakovlev, A.N.: Simulation of patient flow in multiple healthcare units using process and data mining techniques for model identification. J. Biomed. Inform. **82**, 128–142 (2018)
37. Jensen, P.B., Jensen, L.J., Brunak, S.: Mining electronic health records: towards better research applications and clinical care. Nat. Rev. Genet. **13**(6), 395–405 (2012)
38. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: a review for statisticians. J. Am. Stat. Assoc. **112**(518), 859–877 (2017)
39. Corduneanu, A., Bishop, C.: Variational Bayesian model selection for mixture distributions. In: Proceedings Eighth International Conference on Artificial Intelligence and Statistics, pp. 27–34. Morgan Kaufmann, San Francisco (2001)
40. Lawrence, N.D., Bishop, C.M., Jordan, M.I.: Mixture Representations for Inference and Learning in Boltzmann Machines (2013). CoRR abs/1301.7393. 1301.7393
41. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Mach. Learn. **37**(2), 183–233 (1999)
42. Bishop, C.M., Lawrence, N., Jaakkola, T., Jordan, M.I.: Approximating posterior distributions in belief networks using mixtures. In: Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10, pp. 416–422. MIT Press, Cambridge (1998)
43. Amari, S.I.: Natural gradient works efficiently in learning. Neural. Comput. **10**(2), 251–276 (1998)
44. Fan, W., Bouguila, N.: Online variational learning of finite Dirichlet mixture models. Evol. Syst. **3**(3), 153–165 (2012)
45. Hoffman, M., Bach, F.R., Blei, D.M.: Online learning for latent Dirichlet allocation. In: Lafferty, J.D., Williams, C.K.I., Shawe-Taylor, J., Zemel, R.S., Culotta, A. (eds.) Advances in Neural Information Processing Systems, vol. 23, pp. 856–864. Curran Associates, Inc., (2010)
46. Bakas, S., Kuijf, H.J., Keyvan, F., Reyes, M., van Walsum, T.: Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. Springer International Publishing, Berlin (2018)
47. Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., Lanczi, L., Gerstner, E., Weber, M., Arbel, T., Avants, B.B., Ayache, N., Buendia, P., Collins, D.L., Cordier, N., Corso, J.J., Criminisi, A., Das, T., Delingette, H., Demiralp, Durst, C.R., Dojat, M., Doyle, S., Festa, J., Forbes, F., Geremia, E., Glocker, B., Golland, P., Guo, X., Hamamci, A., Iftekharuddin, K.M., Jena, R., John, N.M., Konukoglu, E., Lashkari, D., Mariz, J.A., Meier, R., Pereira, S., Precup, D., Price, S.J., Raviv, T.R., Reza, S.M.S., Ryan, M., Sarikaya, D., Schwartz, L., Shin, H., Shotton, J., Silva, C.A., Sousa, N., Subbanna, N.K., Szekely, G., Taylor, T.J., Thomas, O.M., Tustison, N.J., Unal, G., Vasseur, F., Wintermark, M., Ye, D.H., Zhao, L., Zhao, B., Zikic, D., Prastawa, M., Reyes, M., Van Leemput, K.: The multimodal brain tumor image segmentation benchmark (brats). IEEE Trans. Med. Imaging **34**(10), 1993–2024 (2015)
48. Kistler, M., Bonaretti, S., Pfahrer, M., Niklaus, R., Büchler, P.: The virtual skeleton database: An open access repository for biomedical research and collaboration. J. Med. Internet Res. **15**(11), e245 (2013)
49. Barkhof, F., Scheltens, P.: Imaging of white matter lesions. Cerebrovasc. Dis. **13**(Suppl 2), 21–30 (2002)
50. Arroyo-Camarena, S., Domínguez-Cherit, J., Lammoglia-Ordiales, L., Fabila-Bustos, D.A., Escobar-Pio, A., Stolik, S., Valor-Reed, A., de la Rosa-Vázquez, J.: Spectroscopic and imaging characteristics of pigmented non-melanoma skin cancer and melanoma in patients with skin phototypes iii and iv. Oncol. Ther. **4**(2), 315–331 (2016)

51. Codella, N.C.F., Gutman, D., Celebi, M.E., Helba, B., Marchetti, M.A., Dusza, S.W., Kalloo, A., Liopyris, K., Mishra, N.K., Kittler, H., Halpern, A.: Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium On Biomedical Imaging (ISBI), Hosted By The International Skin Imaging Collaboration (ISIC) (2017). CoRR abs/1710.05006, 1710.05006
52. Asaid, R., Boyce, G., Padmasekara, G.: Use of a smartphone for monitoring dermatological lesions compared to clinical photography. J. Mob. Technol. Med. **1**, 16–18 (2012)
53. Wu, X., Marchetti, M.A., Marghoob, A.A.: Dermoscopy: not just for dermatologists. Melanoma Manag **2**(1), 63–73 (2015)
54. Sakamoto, K.: The pathology of mycobacterium tuberculosis infection. Vet. Pathol. **49**(3), 423–39 (2012)
55. Huda, W., Abrahams, R.B.: Radiographic techniques, contrast, and noise in x-ray imaging. AJR Am. J. Roentgenol. **204**(2), W126–131 (2015)
56. Brady, A., Laoide, R., McCarthy, P., McDermott, R.: Discrepancy and error in radiology: concepts, causes and consequences. Ulster Med. J. **81**(1), 3–9 (2012)
57. Candemir, S., Jaeger, S., Palaniappan, K., P Musco, J., K Singh, R., Xue, Z., Karargyris, A., Antani, S., Thoma, G., Mcdonald, C.: Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. IEEE Trans. Med. Imaging **33**, 577–590 (2014)
58. Jaeger, S., Karargyris, A., Candemir, S., Folio, L., Siegelman, J., Callaghan, F., Xue, Z., Palaniappan, K., Singh, R.K., Antani, S., Thoma, G., Wang, Y., Lu, P., McDonald, C.J.: Automatic tuberculosis screening using chest radiographs. IEEE Trans. Med. Imaging **33**(2), 233–245 (2014)
59. Kohli, M.D., Summers, R.M., Geis, J.R.: Medical image data and datasets in the era of machine learning-whitepaper from the 2016 c-MIMI meeting dataset session. J. Digit. Imaging **30**, 392–399 (2017)
60. Valindria, V.V., Lavdas, I., Bai, W., Kamnitsas, K., Aboagye, E.O., Rockall, A.G., Rueckert, D., Glocker, B.: Reverse classification accuracy: predicting segmentation performance in the absence of ground truth. IEEE Trans. Med. Imaging **36**, 1597–1606 (2017)
61. Kouanou, A.T., Tchiotsop, D., Kengne, R., Zephirin, D.T., Armele, N.M.A., Tchinda, R.: An optimal big data workflow for biomedical image analysis. Inform. Med. Unlocked **11**, 68–74 (2018)

# Part V
# Image Modeling and Segmentation

# Chapter 12
# Color Image Segmentation Using Semi-bounded Finite Mixture Models by Incorporating Mean Templates


**Jaspreet Singh Kalsi, Muhammad Azam, and Nizar Bouguila**

**Abstract** Finite mixture models (FMM) are very popular for image segmentation. But, FMM assumes that each pixel is independent from each other. Thus, it does not consider the spatial information of the pixels which makes FMM more sensitive to noise. Generally, the traditional FMM consists of prior probability (PP) and component conditional probability (CP). In this chapter, we have incorporated mean templates, namely weighted geometric mean template (WGMT) and weighted arithmetic mean template (WAMT) to compute the CP. For estimating PP, the weighted geometric mean prior probability (WGMPP) and weighted arithmetic mean prior probability (WAMPP) templates are used. Lastly, the Expectation-Maximization (EM) algorithm is used to estimate the hyper-parameters of the FMM. Our models are proposed based on inverted Dirichlet (ID), generalized inverted Dirichlet (GID), and inverted Beta-Liouville (IBL) mixture models using the mean templates. For experimentation, the Berkeley 500 (BSD500) and MIT's Computational Visual Cognition Laboratory (CVCL) datasets are used. We have also employed eight image segmentation performance evaluation metrics such as adjusted Rand index and homogeneity score to validate the image segmentation results for the BSD500. Additionally, we have also compared the segmentation outputs for the CVCL dataset which are computed using the traditional RGB and $l_1l_2l_3$ color spaces. The results obtained from IBL mixture models (IBLMM) are more promising than ID mixture models (IDMM) and GID mixture models (GIDMM).

J. Singh Kalsi (✉) · M. Azam
Department of Electrical and Computer Engineering (ECE), Concordia University, Montreal, QC, Canada
e-mail: j_kals@encs.concordia.ca; mu_azam@encs.concordia.ca

N. Bouguila
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca



N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6_12

## 12.1 Introduction

In computer vision, image segmentation plays foundational role [1–8]. Innumerable techniques such as active contour [9–12], graph-cut-based [13–15], model-based [16–19], machine learning [20–22], and clustering-based [13, 23–29] methods have been proposed for tackling the image segmentation problem. But, none of them is universally applicable. Thus, the hunt for optimized and robust models for image segmentation is still under-process and also an open question [30, 31]. The challenges faced in image segmentation are the integration of spatial information, finding the exact number of clusters, and to segment the object smoothly without any inaccuracy specially when the image possesses noise, a complex background, low contrast, and inhomogeneous intensity. The use of FMM for image segmentation is a very popular approach in the field of computer vision [32]. The research on FMM for image representation and classification is discussed in [33–35]. A survey on the mixture expert's history can be found in [36]. The application of image segmentation using FMM ranges from the segmentation of human brain [37], automatic number plate recognition [38], content-based image retrieval [39], texture recognition [40], facial recognition [41], satellite imagery [42], etc. Image segmentation using FMM undergoes some problems. FMM-based image segmentation considers neither spatial correlation among the peer pixels nor the prior knowledge that the adjacent pixels are most likely belong to the same cluster. Also, color images are sensitive to illumination and noise [43]. To overcome these limitations, abundant techniques have been proposed to integrate spatial information. Some common approaches are markov random field (MRF) [44], hidden MRF models [45, 46], etc. But, the main drawback of using MRF models is that they are computationally expensive and require additional parameter to control the degree of image smoothness. In order to solve all the issues discussed above, we have applied mean templates for CP and PP, proposed in [44]. Furthermore, from the mean template, we can obtain four methods that are geometric conditional geometric prior (GCGP), geometric conditional arithmetic prior (GCAP), arithmetic conditional geometric prior (ACGP), and arithmetic conditional arithmetic prior (ACAP).

The remaining chapter is organized as follows. In Sect. 12.2, the general definition of traditional FMM is presented in detail. In Sect. 12.3, the challenges faced during the image segmentation using FMM are discussed along with their solutions. Section 12.4 is devoted to the mean templates for conditional probabilities where the geometric and arithmetic CP along with geometric and arithmetic PP are explained in detail. In Sect. 12.5, the integration of four methods (GCGP, GCAP, ACGP, and GCGP) with IDMM, GIDMM, and IBLMM are presented followed by their algorithms. Section 12.6 contains the experimental results in which eight segmentation evaluation metrics, the results in the form of tables for BSD500 dataset and segmentation outputs in the form of figures for both BSD500 [47] and CVCL [48] datasets are discussed. Lastly, Sect. 12.7 contains the conclusion.

## 12.2 Finite Mixture Model

Consider an image $\mathcal{X} = \{\mathbf{X_1}, \ldots, \mathbf{X_N}\}$ consisting of N pixels, where each pixel $\mathbf{X}_n$ has a dimension $D$ such that $\mathbf{X}_n = (X_{n1}, \ldots, X_{nD})$. We assume that $\mathcal{X}$ can be segmented into $M$ clusters and thus it is appropriate to use distribution as:

$$p(\mathbf{X}|\mathbf{\Theta}) = \sum_{j=1}^{M} \pi_j \, p(\mathbf{X}|\boldsymbol{\theta}_\mathbf{j}) \tag{12.1}$$

where each cluster $j$ has a weight $\pi_j$, $\sum_{j=1}^{M} \pi_j = 1$. $M$ is the number of components in the FMM. $p(\mathbf{X}|\boldsymbol{\theta}_\mathbf{j})$ is the density associated with cluster $j$, and $\mathbf{\Theta} = (\pi_1, \ldots, \pi_M, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M)$ is the set of all the mixture parameters.

## 12.3 Problem Description

Consider the binary image given in Fig. 12.1 [44]. The upper-most part of the image is white in color and has intensity value equal to 1. The lower part of the image is black in color, having intensity value equal to 0. This image is distorted by noise. Two 3 by 3 windows are extracted from upper and lower parts of the image, as shown in Fig. 12.1b and c. For binary image, let the pixels having intensity value equal to 1 and 0 be assigned to classes $\mathcal{U}$ and $\mathcal{V}$, respectively. It can be easily observed that middle pixels of both the windows are corrupted by noise and may result in mis-classification. A possible solution for this problem has two requirements: First, the spatial information of each pixel should be incorporated to prior probability $\pi_j$, therefore $\pi_j$ should be changed to $\pi_{ij}$. The $\pi_{ij}$ of the middle pixels for both the windows should be affected by the prior probability $\pi_{st}$ where $s = i \pm 1$ and $t = j \pm 1$. Each pixel inside the window should have same prior probability which can be calculated using mean prior probability (discussed in later section).

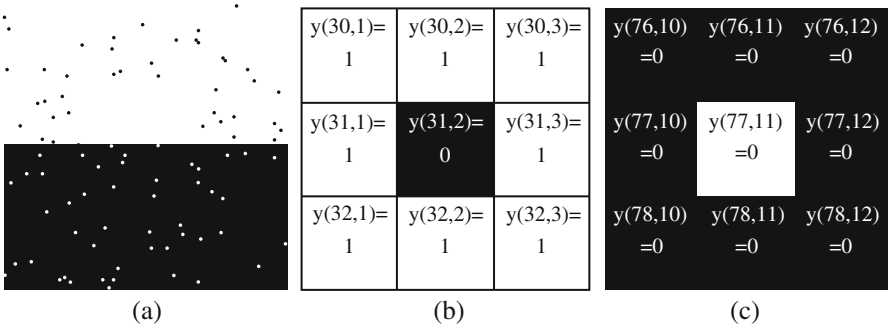



**Fig. 12.1** (**a**) Original image. (**b**) White window. (**c**) Black window. The numbers in parentheses are the coordinates of the image; 0 and 1 are the binary image intensity values

Second, given the component $j$ and intensity value $y$, the FMM satisfies the same conditional probability $p(y_n|\theta_j)$. Sometimes, this is true but not always. The $y$ of middle pixel in Fig. 12.1b is same as the pixels around the central pixel in Fig. 12.1c. These two types of pixels should belong to different clusters. Thus, the traditional FMM is not capable enough to differentiate among these types of pixels. In order to counter this issue, the authors of [44] have suggested the mean template for CP.

The authors of [44] have calculated the windows CP values (CPV) for Fig. 12.1b and c using traditional FMM, GMT, and AMT as illustrated in Fig. 12.2 [44] and Fig. 12.3 [44]. In Fig. 12.2a, the middle pixels have same CPV ($0.606/\sqrt{2\pi}$) as the pixels around the central pixel in Fig. 12.3a. Similarly, the central pixel of Fig. 12.3a and surrounding pixels of middle pixel of Fig. 12.2a have the same CPV($1/\sqrt{2\pi}$). We can observe that the proposed mean templates have removed the effect of noise from the windows shown in Figs. 12.2a and 12.3a. It is noteworthy that the model suggested by the authors in [44] is robust to noise.

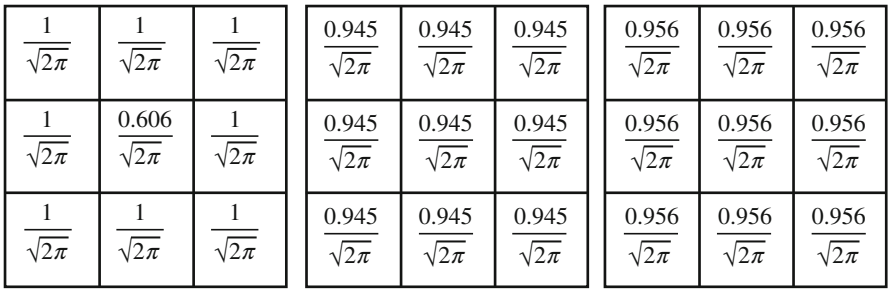



**Fig. 12.2** CPV of Fig. 12.1b, (**a**) CPV with traditional FMM. (**b**) CPV calculated by geometric template. (**c**) CPV calculated by arithmetic template

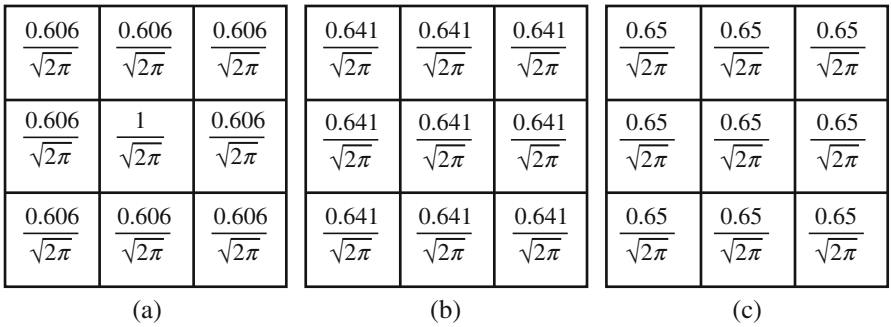



**Fig. 12.3** CPV of Fig. 12.1c, (**a**) CPV with traditional FMM. (**b**) CPV calculated by geometric template. (**c**) CPV calculated by arithmetic template

## 12.4 Mean Templates for Conditional and Prior Probabilities

In order to integrate the spatial information with the PP, Eq. (12.1) can be redefined as:

$$p(\mathbf{X}_n|\mathbf{\Theta}) = \sum_{j=1}^{M} \pi_{nj}\, p(\mathbf{X}_n|\boldsymbol{\theta}_\mathbf{j}) \tag{12.2}$$

where $\pi_{nj}$ is an updated mixing parameter where $j = 1, \ldots, M$, $\sum_{j=1}^{M} \pi_{nj} = 1$, and $n = 1, \ldots, N$. In this section, we discuss the geometric and arithmetic CP mean templates followed by their respective mixture models and complete log-likelihood equations. Furthermore, the equations of PP for both geometric and arithmetic mean templates are discussed.

### *12.4.1 Weighted Geometric Conditional Mean Template*

In this section, we are using a weighted geometric conditional mean template (WGCMT) for calculating the CP of a pixel $\mathbf{X}_n$. Thus, Eq. (12.2) can be rewritten as:

$$p(\mathbf{X}_n|\mathbf{\Theta}) = \sum_{j=1}^{M} \pi_{nj} \prod_{r\in\mathcal{N}_n} p(\mathbf{X_r}|\boldsymbol{\theta}_\mathbf{j})^{\frac{w_r}{R_n}} \tag{12.3}$$

where $\mathcal{N}_n$ is a set of peers of the $n$th pixel. The conditional probability window is (CPW) = $\{\mathcal{N}_n, \mathbf{X}_n\}$. $R_n$ is a normalized factor which is defined as

$$R_n = \sum_{r\in\mathcal{N}_n} w_r \tag{12.4}$$

In order to integrate the spatial information and pixel intensity value, the strength of $w_r$ is inversely proportional to the distance between pixels $r$ and $n$. Therefore, the authors in [44] have defined $w_r$ as a function of $d_{rn}$, which is Euclidean distance between pixels $r$ and $n$.

$$w_r = \frac{1}{\sqrt{2\pi\rho^2}} \exp\left(-\frac{d^2{}_{rn}}{2\rho^2}\right) \tag{12.5}$$

$$\rho = \frac{\text{size of CPW} - 1}{4} \tag{12.6}$$

#### 12.4.1.1 Maximum Likelihood Estimation (MLE) for WGCMT

The pixels class labels are considered as the latent variables. Each pixel $\mathbf{X}_n$ is the observed data. The membership vector is defined as $Z = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_N\}$ where $\mathbf{Z}_n = (Z_{n1}, \ldots, Z_{nM})$. If $\mathbf{X}_n$ belongs to cluster c, then $Z_{nc} = 1$ and $Z_{nl} = 0$ where $l = \{1, \ldots, M\} - \{c\}$, otherwise $Z_{nc} = 0$. The complete log-likelihood is as follows:

$$Q = \sum_n \sum_j Z_{nj} \left[ \log \pi_{nj} + \sum_{r \in \mathcal{N}_n} \frac{w_r}{R_n} \log p(\mathbf{X_r}|\boldsymbol{\theta_j}) \right] \tag{12.7}$$

EM algorithm consists of two phases: E-Step and M-Step [49]. In E-Step, the posterior probability ($\hat{Z}_{nj}$) can be calculated as:

$$\hat{Z}_{nj}^{(t+1)} = \frac{\pi_{nj}^{(t)} \prod_{r \in \mathcal{N}_n} p(\mathbf{X_r}|\boldsymbol{\theta}_\mathbf{j}^{(\mathbf{t})})^{\frac{w_r}{R_n}}}{\sum_{h=1}^{M} \pi_{nh}^{(t)} \prod_{r \in \mathcal{N}_n} p(\mathbf{X_r}|\boldsymbol{\theta}_\mathbf{h}^{(\mathbf{t})})^{\frac{w_r}{R_n}}} \tag{12.8}$$

In M-Step, we have to maximize the complete log-likelihood and solve:

$$\frac{\partial Q}{\partial \theta_j} = 0 \tag{12.9}$$

### *12.4.2 Weighted Arithmetic Conditional Mean Template*

In this part, we are using a weighted arithmetic conditional mean template (WACMT) to calculate the CP of a pixel $\mathbf{X}_n$. Thus, Eq. (12.2) can be rewritten as:

$$p(\mathbf{X}_n|\boldsymbol{\Theta}) = \sum_{j=1}^{M} \pi_{nj} \prod_{r \in \mathcal{N}_n} \frac{w_r}{R_n} p(\mathbf{X_r}|\boldsymbol{\theta_j}) \tag{12.10}$$

#### 12.4.2.1 Maximum Likelihood Estimation (MLE) for WACMT

The complete log-likelihood is as follows:

$$\begin{aligned} Q &= \sum_n \sum_j Z_{nj} \left[ \log \pi_{nj} + \log \left( \sum_{r \in \mathcal{N}_n} \frac{w_r}{R_n} p(\mathbf{X_r}|\boldsymbol{\theta_j}) \right) \right] \\ &= \sum_n \sum_j Z_{nj} \left[ \log \pi_{nj} + G \right] \end{aligned} \tag{12.11}$$

$G$ cannot be calculated directly. It is important to note that $\frac{w_r}{R_n}$ always follows the condition $\frac{w_r}{R_n} \geq 0$ and $\sum_{r\in\mathcal{N}_n} \frac{w_r}{R_n} = 1$. Therefore, we can apply the Jensen's inequality rule which is defined as, given a set of numbers $\tau \geq 0$ and $\sum_n \tau = 1$, we have $\log(\sum_n \tau x_i) \geq \sum_n \tau \log(x_i)$. Then, the G can be modified and the complete log-likelihood is:

$$Q = \sum_n \sum_j Z_{nj} \left[ \log \pi_{nj} + \sum_{r\in\mathcal{N}_n} \frac{w_r}{R_n} \log p(\mathbf{X_r}|\boldsymbol{\theta_j}) \right] \tag{12.12}$$

In E-Step, $\hat{Z}_{nj}$ can be calculated as

$$\hat{Z}_{nj}^{(t+1)} = \frac{\pi_{nj}^{(t)} \prod_{r\in\mathcal{N}_n} \frac{w_r}{R_n} p(\mathbf{X_r}|\boldsymbol{\theta}_\mathbf{j}^{(\mathbf{t})})}{\sum_{h=1}^{M} \pi_{nh}^{(t)} \prod_{r\in\mathcal{N}_n} \frac{w_r}{R_n} p(\mathbf{X_r}|\boldsymbol{\theta}_\mathbf{h}^{(\mathbf{t})})} \tag{12.13}$$

The M-Step can be computed using Eq. (12.9).

### *12.4.3 Weighted Prior Probability Estimation*

The prior probability for FMM is as follows:

$$\pi_j = \frac{\sum_{n=1}^{N} Z_{nj}}{\sum_{n=1}^{N} \sum_{j=1}^{M} Z_{nj}} \tag{12.14}$$

According to the authors of [44], the weighted geometric prior mean template (WGPMT) is given as:

$$\pi_{nj}^{(t+1)} = \frac{\pi_{nj}^{(t)} \prod_{r\in\rho_n} Z_{nj}^{\frac{w_r}{R_n}}}{\sum_{h=1}^{M} \pi_{nh}^{(t)} \prod_{r\in\rho_n} Z_{nj}^{\frac{w_r}{R_n}}} \tag{12.15}$$

The weighted arithmetic prior mean template (WAPMT) is defined by:

$$\pi_{nj}^{(t+1)} = \frac{\pi_{nj}^{(t)} \prod_{r\in\rho_n} \frac{w_r}{R_n} Z_{nj}}{\sum_{h=1}^{M} \pi_{nh}^{(t)} \prod_{r\in\rho_n} \frac{w_r}{R_n} Z_{nj}} \tag{12.16}$$

## 12.5 Integration of Mean Templates with IDMM, GIDMM, and IBLMM

In this section, we propose our mixture models based on three probability density functions including ID, GID, IBL with incorporation of WGCMT and WACMT.

### *12.5.1 Incorporation of Mean Template with IDMM*

In this subsection, we explain the integration of mean template with IDMM.

#### 12.5.1.1 The Probability Density Function of ID

If $\mathbf{X}$ is a positive vector which consists of $D$ dimensions and following an ID distribution, then it has a joint density function given as follows [50]:

$$p(\mathbf{X}|\boldsymbol{\alpha}) = \frac{\Gamma(|\boldsymbol{\alpha}|)}{\prod_{d=1}^{D}\Gamma(\alpha_d)}\prod_{d=1}^{D+1} X_d^{\alpha_d-1}\left(1+|\mathbf{X}|\right)^{-|\boldsymbol{\alpha}|} \tag{12.17}$$

where $|\mathbf{X}| = \sum_{d=1}^{D} X_d$, each $X_d > 0$. The parameter of ID is $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{D+1}]$, $|\boldsymbol{\alpha}| = \sum_{d=1}^{D+1}\alpha_d$, $\alpha_d > 0$ where $d = 1, \ldots, D+1$. The mean and the variance of ID are given as follows:

$$E(X_d) = \frac{\alpha_d}{\alpha_{D+1}-1} \tag{12.18}$$

$$Var(X_d) = \frac{\alpha_d(\alpha_d+\alpha_{D+1}-1)}{(\alpha_{D+1}-1)^2(\alpha_{D+1}-2)} \tag{12.19}$$

#### 12.5.1.2 Incorporation of IDMM with WGCMT

By substituting Eq. (12.17) into Eq. (12.7), the complete log-likelihood is as follows:

$$Q = \sum_{n}\sum_{j} Z_{nj}\Bigg[\log \pi_{nj} + \sum_{r\in\mathcal{N}_n}\frac{w_r}{R_n}\log\left(\frac{\Gamma(|\boldsymbol{\alpha}|)}{\prod_{d=1}^{D}\Gamma(\alpha_d)}\prod_{d=1}^{D+1} X_d^{\alpha_d-1}\left(1+|\mathbf{X}|\right)^{-|\boldsymbol{\alpha}|}\right)\Bigg] \tag{12.20}$$

In E-Step, $\hat{Z}_{nj}$ can be calculated using Eq. (12.8). In M-step, we need to maximize the complete log-likelihood. From Eq. (12.9), the partial derivative of $Q$ with respect to $\alpha_{jd}$ where $j = 1, \ldots, M$ and $d = 1, \ldots, D$ is as follows:

$$\frac{\partial Q}{\partial \alpha_{jd}} = \sum_{n=1}^{N} \sum_{j=1}^{M} \hat{Z}_{nj} \left\{ \Psi|\boldsymbol{\alpha}_j| - \Psi|\alpha_{jd}| + \sum_{r \in \mathcal{N}_n} \frac{w_r}{R_n} \log \left( \frac{X_{rd}}{1 + |\mathbf{X}_r|} \right) \right\} \tag{12.21}$$

where $\Psi(.)$ is the digamma function.

The partial derivative of $Q$ with respect to $\alpha_{jD+1}$ is as follows:

$$\frac{\partial Q}{\partial \alpha_{jD+1}} = \sum_{n=1}^{N} \sum_{j=1}^{M} \hat{Z}_{nj} \left\{ \Psi|\boldsymbol{\alpha}_j| - \Psi|\alpha_{jd+1}| + \sum_{r \in \mathcal{N}_n} \frac{w_r}{R_n} \log \left( \frac{X_{rd}}{1 + |\mathbf{X}_r|} \right) \right\} \tag{12.22}$$

Considering Eqs. (12.21) and (12.22), it can be observed that no closed solution exists for $\boldsymbol{\alpha}_j$. Therefore, we have used Newton–Raphson method as follows:

$$\boldsymbol{\alpha}_j^{(k+1)} = \boldsymbol{\alpha_j}^{(k)} - \mathbf{G}_j H_j^{-1} \tag{12.23}$$

where $\boldsymbol{\alpha}_j^{(k+1)}$ is the updated hyper-parameter, $\boldsymbol{\alpha_j}^{(k)}$ is the old hyper-parameter, $\mathbf{G}_j$ is the gradient followed by $H_j^{-1}$, which is the inverse of Hessian matrix. The gradient is the first partial order derivative of $Q$ and described as follows:

$$\mathbf{G}_j = \left( \frac{\partial Q}{\partial \alpha_{j1}}, \ldots, \frac{\partial Q}{\partial \alpha_{jD+1}} \right) \tag{12.24}$$

To find the Hessian of Q, we have to calculate the second and mixed derivatives:

$$\frac{\partial^2 Q}{\partial^2 \alpha_{jd}} = \sum_{n=1}^{N} \hat{Z}_{nj} \Big( \Psi'(|\boldsymbol{\alpha_j}|) - \Psi'(\alpha_{jd}) \Big), \quad d = 1, \ldots, D+1 \tag{12.25}$$

$$\frac{\partial^2 Q}{\partial^2 \alpha_{jd_1} \alpha_{jd_2}} = \Psi'(|\boldsymbol{\alpha_j}|) \sum_{n=1}^{N} \hat{Z}_{nj}, \quad d1 \neq d2, \quad d_1, d_2 = 1, \ldots, D+1 \tag{12.26}$$

where $\Psi'(.)$ is the trigamma function. The Hessian can be described as:

$$H_j = \sum_{n=1}^{N} \hat{Z}_{nj} \begin{bmatrix} \Psi'(|\boldsymbol{\alpha_j}|) - \Psi'(\alpha_{j1}) & \Psi'(|\boldsymbol{\alpha_j}|) & \ldots & \Psi'(|\boldsymbol{\alpha_j}|) \\ \Psi'(|\boldsymbol{\alpha_j}|) & \Psi'(|\boldsymbol{\alpha_j}|) - \Psi'(\alpha_{j2}) & \ldots & \Psi'(|\boldsymbol{\alpha_j}|) \\ \vdots & \ldots & \ddots & \vdots \\ \Psi'(|\boldsymbol{\alpha_j}|) & \ldots & & \Psi'(|\boldsymbol{\alpha_j}|) - \Psi'(\alpha_{jD+1}) \end{bmatrix} \tag{12.27}$$

Thus, $H_j$ can be written as:

$$H_j = D_j + \rho_j {A_j}^T A_j \tag{12.28}$$

where $D_j$ is a diagonal matrix and described by:

$$D_j = \text{diag}\left( -\sum_{n=1}^{N} \hat{Z}_{nj} \Psi'(\alpha_{j1}), \ldots, -\sum_{n=1}^{N} \hat{Z}_{nj} \Psi'(\alpha_{jD+1}) \right) \tag{12.29}$$

The constant $\rho_j$ is defined as:

$$\rho_j = \left[ \left( (\Psi'(|\boldsymbol{\alpha_j}|) \sum_{d=1}^{D+1} \frac{1}{\Psi'(\alpha_{jd})} \right) - 1 \right] \Psi'(|\boldsymbol{\alpha_j}|) \sum_{n=1}^{N} \hat{Z}_{nj} \tag{12.30}$$

$A_j^T = (a_1, \ldots, a_{D+1}), a_d = 1$ where $d = 1, \ldots, D+1$. In order to find $H_j^{-1}$, a matrix inverse theorem given in [51] can be used [52]:

$$H_j^{-1} = D_j^{-1} + {\rho^*}_j A_j^{*T} A_j^* \tag{12.31}$$

$D_j^{-1}$ can be easily computed. ${A_j}^*$ and ${\rho^*}_j$ are expressed by two following equations:

$${A_j}^* = \frac{-1}{\sum_{n=1}^{N} \hat{Z}_{nj}} \left[ \frac{1}{\Psi'(\alpha_{j1})}, \ldots, \frac{1}{\Psi'(\alpha_{j,D+1})} \right] \tag{12.32}$$

$${\rho^*}_j = \Psi'(|\boldsymbol{\alpha_j}|) \sum_{n=1}^{N} \hat{Z}_{nj} \left[ \Psi'(|\boldsymbol{\alpha_j}|) \sum_{n=1}^{N} \frac{1}{\Psi'(\alpha_{jd})} - 1 \right] \tag{12.33}$$

#### 12.5.1.3 Incorporation of IDMM with WACMT

In E-Step, the $\hat{Z}_{nj}$ can be calculated using Eq. (12.13). The M-Step can be calculating using Eq. (12.23).

#### 12.5.1.4 IDMM's Algorithm

We have two conditional probabilities that are WGCMT, WACMT and two prior probabilities that are WGPMT, WAPMT. Therefore, we can have four models as following:

1. GCGP: the application of weighted geometric conditional mean template to weighted geometric prior mean template.
2. GCAP: the application of weighted geometric conditional mean template to weighted arithmetic prior mean template.
3. ACGP: the application of weighted arithmetic conditional mean template to weighted geometric prior mean template.
4. ACAP: the application of weighted arithmetic conditional mean template to weighted arithmetic prior mean template.

The algorithm is as follows:

1. INPUT: An image $\mathcal{X}$ and $M$.
2. Apply K-means clustering algorithm to group pixels into $M$ clusters.
3. Apply method of moments to calculate the initial value for $\boldsymbol{\alpha}$ parameter.
4. E-Step:
   - Calculate $\boldsymbol{\pi}$ using Eq. (12.15) for GCGP and ACGP, and using Eq. (12.16) for GCAP and ACAP.
   - Calculate $\hat{Z}_{nj}$ using Eq. (12.8) for GCGP and GCAP, and using Eq. (12.13) for ACGP and ACAP.
5. M-Step: Calculate the updated value of $\boldsymbol{\alpha}$ parameter for each cluster $j$, using Eq. (12.23).
6. Iterate through E-Step and M-Step until convergence.

## *12.5.2 Mean Template Incorporation with GIDMM*

The second mixture model is GIDMM and we are integrating the mean template with it.

### 12.5.2.1 The Probability Density Function of GID

If $\mathbf{X}$ is a positive vector which consists of $D$ dimensions and following a GID, then its joint density function is given by:

$$p(\mathbf{X}|\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j) = \prod_{d=1}^{D} \frac{\Gamma(\alpha_{jd} + \beta_{jd})}{\Gamma(\alpha_{jd})\Gamma(\beta_{jd})} T_{nd}^{\alpha_{jd}-1} \left(1 + |\mathbf{X}|\right)^{-\Omega_{jd}} \tag{12.34}$$

where $\boldsymbol{\alpha_j} = [\alpha_{j1}, \ldots, \alpha_{jD}]$, $\boldsymbol{\beta_j} = [\beta_{j1}, \ldots, \beta_{jD}]$. $|\mathbf{X}| = \sum_{d=1}^{D} X_d$. We define $\Omega$ such that $\Omega_{jd} = \alpha_{jd} + \beta_{jd} - \beta_{jd+1}$ for $d = 0, \ldots, D$ with $\beta_{jD+1} = 0$. The GID possesses a property which makes its estimation simple. If there exists a vector $\mathbf{X}$ that follows GID, then we can create another vector $\mathbf{W}_n = [W_{n1}, \ldots, W_{nD}]$ where elements follow inverted beta (IB) distributions via the following transformation:

$$f(X_{nd}) = \begin{cases} X_{nd} & d = 1 \\ \frac{X_{nd}}{1 - X_{n1} - , \ldots, - X_{nd-1}} & d = 2, \ldots, D \end{cases} \tag{12.35}$$

The pdf of IB is defined as:

$$p_{i\text{Beta}}(W_{nd}|\alpha_{jd}, \beta_{jd}) = \frac{\Gamma(\alpha_{jd} + \beta_{jd})}{\Gamma(\alpha_{jd})\Gamma(\beta_{jd})} W_{nd}^{\alpha_{jd}-1}(1 + W_{jd})^{-(\alpha_{jd}+\beta_{jd})} \tag{12.36}$$

The mean of inverted beta (IB) is given by:

$$E(W_d) = \frac{\alpha_d}{\beta_d - 1} \tag{12.37}$$

The variance of IB is as follows:

$$\text{Var}(W_d) = \frac{\alpha_d(\alpha_d + \beta_d - 1)}{(\beta_d - 2)(\beta_d - 1)^2} \tag{12.38}$$

#### 12.5.2.2 Incorporation of GIDMM with WGCMT

By substituting Eq. (12.36) into Eq. (12.7), the complete log-likelihood is given by:

$$Q = \sum_{n=1}^{N}\sum_{j=1}^{M} Z_{nj}\Bigg[\log \pi_{nj} + \sum_{r \in \mathcal{N}_n} \frac{w_r}{R_n} \log\left(\frac{\Gamma(\alpha_{jd} + \beta_{jd})}{\Gamma(\alpha_{jd})\Gamma(\beta_{jd})} W_{nd}^{\alpha_{jd}-1}(1 + W_{jd})^{-(\alpha_{jd}+\beta_{jd})}\right)\Bigg] \tag{12.39}$$

In E-Step, the $\hat{Z}_{nj}$ can be calculated using Eq. (12.6), and in M-Step, the partial derivatives of $Q$ with respect to $\alpha_{jd}$ and $\beta_{jd}$ are as follows:

$$\frac{\partial Q}{\partial \alpha_{jd}} = \sum_{i=1}^{N} \hat{Z}_{nj}\left\{\Psi(\alpha_{jd} + \beta_{jd}) - \Psi(\alpha_{jd}) + \log\left(\frac{W_{nd}}{1 + W_{nd}}\right)\right\} \tag{12.40}$$

$$\frac{\partial Q}{\partial \beta_{jd}} = \sum_{i=1}^{N} \hat{Z}_{nj}\left\{\Psi(\alpha_{jd} + \beta_{jd}) - \Psi(\beta_{jd}) + \log\left(\frac{W_{nd}}{1 + W_{nd}}\right)\right\} \tag{12.41}$$

From Eqs. (12.40) and (12.41), it can be observed that no closed-form solution exists for $\boldsymbol{\theta_{jd}}$. Therefore, we have to use Newton–Raphson method as follows:

$$\boldsymbol{\theta_{jd}}^{(k+1)} = \boldsymbol{\theta_{jd}}^{(k)} - H_{jd}^{-1} G_{jd} \tag{12.42}$$

where $H_{jd}$ is the Hessian matrix [53] and given as:

$$H_{jd} = \begin{bmatrix} \frac{\partial^2 Q}{\partial^2 \alpha_{jd}} & \frac{\partial^2 Q}{\partial^2 \alpha_{jd} \beta_{jd}} \\ \frac{\partial^2 Q}{\partial^2 \alpha_{jd} \beta_{jd}} & \frac{\partial^2 Q}{\partial^2 \beta_{jd}} \end{bmatrix} \tag{12.43}$$

The second and mixed derivatives of $Q$ are as follows:

$$\frac{\partial^2 Q}{\partial^2 \alpha_{jd}} = \sum_{n=1}^{N} \hat{Z}_{nj} \Big( \Psi'(\alpha_{jd} + \beta_{jd}) - \Psi'(\alpha_{jd}) \Big), \quad d = 1, \ldots, D+1 \tag{12.44}$$

$$\frac{\partial^2 Q}{\partial^2 \beta_{jd}} = \sum_{n=1}^{N} \hat{Z}_{nj} \Big( \Psi'(\alpha_{jd} + \beta_{jd}) - \Psi'(\beta_{jd}) \Big) \tag{12.45}$$

$$\frac{\partial^2 Q}{\partial^2 \alpha_{jd} \beta_{jd}} = \Psi'(\alpha_{jd} + \beta_{jd}) \sum_{n=1}^{N} \hat{Z}_{nj} \tag{12.46}$$

$\mathbf{G_{jd}}$ is defined as follows:

$$\mathbf{G_{jd}} = \left( \frac{\partial Q}{\partial \alpha_{jd}}, \frac{\partial Q}{\partial \beta_{jd}} \right) \tag{12.47}$$

#### 12.5.2.3 Incorporation of GIDMM with WACMT

In E-Step, the $\hat{Z}_{nj}$ can be calculated using Eq. (12.13). The M-Step is based on Eq. (12.42).

#### 12.5.2.4 GIDMM's Algorithm

In this section, we propose an algorithm for the four models that are GCGP, GCAP, ACGP, and ACAP.

1. INPUT: An image $\mathcal{X}$ and $M$.
2. Apply K-means clustering algorithm to group pixels into $M$ clusters.
3. Apply method of moments to calculate the initial value for $\boldsymbol{\alpha}$ parameter.
4. E-Step:
   - Calculate $\boldsymbol{\pi}$ using Eq. (12.15) for GCGP and ACGP, and using Eq. (12.16) for GCAP and ACAP.

- Calculate $\hat{Z}_{nj}$ using Eq. (12.8) for GCGP and GCAP, and Eq. (12.13) for ACGP and ACAP.

5. M-Step: Calculate the updated value of $\theta$ parameter for each cluster $j$, using Eq. (12.42).
6. Iterate through E-Step and M-Step until convergence.

### *12.5.3 Incorporation of Mean Template with IBLMM*

In this subsection, we integrate the mean template with IBLMM.

#### 12.5.3.1 The Probability Density Function of IBL

If $\mathbf{X}$ is a positive vector which consists of $D$ dimensions and following an IBL distribution, then it has a joint density function which is given in [54] as:

$$p(\mathbf{X}|\alpha_1 \ldots \alpha_d, \alpha, \beta, \lambda) = \frac{\Gamma(|\boldsymbol{\alpha}|)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \times \prod_{d=1}^{D+1} \frac{X_d^{\alpha_d-1}}{\Gamma(\alpha_d)} \lambda^{\beta} \Big(|\mathbf{X}|\Big)^{\alpha-\sum_{d=1}^{D}\alpha_d} \Big(\lambda+|\mathbf{X}|\Big)^{-(\alpha+\beta)} \tag{12.48}$$

where $|\mathbf{X}| = \sum_{d=1}^{D} X_d$, each $X_d > 0$, $\alpha > 0$, $\beta > 0$, and $\lambda > 0$. The mean and the variance of IBL are given by:

$$E(X_d) = \frac{\alpha_d}{\alpha_{D+1} - 1}; \tag{12.49}$$

$$\text{Var}(X_d) = \frac{\alpha_d(\alpha_d + \alpha_{D+1} - 1)}{(\alpha_{D+1} - 1)^2(\alpha_{D+1} - 2)} \tag{12.50}$$

#### 12.5.3.2 Incorporation of IBLMM with WGCMT

By substituting Eq. (12.48) into Eq. (12.7), the complete log-likelihood is as follows:

$$Q = \sum_n \sum_j Z_{nj} \Bigg[ \log \pi_{nj} + \sum_{r \in \mathcal{N}_n} \frac{w_r}{R_n} \log \Bigg( \frac{\Gamma(|\boldsymbol{\alpha}|)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \prod_{d=1}^{D+1} \frac{X_d^{\alpha_d-1}}{\Gamma(\alpha_d)}$$

$$\times \lambda^{\beta}\left(|\mathbf{X}|\right)^{\alpha-\sum_{d=1}^{D}\alpha_d}\left(\lambda+|\mathbf{X}|\right)^{-(\alpha+\beta)}\Bigg)\Bigg] \tag{12.51}$$

In E-Step, $\hat{Z}_{nj}$ can be calculated using Eq. (12.8). In M-step, we need to calculate the partial derivative of $Q$ with respect to the parameters of IBL [55].

The partial derivative of $Q$ with respect to $\alpha_j$ where $j = 1, \ldots, M$ is as follows:

$$\frac{\partial Q}{\partial \alpha_j} = \sum_{i=1}^{N} \hat{Z}_{nj}\left\{\log \sum_{d=1}^{D} X_{nd} - \log\left(\lambda_j + \sum_{d=1}^{D} X_{nd}\right) + \Psi(\alpha_j+\beta_j) - \Psi(\alpha_j)\right\} \tag{12.52}$$

The partial derivative of $Q$ with respect to $\beta_j$ is given by:

$$\frac{\partial Q}{\partial \beta_j} = \sum_{i=1}^{N} \hat{Z}_{nj}\left\{\log \lambda_j - \log\left(\lambda_j + \sum_{d=1}^{D} X_{nd}\right) + \Psi(\alpha_j+\beta_j) - \Psi(\beta_j)\right\} \tag{12.53}$$

The partial derivative of $Q$ with respect to $\alpha_{jd}$ is defined as:

$$\frac{\partial Q}{\partial \alpha_{jd}} = \sum_{i=1}^{N} \hat{Z}_{nj}\left\{\log X_{nd} - \log \sum_{d=1}^{D} X_{nd} + \Psi\left(\sum_{d=1}^{D} \alpha_{jd}\right) - \Psi(\alpha_{jd})\right\} \tag{12.54}$$

The partial derivative of $Q$ with respect to $\lambda_j$ is expressed as follows:

$$\frac{\partial Q}{\partial \lambda_j} = \sum_{i=1}^{N} \hat{Z}_{nj}\left\{\frac{\beta_j}{\lambda_j} - \frac{\alpha_j+\beta_j}{\lambda_j + \sum_{d=1}^{D} X_{nd}}\right\} \tag{12.55}$$

From Eq. (12.52) to Eq. (12.55), it can be observed that a closed-form solution does not exist for $\boldsymbol{\theta_j}$.

In order to estimate these parameters, the Newton–Raphson method can be used.

$$\boldsymbol{\theta_j}^{(k+1)} = \boldsymbol{\theta_j}^{(k)} - H_j^{-1} G_j \tag{12.56}$$

#### 12.5.3.3 Incorporation of IBMM with WACMT

In E-Step, the $\hat{Z}_{nj}$ can be calculated using Eq. (12.13). The M-Step is performed via Eq. (12.56).

#### 12.5.3.4 IBLMM's Algorithm

In this section, we propose an algorithm for the four models, namely GCGP, GCAP, ACGP, and ACAP.

1. INPUT: An image $\mathcal{X}$ and $M$.
2. Apply K-means clustering algorithm to group pixels into $M$ clusters.
3. Apply method of moments to calculate the initial value of $\boldsymbol{\theta}$ parameters.
4. E-Step:
   - Calculate $\boldsymbol{\pi}$ using Eq. (12.15) for GCGP and ACGP, and using Eq. (12.16) for GCAP and ACAP.
   - Calculate $\hat{Z}_{nj}$ using Eq. (12.8) for GCGP and GCAP, and using Eq. (12.13) for ACGP and ACAP.
5. M-Step: Calculate the updated value of $\boldsymbol{\theta}$ parameter for each cluster $j$, using Eq. (12.56).
6. Iterate through E-Step and M-Step until convergence.

## 12.6 Experimental Results

To investigate the performance of our proposed framework, we test the models on two different datasets that are BSD500 and CSCV. The BSD500 dataset is known as a reliable source to compare different image segmentation algorithms, contains 500 color images, and has at least five ground-truth segments for each image. The CSCV dataset is composed of many categories such as Coast and Beach, Highway, etc. Each category contains few hundred images. All the images are in color, in jpeg format, and are $256 \times 256$ pixels. Their sources vary from digital cameras, websites, and commercial databases. This section is composed of two experiments. In first one, we tested the proposed models on BSD500 and evaluated the results using segmentation evaluation metrics. In second experiment, we employed CSCV dataset and compared our models using two color spaces that are rgb and $l_1 l_2 l_3$, which are explained below.

### *12.6.1 Metrics for Segmentation Performance Evaluation*

In order to compare the performances of the proposed algorithms, we have used eight image segmentation evaluation metrics. From BSD500, we have ground truth labels (actual labels) of each image and also have class labels, predicted from the proposed algorithms (predicted labels). The performance evaluation metrics are as follows:

#### 12.6.1.1 Adjusted Rand Index (ARI)

It is defined as the level of similarity among the actual labels and predicted labels. In ARI, the permutations are not considered. The ARI's value will tend to 0, if the predicted labels are arranged randomly. ARI has a range of $[-1, 1]$, and values closer to zero are considered as bad clustering and values closer to 1 means good clustering. The ARI [56] is given as:

$$\text{ARI} = \frac{\text{RI} - E[\text{RI}]}{\max(\text{RI}) - E[\text{RI}]} \tag{12.57}$$

where $E[\text{RI}]$ is expected value of RI (Rand index). The RI is defined as:

$$\text{ARI} = \frac{a+b}{C_2^{n_{\text{samples}}}} \tag{12.58}$$

where $K$ and $C$ are the actual and predicted labels, respectively. The $a$ is the number of element pairs having same class labels in $K$ and $C$. $b$ is the number of element pairs having different class labels in $K$ and $C$.

#### 12.6.1.2 Adjusted Mutual Information Score (AMIS)

The mutual information (MI) is defined as the level of agreement of actual labels and predicted labels, without permutation. The AMIS [57–59] is the adjusted version of MI and is defined as:

$$\text{AMIS} = \frac{\text{MI}(T, U) - E[\text{MI}(T, U)]}{\text{mean}(H(T), H(U)) - E[\text{MI}(T, U)]} \tag{12.59}$$

where $T$ and $U$ are two class labels assignments, $H(T)$ and $H(U)$ define the entropy for $T$ and $U$, respectively. $E[\text{MI}(T, U)]$ is the expected value of $\text{MI}(T, U)$.

#### 12.6.1.3 Normalized Mutual Information Score (NMIS)

The NMIS [57–59] is the normalized version of MI and given as:

$$\text{NMIS}(T, U) = \frac{\text{MI}(T, U)}{\text{mean}\big(H(T), H(U)\big)} \tag{12.60}$$

#### 12.6.1.4 Homogeneity Score (HS)

HS [60, 61] uses a criteria related to analysis of the conditional entropy. HS is defined as each cluster contains only members of the single class. It has a range of [0, 1], where 1 means each cluster only contains members of just one class. On the other hand, 0 means that almost every data inside a cluster contains different class labels. It is given by:

$$\text{HS} = 1 - \frac{H(C|K)}{H(C)} \tag{12.61}$$

where $H(C|K)$ is the conditional entropy of the classes given the cluster assignments:

$$H(C|K) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{n_{ck}}{n}\log\left(\frac{n_{ck}}{n_k}\right); \qquad H(C) = -\sum_{c=1}^{|C|}\frac{n_c}{n}.\log\left(\frac{n_c}{n}\right) \tag{12.62}$$

where $H(C)$ is the entropy of the classes, $n$ is the number of pixels in the image, $n_c$ is the number of pixels that belong to class $c$, and $n_k$ is the number of pixels that belong to cluster $k$.

#### 12.6.1.5 Completeness Score (CS)

CS [60, 61] is also a criteria related to analysis of the conditional entropy. CS is defined as all the members of a given class that belong to the same clusters. It has a range of [0, 1], where 0 means worst clustering and 1 means perfect clustering.

$$\text{CS} = 1 - \frac{H(K|C)}{H(K)} \tag{12.63}$$

where $H(K|C)$ and $H(K)$ can be computed in a symmetric manner.

#### 12.6.1.6 V-Measure Score (VMS)

The VMS [60, 61] is defined as the harmonic mean of HS and CS. It is symmetrical in nature. It is as follows:

$$\text{VMS} = 2 \times \left(\frac{\text{HS} \times \text{CS}}{\text{HS} + \text{CS}}\right) \tag{12.64}$$

#### 12.6.1.7 Calinski-Harabaz Index (CHI)

It is one of the most flexible metrics for image segmentation. If the human segmentations are not available, then CHI can be used for model evaluation, where the higher value signifies that the clusters are well defined. For $M$ clusters, the CHI [62] is defined as the ratio of the between-clusters dispersion mean and the within-cluster dispersion:

$$\text{CHI}(M) = \frac{T_r(B_M)}{T_r(W_M)} \times \frac{n-M}{M-1} \tag{12.65}$$

where $W_M$ is the within-cluster dispersion matrix and $B_M$ is defined as the between-group dispersion matrix:

$$W_M = \sum_{m=1}^{M} \sum_{x \in C_m} (x - C_m)(x - C_m)^T; \quad B_M = \sum_{m} n_m (c_m - c)(c_m - c)^T \tag{12.66}$$

$C_m$: set of pixels in cluster $m$, $c_m$: the center of the cluster $m$, $n_m$: number of pixels in cluster $m$.

#### 12.6.1.8 Jaccard Similarity Score (JSS)

JSS [63–65] is also called Jaccard index. It is defined as the ratio of intersection (of predicted labels and actual labels) and the union (of predicted labels and actual labels). It also ranges between 0 and 1 where 0 means very bad score and 1 means the segmentation output is perfect.

### *12.6.2 Color Spaces for Image Segmentation*

The selection of color space is crucial for image segmentation. It is desirable to have a color space robust against varying illumination. Few color spaces are assessed, dissected, and examined in [66]. Out of many color spaces, we have selected the rgb and $l_1 l_2 l_3$ color spaces which are as follows:

$$r(\mathcal{R}, \mathcal{G}, \mathcal{B}) = \frac{\mathcal{R}}{\mathcal{R} + \mathcal{G} + \mathcal{B}} \tag{12.67}$$

$$g(\mathcal{R}, \mathcal{G}, \mathcal{B}) = \frac{\mathcal{G}}{\mathcal{R} + \mathcal{G} + \mathcal{B}} \tag{12.68}$$

$$b(\mathcal{R},\mathcal{G},\mathcal{B}) = \frac{\mathcal{B}}{\mathcal{R}+\mathcal{G}+\mathcal{B}} \tag{12.69}$$

$$l_1(\mathcal{R},\mathcal{G},\mathcal{B}) = \frac{(\mathcal{R}-\mathcal{G})^2}{\text{SUM(RGB)}} \tag{12.70}$$

$$l_2(\mathcal{R},\mathcal{G},\mathcal{B}) = \frac{(\mathcal{R}-\mathcal{B})^2}{\text{SUM(RGB)}} \tag{12.71}$$

$$l_3(\mathcal{R},\mathcal{G},\mathcal{B}) = \frac{(\mathcal{G}-\mathcal{B})^2}{\text{SUM(RGB)}} \tag{12.72}$$

where SUM(RGB) = $(\mathcal{R}-\mathcal{G})^2 + (\mathcal{R}-\mathcal{B})^2 + (\mathcal{G}-\mathcal{B})^2$. The $l_1l_2l_3$ and rgb outperform the traditional $\mathcal{RGB}$ color space and hence used in our experiments also.

### *12.6.3 Experiment 1*

Here, we present some results of testing our models on images from the BSD500 using $l_1l_2l_3$ color space. Figure 12.4 contains the segmentation outputs of image 29030 (obtained by using ID, GID, and IBL versions of GCGP, GCAP, ACGP, and ACAP models). Considering the second, third, and fourth columns of Fig. 12.4, it can be observed that IBL is able to detect car as an object properly, followed by GID and ID. Tables 12.1 and 12.2 verify our visual analysis by means of qualitative approach using the image segmentation evaluation metrics. Images in BSD500 may contain upto six human segmentations. The output of each model is compared with each human segmentation by using evaluation metrics. By incorporating 3 pdfs with 4 mean template models, we have 12 models. Each result of these models is compared with 6 human segments which led to have upto 72 comparisons. In order to reduce the complexity in understanding, we have compared the mean of each model with the human segmentations.

Figure 12.5 contains the segmentation outputs of image 118035. Again, from the second, third, and fourth columns of Fig. 12.5, it can be observed that IBLMM is able to detect the different components of a building much accurately as compared to GIDMM and IDMM. Tables 12.3 and 12.4 contain the qualitative analysis of outputs from image 118035.

Similarly, Fig. 12.6 contains the segmentation results of 124084. From the second, third, and fourth columns of Fig. 12.6, it can be seen that IBLMM is able to detect the flower petals pretty smoothly as compared to its competitors. Tables 12.5 and 12.6 contain the qualitative analysis of outputs from image 124084.

Figure 12.7 contains the segmentation results of image 376086. From the second, third, and fourth columns of Fig. 12.7, it can be seen that IBLMM is able to detect

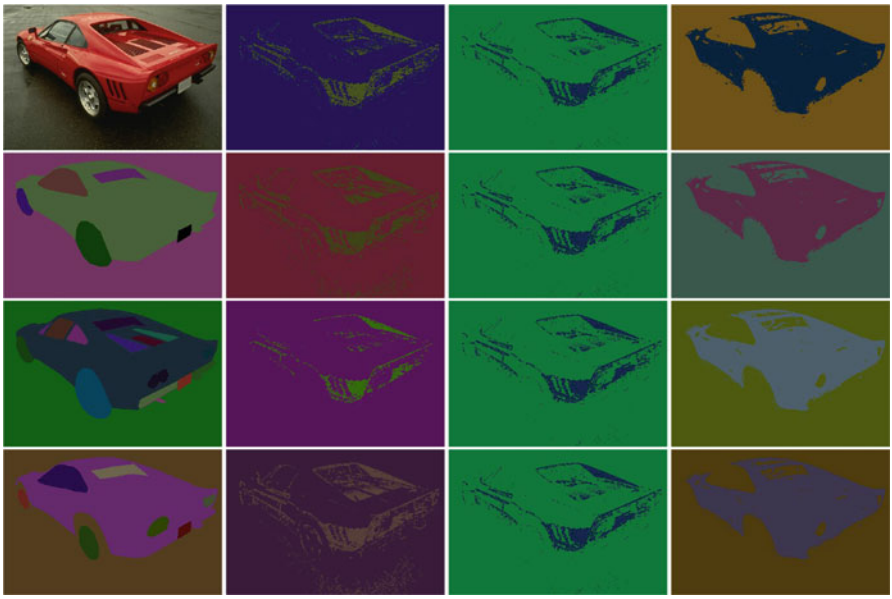


**Fig. 12.4** Column 1: Contains the original image (29030) followed by the three ground-truths. Column 2: Contains the segmentation outputs from ID version of GCGP, GCAP, ACGP, and ACAP models, Column 3: Contains the segmentation outputs from GID version of GCGP, GCAP, ACGP, and ACAP models, Column 4: Contains the segmentation outputs from IBL version of GCGP, GCAP, ACGP, and ACAP models

**Table 12.1** Performance evaluation of the 29030 image with the ARI, AMIS, NMIS, MIS and HS metrics

| Algorithm | Model | K | ARI's mean | AMIS's mean | NMIS's mean | MIS's mean | HS's mean |
|---|---|---|---|---|---|---|---|
| ID | GCGP | 2 | 0.092 | 0.063 | 0.135 | 0.078 | 0.063 |
| | GCAP | 2 | 0.111 | 0.052 | 0.099 | 0.065 | 0.052 |
| | ACGP | 2 | 0.042 | 0.039 | 0.104 | 0.048 | 0.039 |
| | ACAP | 2 | 0.112 | 0.061 | 0.119 | 0.075 | 0.061 |
| | Mean | | **0.089** | **0.054** | **0.114** | **0.067** | **0.054** |
| GID | GCGP | 2 | 0.084 | 0.059 | 0.129 | 0.073 | 0.059 |
| | GCAP | 2 | 0.081 | 0.057 | 0.126 | 0.071 | 0.057 |
| | ACGP | 2 | 0.079 | 0.056 | 0.124 | 0.069 | 0.056 |
| | ACAP | 2 | 0.077 | 0.055 | 0.123 | 0.068 | 0.055 |
| | Mean | | **0.080** | **0.057** | **0.126** | **0.070** | **0.057** |
| IBL | GCGP | 2 | 0.662 | 0.420 | 0.587 | 0.520 | 0.420 |
| | GCAP | 2 | 0.664 | 0.419 | 0.585 | 0.519 | 0.419 |
| | ACGP | 2 | 0.650 | 0.412 | 0.578 | 0.511 | 0.412 |
| | ACAP | 2 | 0.651 | 0.412 | 0.578 | 0.510 | 0.412 |
| | Mean | | **0.6569** | **0.4158** | **0.5821** | **0.5151** | **0.4158** |

**Table 12.2** Quality analysis of the 29030 image with the CS, VM, JSS, and CHI metrics

| Algorithm | Model | K | CS's mean | VM's mean | JSS's mean | CHI |
|---|---|---|---|---|---|---|
| ID | GCGP | 2 | 0.289 | 0.103 | 0.001 | 73962.307 |
| | GCAP | 2 | 0.190 | 0.081 | 0.013 | 54791.895 |
| | ACGP | 2 | 0.281 | 0.068 | 0.000 | 42512.321 |
| | ACAP | 2 | 0.236 | 0.096 | 0.007 | 64281.498 |
| | Mean | | **0.249** | **0.087** | **0.005** | **58887.006** |
| GID | GCGP | 2 | 0.285 | 0.098 | 0.001 | 71021.648 |
| | GCAP | 2 | 0.282 | 0.095 | 0.001 | 67895.016 |
| | ACGP | 2 | 0.279 | 0.092 | 0.001 | 65785.996 |
| | ACAP | 2 | 0.277 | 0.091 | 0.001 | 64006.640 |
| | Mean | | **0.281** | **0.094** | **0.001** | **268709.300** |
| IBL | GCGP | 2 | 0.826 | 0.553 | 0.003 | 262224.285 |
| | GCAP | 2 | 0.823 | 0.552 | 0.004 | 259172.527 |
| | ACGP | 2 | 0.817 | 0.545 | 0.003 | 265561.038 |
| | ACAP | 2 | 0.815 | 0.544 | 0.003 | 264630.709 |
| | Mean | | **0.8202** | **0.5487** | **0.0031** | **262897.140** |

**Fig. 12.5** Column 1: Contains the original image (118035) followed by the three ground-truths. Column 2: Contains the segmentation outputs from ID's version of GCGP, GCAP, ACGP, and ACAP models, Column 3: Contains the segmentation outputs from GID's version of GCGP, GCAP, ACGP, and ACAP models, Column 4: Contains the segmentation outputs from IBL's version of GCGP, GCAP, ACGP, and ACAP models

**Table 12.3** Performance evaluation of the 118035 image with the ARI, AMIS, NMIS, MIS and HS metrics

| Algorithm | Model | K | ARI's mean | AMIS's mean | NMIS's mean | MIS's mean | HS's mean |
|---|---|---|---|---|---|---|---|
| ID | GCGP | 2 | 0.157 | 0.174 | 0.372 | 0.258 | 0.174 |
| | GCAP | 2 | 0.158 | 0.174 | 0.371 | 0.258 | 0.174 |
| | ACGP | 2 | 0.171 | 0.174 | 0.351 | 0.260 | 0.174 |
| | ACAP | 2 | 0.318 | 0.223 | 0.345 | 0.334 | 0.223 |
| | Mean | | **0.201** | **0.186** | **0.360** | **0.278** | **0.186** |
| GID | GCGP | 2 | 0.179 | 0.143 | 0.249 | 0.217 | 0.143 |
| | GCAP | 2 | 0.159 | 0.130 | 0.232 | 0.198 | 0.130 |
| | ACGP | 2 | 0.162 | 0.133 | 0.236 | 0.201 | 0.133 |
| | ACAP | 2 | 0.165 | 0.134 | 0.237 | 0.204 | 0.134 |
| | Mean | | **0.166** | **0.135** | **0.238** | **0.205** | **0.135** |
| IBL | GCGP | 2 | 0.589 | 0.438 | 0.631 | 0.644 | 0.438 |
| | GCAP | 2 | 0.589 | 0.438 | 0.632 | 0.644 | 0.438 |
| | ACGP | 2 | 0.589 | 0.438 | 0.631 | 0.644 | 0.438 |
| | ACAP | 2 | 0.588 | 0.436 | 0.629 | 0.641 | 0.436 |
| | Mean | | **0.5887** | **0.4376** | **0.6308** | **0.6432** | **0.4376** |

**Table 12.4** Quality analysis of the 118035 image with the CS, VM, JSS, and CHI metrics

| Algorithm | Model | K | CS's mean | VM's mean | JSS's mean | CHI |
|---|---|---|---|---|---|---|
| ID | GCGP | 2 | 0.810 | 0.283 | 0.453 | 184910.025 |
| | GCAP | 2 | 0.810 | 0.283 | 0.453 | 184682.459 |
| | ACGP | 2 | 0.722 | 0.277 | 0.452 | 163659.995 |
| | ACAP | 2 | 0.543 | 0.311 | 0.452 | 68588.861 |
| | Mean | | **0.721** | **0.288** | **0.452** | **150460.335** |
| ID | GCGP | 2 | 0.810 | 0.283 | 0.453 | 184910.025 |
| | GCAP | 2 | 0.810 | 0.283 | 0.453 | 184682.459 |
| | ACGP | 2 | 0.722 | 0.277 | 0.452 | 163659.995 |
| | ACAP | 2 | 0.543 | 0.311 | 0.452 | 68588.861 |
| | Mean | | **0.721** | **0.288** | **0.452** | **150460.335** |
| IBL | GCGP | 2 | 0.927 | 0.586 | 0.452 | 133485.637 |
| | GCAP | 2 | 0.928 | 0.586 | 0.452 | 133528.359 |
| | ACGP | 2 | 0.927 | 0.585 | 0.452 | 133460.767 |
| | ACAP | 2 | 0.923 | 0.583 | 0.452 | 133191.467 |
| | Mean | | **0.9262** | **0.5850** | **0.4521** | **133416.557** |

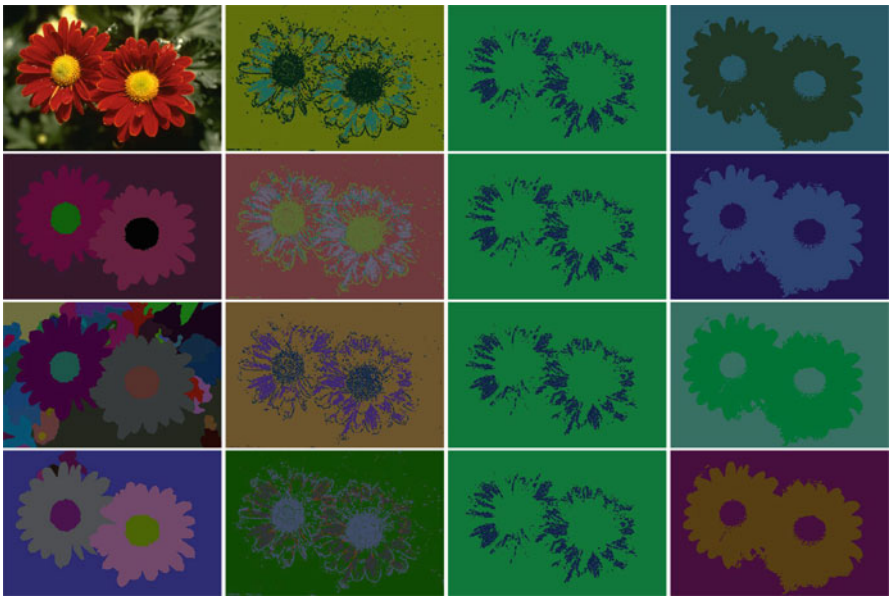


**Fig. 12.6** Column 1: Contains the original image (124084.jpg) followed by the three ground-truths. Column 2: Contains the segmentation outputs from ID's version of GCGP, GCAP, ACGP, and ACAP models, Column 3: Contains the segmentation outputs from GID's version of GCGP, GCAP, ACGP, and ACAP models, Column 4: Contains the segmentation outputs from IBL's version of GCGP, GCAP, ACGP, and ACAP models

the two men more accurately as compared to GIDMM and IDMM. Tables 12.7 and 12.8 contain the image segmentation results for image 376086.

### *12.6.4 Experiment 2*

Considering Figs. 12.8 and 12.9, the first image is the original one (n291030), followed by eight outputs, out of which the first four outputs are computed using rgb color space and the remaining four outputs are obtained by using $l_1l_2l_3$ color space.

Also, Figs. 12.10 and 12.11 contain the segmentation output of images n291030 and art255, respectively. In Figs. 12.10 and 12.11, the first image is the original one, followed by eight outputs, out of which in the first four image, rgb color space is used and for rest segmentation outputs, $l_1l_2l_3$ color space is used.

Similarly, Figs. 12.12 and 12.13 contain the segmentation output of image n291030 and art255, respectively. In Figs. 12.12 and 12.13, the first image is the original one, followed by eight outputs, out of which the first four segmentation output, rgb color space is used and for remaining four outputs, $l_1l_2l_3$ color space is used.

**Table 12.5** Performance evaluation of the 124084 image with the ARI, AMIS, NMIS, MIS and HS metrics

| Algorithm | Model | K | ARI's mean | AMIS's mean | NMIS's mean | MIS's mean | HS's mean |
|---|---|---|---|---|---|---|---|
| ID | GCGP | 2 | 0.046 | 0.030 | 0.084 | 0.044 | 0.030 |
| | GCAP | 2 | 0.134 | 0.085 | 0.162 | 0.126 | 0.086 |
| | ACGP | 2 | 0.053 | 0.035 | 0.092 | 0.052 | 0.035 |
| | ACAP | 2 | 0.065 | 0.042 | 0.104 | 0.062 | 0.042 |
| | Mean | | **0.075** | **0.048** | **0.110** | **0.071** | **0.048** |
| GID | GCGP | 2 | 0.112 | 0.071 | 0.150 | 0.104 | 0.071 |
| | GCAP | 2 | 0.112 | 0.071 | 0.150 | 0.104 | 0.071 |
| | ACGP | 2 | 0.112 | 0.071 | 0.150 | 0.104 | 0.071 |
| | ACAP | 2 | 0.112 | 0.071 | 0.150 | 0.104 | 0.071 |
| | Mean | | **0.112** | **0.071** | **0.150** | **0.104** | **0.071** |
| IBL | GCGP | 2 | 0.502 | 0.354 | 0.511 | 0.519 | 0.354 |
| | GCAP | 2 | 0.500 | 0.352 | 0.508 | 0.516 | 0.352 |
| | ACGP | 2 | 0.500 | 0.353 | 0.510 | 0.517 | 0.353 |
| | ACAP | 2 | 0.501 | 0.353 | 0.510 | 0.517 | 0.353 |
| | Mean | | **0.5007** | **0.3529** | **0.5097** | **0.5171** | **0.3529** |

**Table 12.6** Quality analysis of the 124084 image with the CS, VM, JSS, and CHI metrics

| Algorithm | Model | K | CS's mean | VM's mean | JSS's mean | CHI |
|---|---|---|---|---|---|---|
| ID | GCGP | 2 | 0.239 | 0.053 | 0.454 | 5717.493 |
| | GCAP | 2 | 0.313 | 0.132 | 0.454 | 15553.412 |
| | ACGP | 2 | 0.249 | 0.061 | 0.454 | 6769.251 |
| | ACAP | 2 | 0.264 | 0.071 | 0.454 | 8221.494 |
| | Mean | | **0.266** | **0.079** | **0.454** | **9065.413** |
| GID | GCGP | 2 | 0.327 | 0.115 | 0.000 | 47850.229 |
| | GCAP | 2 | 0.326 | 0.115 | 0.000 | 47616.519 |
| | ACGP | 2 | 0.326 | 0.115 | 0.000 | 47471.541 |
| | ACAP | 2 | 0.326 | 0.114 | 0.000 | 47390.939 |
| | Mean | | **0.326** | **0.115** | **0.000** | **190329.228** |
| IBL | GCGP | 2 | 0.759 | 0.472 | 0.427 | 592730.642 |
| | GCAP | 2 | 0.754 | 0.469 | 0.427 | 589902.293 |
| | ACGP | 2 | 0.756 | 0.471 | 0.427 | 587628.857 |
| | ACAP | 2 | 0.756 | 0.471 | 0.427 | 591670.571 |
| | Mean | | **0.7562** | **0.4707** | **0.4268** | **590483.091** |

**Fig. 12.7** Column 1: Contains the original image (376086) followed by three ground-truths. Column 2: Contains the segmentation outputs from ID's version of GCGP, GCAP, ACGP, and ACAP models, Column 3: Contains the segmentation outputs from GID's version of GCGP, GCAP, ACGP, and ACAP models, Column 4: Contains the segmentation outputs from IBL's version of GCGP, GCAP, ACGP, and ACAP models

**Table 12.7** Performance evaluation of the 376086 image with the ARI, AMIS, NMIS, MIS and HS metrics

| Algorithm | Model | K | ARI's mean | AMIS's mean | NMIS's mean | MIS's mean | HS's mean |
|---|---|---|---|---|---|---|---|
| ID | GCGP | 2 | -0.004 | 0.047 | 0.109 | 0.097 | 0.047 |
| | GCAP | 2 | -0.003 | 0.040 | 0.091 | 0.084 | 0.041 |
| | ACGP | 2 | -0.006 | 0.023 | 0.071 | 0.049 | 0.023 |
| | ACAP | 2 | -0.004 | 0.047 | 0.112 | 0.098 | 0.047 |
| | Mean | | **−0.004** | **0.039** | **0.096** | **0.082** | **0.039** |
| GID | GCGP | 2 | -0.004 | 0.046 | 0.107 | 0.095 | 0.046 |
| | GCAP | 2 | -0.004 | 0.047 | 0.110 | 0.097 | 0.047 |
| | ACGP | 2 | -0.004 | 0.048 | 0.112 | 0.098 | 0.048 |
| | ACAP | 2 | -0.004 | 0.048 | 0.112 | 0.098 | 0.048 |
| | Mean | | **−0.004** | **0.047** | **0.110** | **0.097** | **0.047** |
| IBL | GCGP | 2 | 0.064 | 0.103 | 0.180 | 0.208 | 0.103 |
| | GCAP | 2 | 0.061 | 0.102 | 0.178 | 0.206 | 0.102 |
| | ACGP | 2 | 0.062 | 0.102 | 0.178 | 0.206 | 0.102 |
| | ACAP | 2 | 0.064 | 0.103 | 0.179 | 0.208 | 0.103 |
| | Mean | | **0.0627** | **0.1027** | **0.1788** | **0.2070** | **0.1028** |

**Table 12.8** Quality analysis of the 376086 image with the CS, VM, JSS, and CHI metrics

| Algorithm | Model | K | CS's mean | VM's mean | JSS's mean | CHI |
|---|---|---|---|---|---|---|
| D | GCGP | 2 | 0.267 | 0.078 | 0.000 | 197432.441 |
| | GCAP | 2 | 0.212 | 0.066 | 0.003 | 165708.431 |
| | ACGP | 2 | 0.231 | 0.041 | 0.000 | 71589.989 |
| | ACAP | 2 | 0.274 | 0.079 | 0.000 | 197429.295 |
| | Mean | | **0.249** | **0.066** | **0.001** | **158040.039** |
| GID | GCGP | 2 | 0.260 | 0.076 | 0.001 | 195016.547 |
| | GCAP | 2 | 0.268 | 0.078 | 0.000 | 198092.848 |
| | ACGP | 2 | 0.273 | 0.079 | 0.000 | 198893.445 |
| | ACAP | 2 | 0.273 | 0.079 | 0.000 | 198893.445 |
| | Mean | | **0.269** | **0.078** | **0.000** | **790896.285** |
| IBL | GCGP | 2 | 0.324 | 0.153 | 0.136 | 54345.372 |
| | GCAP | 2 | 0.322 | 0.151 | 0.136 | 54980.749 |
| | ACGP | 2 | 0.322 | 0.151 | 0.136 | 56306.326 |
| | ACAP | 2 | 0.324 | 0.152 | 0.136 | 54602.056 |
| | Mean | | **0.3229** | **0.1516** | **0.1359** | **55058.626** |

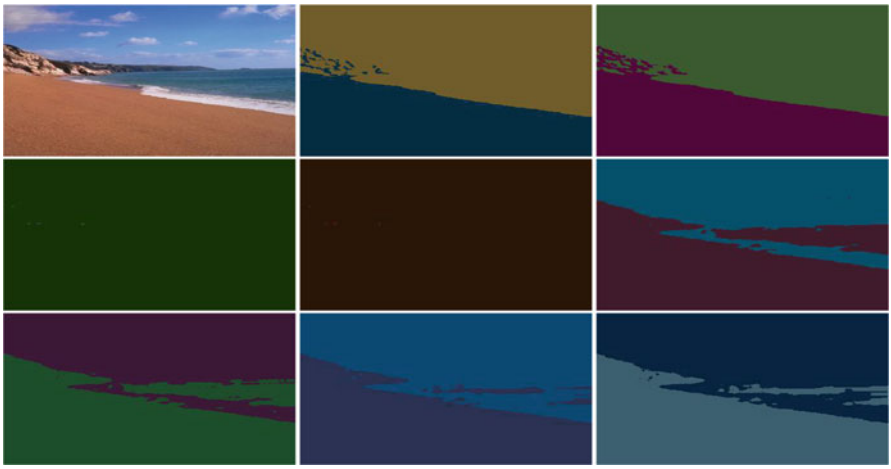


**Fig. 12.8** Original image (n291030) followed by the eight output images from ID's version of the GCGP, GCAP, ACGP, and ACAP models, out of which the first four images, have used rgb color space and remaining four have used $l_1l_2l_3$ color space

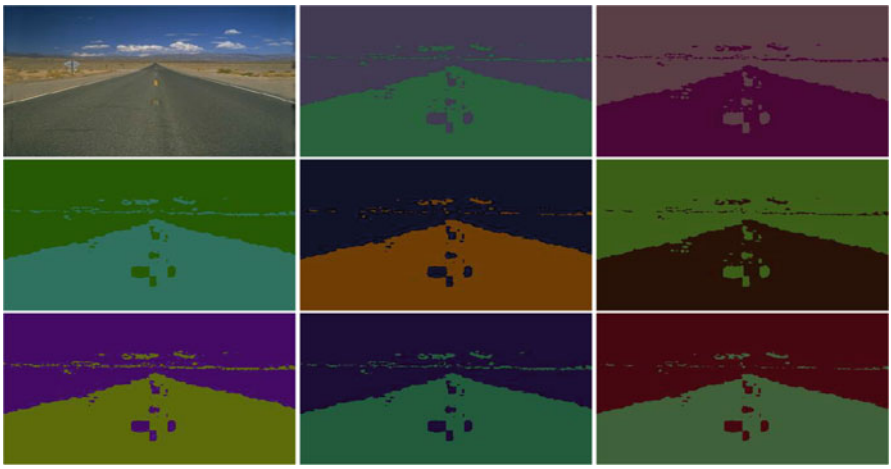


**Fig. 12.9** Contains the original image (art255) followed by the eight output images from ID's version of the GCGP, GCAP, ACGP, and ACAP models. Out of which the first four images, have used rgb color space and remaining four have used $l_1l_2l_3$ color space

**Fig. 12.10** Original image (n291030) followed by the eight output images from GID's version of the GCGP, GCAP, ACGP, and ACAP models out of which the first four images, have used rgb color space and remaining four have used $l_1l_2l_3$ color space

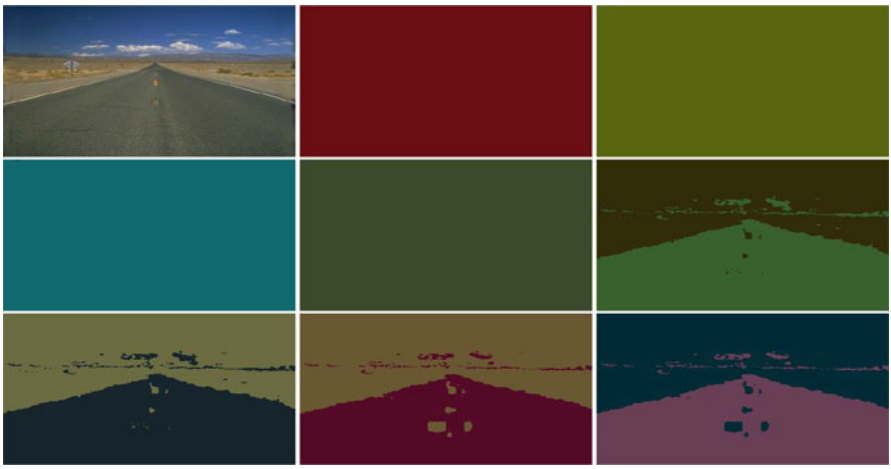


**Fig. 12.11** Original image (art255) followed by the eight output images from GID's version of the GCGP, GCAP, ACGP, and ACAP models, out of which the first four images, have used rgb color space and remaining four have used $l_1 l_2 l_3$ color space

**Fig. 12.12** Original image (n291030) followed by the eight output images from IBL's version of the GCGP, GCAP, ACGP, and ACAP models, out of which the first four images, have used rgb color space and remaining four have used $l_1 l_2 l_3$ color space

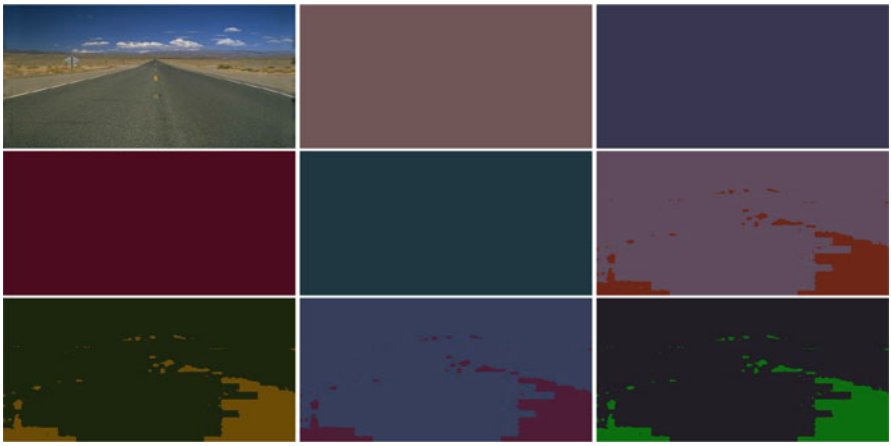


**Fig. 12.13** Original image (art255) followed by the eight output images from IBL's version of the GCGP, GCAP, ACGP, and ACAP models, out of which the first four images, have used rgb color space and remaining four have used $l_1l_2l_3$ color space

## 12.7 Conclusion

The main aim of this chapter is to develop sophisticated algorithms for image segmentation. We have used an approach proposed [44] in which the authors have suggested the incorporation of traditional FMM with CP and PP mean templates. These methods ensure the integration of spatial information by using peer pixels information and thus makes the FMM more robust to noise. We explained how the mean templates integrate spatial information by introducing the pixel's weight in mixture model estimation. We have implemented the IDMM, GIDMM, and IBLMM versions of GCGP, GCAP, ACGP, and ACAP models. These semi-bounded FMM are chosen precisely because of their flexibility that allow to describe many shapes. We have used BSD500 and CVCL dataset for experimentation. It has been found that out of the proposed algorithms IBLMM outperformed the GIDMM and IDMM. Also, the $l_1l_2l_3$ color space is far better than the rgb and the traditional RGB color space. Future works could be devoted to the application of the proposed models and approaches for object detection and tracking as well as video segmentation.

## References

1. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Comput. Vis. **22**(1), 61–79 (1997)
2. Fabijańska, A., Gocławski, J.: The segmentation of 3D images using the random walking technique on a randomly created image adjacency graph. IEEE Trans. Image Process. **24**(2), 524–537 (2015)

3. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. Int. J. Comput. Vis. **1**(4), 321–331 (1988)
4. Li, C., Kao, C., Gore, J.C., et al.: Implicit active contours driven by local binary fitting energy. In: Anonymous 2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Piscataway, pp. 1–7 (2007)
5. Li, C., Xu, C., Gui, C., et al.: Level set evolution without re-initialization: a new variational formulation. In: Anonymous 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 430–436. IEEE, Piscataway (2005)
6. Ren, Z.: Adaptive active contour model driven by fractional order fitting energy. Signal Process. **117**, 138–150 (2015)
7. Wang, X., Shan, J., Niu, Y., et al.: Enhanced distance regularization for re-initialization free level set evolution with application to image segmentation. Neurocomputing **141**, 223–235 (2014)
8. Wu, J., Zhao, Y., Zhu, J., et al.: Milcut: a sweeping line multiple instance learning paradigm for interactive image segmentation. In: Anonymous Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 256–263 (2014)
9. Zhang, K., Zhang, L., Song, H., Zhou, W.: Active contours with selective local or global segmentation: a new formulation and level set method. Image Vis. Comput. **28**, 668–676 (2010)
10. Han, B., Wu, Y.: A novel active contour model based on modified symmetric cross entropy for remote sensing river image segmentation. Pattern Recogn. **67**, 396–409 (2017)
11. Chen, Y.-T.: A novel approach to segmentation and measurement of medical image using level set methods. Magn. Reson. Imaging **39**, 175–193 (2017)
12. Feng, C., Zhao, D., Huang, M.: Image segmentation and bias correction using local inhomogeneous intensity clustering (LINC): a region-based level set method. Neurocomputing **219**, 107–129 (2017)
13. Liu, G., Zhang, Y., Wang, A.: Incorporating adaptive local information into fuzzy clustering for image segmentation. IEEE Trans. Image Process. **24**, 3990–4000 (2015)
14. Li, G., Chen, X., Shi, F., et al.: Automatic liver segmentation based on shape constraints and deformable graph cut in CT images. IEEE Trans. Image Process. **24**, 5315–5329 (2015)
15. Dai, S., Lu, K., Dong, J., et al.: A novel approach of lung segmentation on chest CT images using graph cuts. Neurocomputing **168**, 799–807 (2015)
16. Ji, Z., Xia, Y., Sun, Q., et al.: Fuzzy local Gaussian mixture model for brain MR image segmentation. IEEE Trans. Inf. Technol. Biomed. **16**, 339–347 (2012)
17. Boudaren, M.E.Y., An, L., Pieczynski, W.: Unsupervised segmentation of SAR images using Gaussian mixture-hidden evidential Markov fields. IEEE Geosci. Remote Sens. Lett. **13**, 1865–1869 (2016)
18. Xia, Y., Ji, Z., Zhang, Y.: Brain MRI image segmentation based on learning local variational Gaussian mixture models. Neurocomputing **204**, 189–197 (2016)
19. Nguyen, T.M., Wu, Q.M.J.: Fast and robust spatially constrained Gaussian mixture model for image segmentation. IEEE Trans. Circuits Syst. Video Technol. **23**, 621–635 (2013)
20. Orlando, J.I., Prokofyeva, E., Blaschko, M.B.: A discriminatively trained fully connected conditional random field model for blood vessel segmentation in fundus images. IEEE Trans. Biomed. Eng. **64**, 16–27 (2017)
21. Salehi, S.S.M., Erdogmus, D., Gholipour, A.: Auto-context convolutional neural network (Auto-Net) for brain extraction in magnetic resonance imaging. IEEE Trans. Med. Imaging **36**, 2319–2330 (2017)
22. Duan, Y., Liu, F., Jiao, L., et al.: SAR image segmentation based on convolutional-wavelet neural network and Markov random field. Pattern Recogn. **64**, 255–267 (2017)
23. Sefidpour, A., Bouguila, N.: Spatial color image segmentation based on finite non-Gaussian mixture models. Expert Syst. Appl. **39**(10), 8993–9001 (2012)
24. Bdiri, T., Bouguila, N., Ziou, D.: Variational Bayesian inference for infinite generalized inverted Dirichlet mixtures with feature selection and its application to clustering. Appl. Intell. **44**(3), 507–525 (2016)

25. Bdiri, T., Bouguila, N., Ziou, D.: Visual scenes categorization using a flexible hierarchical mixture model supporting users ontology. In: Anonymous 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, pp. 262–267 (2013)
26. Al Mashrgy, M., Bdiri, T., Bouguila, N.: Robust simultaneous positive data clustering and unsupervised feature selection using generalized inverted Dirichlet mixture models. Knowl. Based Syst. **59**, 182–195 (2014)
27. Bdiri, T., Bouguila, N., Ziou, D.: A statistical framework for online learning using adjustable model selection criteria. Eng. Appl. Artif. Intell. **49**, 19–42 (2016)
28. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
29. Alush, A., Goldberger, J.: Hierarchical image segmentation using correlation clustering. IEEE Trans. Neural Netw. Learn. Syst. **27**, 1358–1367 (2016)
30. Shi, J., Malik, J.: Normalized cuts and image segmentation. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2000)
31. Pal, N.R., Pal, S.K.: A review on image segmentation techniques. Pattern Recogn. **26**, 1277–1294 (1993)
32. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley, New York (2000)
33. Bouguila, N.: Spatial color image databases summarization. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, vol. 1, pp. I-953–I-956. IEEE, Piscataway (2007)
34. Bouguila, N.: Count data modeling and classification using finite mixtures of distributions. IEEE Trans. Neural Netw. **22**(2), 186–198 (2011)
35. Fan, W., Bouguila, N., Ziou, D.: Variational learning for finite Dirichlet mixture models and applications. IEEE Trans. Neural Netw. Learn. Syst. **23**(5), 762–774 (2012)
36. Yuksel, S.E., Wilson, J.N., Gader, P.D.: Twenty years of mixture of experts. IEEE Trans. Neural Netw. Learn. Syst. **23**(8), 1177–1193 (2012)
37. Zhang, Y., Brady, M., Smith, S.: Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE Trans. Med. Imaging **20**(1), 45–57 (2001)
38. Panahi, R., Gholampour, I.: Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications. IEEE Trans. Intell. Transp. Syst. **18**(4), 767–779 (2017)
39. Cheng, M., Mitra, N.J., Huang, X., et al.: Global contrast based salient region detection. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 569–582 (2015)
40. Chen, L., Papandreou, G., Kokkinos, I., et al.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2018)
41. Hershkovitch, T., Riklin-Raviv, T.: Model-dependent uncertainty estimation of medical image segmentation. In: Anonymous 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), pp. 1373–1376 (2018)
42. Price, S.R., Price, S.R., Price, C.D., et al.: Pre-screener for automatic detection of road damage in SAR imagery via advanced image processing techniques. In: Anonymous Pattern Recognition and Tracking XXIX, vol. 10649, pp. 1064913. International Society for Optics and Photonics, Bellingham (2018)
43. Sanjay-Gopal, S., Hebert, T.J.: Bayesian pixel classification using spatially variant finite mixtures and the generalized EM algorithm. IEEE Trans. Image Process. **7**(7), 1014–1028 (1998)
44. Zhang, H., Wu, Q.J., Nguyen, T.M.: Incorporating mean template into finite mixture model for image segmentation. IEEE Trans. Neural Netw. Learn. Syst. **24**(2), 328–335 (2013)
45. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
46. Zhang, Y., Brady, M., Smith, S.: Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE Trans. Med. Imaging **20**(1), 45–57 (2001)

47. Martin, D., Fowlkes, C., Tal, D., et al.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Anonymous. ICCV, Vancouver (2001)
48. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput. Vis. **42**(3), 145–175 (2001)
49. Moon, T.K.: The expectation-maximization algorithm. IEEE Signal Process. Mag. **13**(6), 47–60 (1996)
50. Tiao, G.G., Cuttman, I.: The inverted Dirichlet distribution with applications. J. Am. Stat. Assoc. **60**(311), 793–805 (1965)
51. Graybill, F.A.: Matrices with Applications in Statistics. Wadsworth, Belmont (1983)
52. Bdiri, T., Bouguila, N.: Positive vectors clustering using inverted Dirichlet finite mixture models. Expert Syst. Appl. **39**(2), 1869–1882 (2012)
53. Bdiri, T., Bouguila, N., Ziou, D.: A statistical framework for online learning using adjustable model selection criteria. Eng. Appl. Artif. Intell. **49**, 19–42 (2016)
54. Fang, K.: Symmetric multivariate and related distributions. Chapman and Hall/CRC, New York (1990)
55. Hu, C., Fan, W., Du, J., et al.: A novel statistical approach for clustering positive data based on finite inverted Beta-Liouville mixture models. Neurocomputing **333**, 110–123 (2019)
56. Strehl, A., Joydeep, G.: Cluster ensembles – a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)
57. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison. In: Proceedings of the 26th Annual International Conference on Machine Learning – ICML (2009)
58. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J. Mach. Learn. Res. **11**, 2837 (2010)
59. Yang, Z., Algesheimer, R., Tessone, C.J.: A comparative analysis of community detection algorithms on artificial networks. Sci. Rep. **6**, 30750 (2016)
60. Rosenberg, A., Hirschberg, J.: V-measure: a conditional entropy-based external cluster evaluation measure. In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 410–420 (2007)
61. Becker, H.: Identification and Characterization of Events in Social Media, PhD Thesis (2011)
62. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Commun. Stat. Theory Methods **3**, 1–27 (1974)
63. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, Reading (2005)
64. Jaccard, P.: Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull. Soc. Vaud. Sci. Nat. **37**, 547–579 (1901)
65. Jaccard, P.: The distribution of the flora in the alpine zone. New Phytol. **11**, 37–50 (1912)
66. Gevers, T., Smeulders, A.W.: Color-based object recognition. Pattern Recogn. **32**(3), 453–464 (1999)

# Chapter 13
# Medical Image Segmentation Based on Spatially Constrained Inverted Beta-Liouville Mixture Models


**Wenmin Chen, Wentao Fan, Nizar Bouguila, and Bineng Zhong**

**Abstract** In this chapter, we propose an image segmentation method based on a spatially constrained inverted Beta-Liouville (IBL) mixture model for segmenting medical images. Our method adopts the IBL distribution as the basic distribution, which can demonstrate better performance than commonly used distributions (such as Gaussian distribution) in image segmentation. To improve the robustness of our image segmentation method against noise, the spatial relationship among nearby pixels is imposed into our model by using generalized means. We develop a variational Bayes inference algorithm to learn the proposed model, such that model parameters can be efficiently estimated in closed form. In our experiments, we use both simulated and real brain magnetic resonance imaging (MRI) data to validate our model.

## 13.1 Introduction

Image segmentation is a fundamental task in image analysis and has been applied to various fields include medical imaging, face recognition, and pedestrian detection [1–3]. The purpose of image segmentation is to divide an image into several different regions according to the characteristics of regions within that image. Usually the effect of segmentation is related to the noise level, the sharpness, the brightness, the shadows, the illumination in the images. These factors increase the difficulty of image segmentation and sometimes may result in poor segmentation. With the upgrade of medical imaging equipment, image segmentation has been widely used in medical image analysis [4, 5]. Through medical image segmentation, the

W. Chen · W. Fan (✉) · B. Zhong
Department of Computer Science and Technology, Huaqiao University, Xiamen, China
e-mail: 1611414003@hqu.edu.cn; fwt@hqu.edu.cn; bnzhong@hqu.edu.cn

N. Bouguila
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca



N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6_13

efficiency of doctor diagnosis can be significantly improved. Thus, in this chapter we focus on developing an efficient medical image segmentation method through spatially constrained mixture models.

Many image segmentation methods have been previously proposed, such as edge-based methods [6, 7], region-based methods [8–10], graph-based methods [11–13], cluster-based methods [14, 15], and so on. Among the existing methods, the model-based segmentation methods, especially finite mixture models, have attracted more and more attention. Finite mixture model is composed of linear combinations of a finite number of basic distributions. It is a powerful tool in clustering analysis and has demonstrated its effectiveness in image segmentation. Nevertheless, image segmentation methods based on conventional finite mixture models are very sensitive to noise since they do not take the prior knowledge that neighboring pixels most probably belong to the same segment into account. In order to include the spatial dependency between pixels into mixture models, several works based on spatially constrained finite Gaussian mixture models have been successfully developed for image segmentation [16–19]. In these methods, since the spatial dependence of pixels in an image is considered, they are more robust against noise than conventional mixture models. However, one disadvantage of these spatially constrained mixture models is that they were learnt using the expectation maximization (EM) algorithm, which may be significantly affected by the initial values of parameters and can easily converge to a local maximum with an inappropriate initialization. Moreover, as shown in several recent works, mixture models based on non-Gaussian distributions (such as Dirichlet [20, 21], inverted Dirichlet [22, 23], generalized inverted Dirichlet [24–27], or Beta-Liouville distributions [28, 29]) may provide better clustering performance than those methods based on Gaussian mixture models, particulary in image segmentation [30–32].

In this work, we propose an image segmentation method based on spatially constrained non-Gaussian mixture models. Our mixture model is constructed by considering inverted Beta-Liouville (IBL) as the basic distribution. The motivation of choosing the IBL distribution to build our mixture model for image segmentation is that it contains inverted Dirichlet distribution as a special case and therefore can provide more flexibility [33]. Also, compared to Gaussian that can only approximate symmetric distributions, IBL allows both symmetric and asymmetric distributions. In addition, we add the spatial relationship between nearby pixels in our model by using generalized means (GM) [34]. Thus, the prior knowledge that neighboring pixels most probably belong to the same segment is taken into account in our model. In order to learn the proposed model for image segmentation, we develop a learning method based on variational Bayes (VB) with mean-field assumption [35–37], such that model parameters can be effectively estimated in closed form. The effectiveness of the proposed image segmentation method is validated though experiments with both simulated and real MRI brain images.

The remaining part of this chapter can be listed as follows. In Sect. 13.2, we introduce the spatially constrained IBL mixture model. In Sect. 13.3, we develop a learning algorithm based on variational Bayes to estimate the parameters of our

model. In Sect. 13.4, we provide experimental results of our model in segmenting the simulated and real MRI brain images. Finally, conclusion is given in Sect. 13.5.

## 13.2 The Spatially Constrained Inverted Beta-Liouville Mixture Model

### *13.2.1 Finite IBL Mixture Model*

Finite IBL mixture model is composed of a finite number of IBL distributions, where each distribution has a certain proportion. Given a $D$-dimensional positive random vector $\mathbf{X} = (X_1, \ldots, X_D)$ that is distributed according to an IBL mixture model with $M$ components, the probability density function (pdf) is given by

$$p(\mathbf{X}|\boldsymbol{\pi}, \Theta) = \sum_{j=1}^{M} \xi_j \mathrm{IBL}(\mathbf{X}|\Theta_j), \tag{13.1}$$

where $\xi_j$ is the coefficient of the $M$ components with the constrains that $\xi_j > 0$ and $\sum_{j=1}^{M} \xi_j = 1$. $\mathrm{IBL}(\mathbf{X}|\Theta_j)$ is an IBL distribution of the $j$th component with its own parameters $\Theta_j$ and is defined by

$$\begin{aligned}\mathrm{IBL}(\mathbf{X}|\Theta_j) = {} & \frac{\Gamma(\sum_{d=1}^{D} \alpha_{jd})\Gamma(\alpha_j + \beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)} \prod_{d=1}^{D} \frac{X_d^{\alpha_{jd}-1}}{\Gamma(\alpha_{jd})} \\ & \times \lambda_j^{\beta_j} \left( \sum_{d=1}^{D} X_d \right)^{\alpha_j - \sum_{d=1}^{D} \alpha_{jd}} \left( \lambda_j + \sum_{d=1}^{D} X_d \right)^{-(\alpha_j + \beta_j)}\end{aligned} \tag{13.2}$$

where $\Theta_j = \{\boldsymbol{\alpha}_j, \alpha_j, \beta_j, \lambda_j\}$ denotes the set of IBL parameters associated with the $j$th component.

Then, a binary latent variable $\mathbf{Z_i}$ is added for each $\mathbf{X_i}$, such that if the data $\mathbf{X_i}$ belong to the $j$th component, then $Z_{ij}$ equals to 1, otherwise it equals to 0. If we assume that each $Z_{ij}$ is independent of each other, we can write the probability distribution of $\mathbf{Z_i}$ as

$$p(\mathbf{Z_i}) = \prod_{j=1}^{M} \pi_j^{Z_{ij}} \tag{13.3}$$

Though finite mixture models have been widely used in solving many computer vision problems, but they cannot get good performance in image processing, especially in image segmentation. The problem of finite mixture model is that it deals with each pixel individually and does not consider the relationship between

the current pixel and its neighbors. As a result, conventional finite mixture model is not robust to noise. In addition, conventional finite mixture models assume that each pixel $X_i$ in the component $j$ has the same coefficient $\xi_j$, which clearly is not a practical prior assumption since pixels may vary in their locations and intensity values for different components.

### 13.2.2 Finite IBL Mixture Model with Spatial Constraints

In order to make the finite IBL mixture model more robust against noise for image segmentation, we propose a spatially constrained IBL mixture model based on contextual mixing proportions and generalized means. If each pixel $\mathbf{X}_i$ follows an IBL mixture model, then its probability density function is given by

$$p(\mathbf{X}_i|\boldsymbol{\pi}_i, \Theta) = \sum_{j=1}^{M} \pi_{ij} \mathrm{IBL}(\mathbf{X}_i|\Theta_j). \tag{13.4}$$

where $\boldsymbol{\pi}_i = (\pi_{i1}, \ldots, \pi_{iM})$ is the vector that contains the contextual mixing proportions. Here, each $\pi_{ij} > 0$ represents the probability that the $i$th pixel belongs to the $j$th segment with the constraint that $\sum_{j=1}^{M} \pi_{ij} = 1$.

Next, we impose the spatial relationship between nearby pixels into our model through generalized mean. Compared with other spatial constrained models such as MRF [38], the method of incorporating generalized mean is more simpler and easier to implement. The generalized mean is a family of functions for aggregating sets of numbers. By including the generalized mean into the IBL mixture model, we have

$$p(\mathbf{X}_i|\boldsymbol{\pi}_\mathbf{i}, \boldsymbol{\Theta}) = \prod_{m \epsilon N_i} \left[ \sum_{j=1}^{M} \pi_{mj} \mathrm{IBL}(\mathbf{X}_m|\theta_j) \right]^{\frac{1}{N_i}}. \tag{13.5}$$

where $N_i$ is the neighboring pixel of the $i$th pixel. After adding the spatial constraints, the probability distribution of latent variable $\mathcal{Z}$ can be defined in terms of the contextual mixing proportions $\boldsymbol{\pi}$:

$$p(\mathcal{Z}|\boldsymbol{\pi}) = \prod_{i=1}^{N} \prod_{j=1}^{M} \pi_{ij}^{Z_{ij}}. \tag{13.6}$$

Then, the likelihood function of data set $\mathbf{X}$ is defined by

$$p(\mathbf{X}|\mathbf{Z}, \Theta) = \prod_{i=1}^{N} \prod_{m \epsilon N_i} \left[ \prod_{j=1}^{M} \mathrm{IBL}(\mathbf{X}_m|\theta_j)^{Z_{mj}} \right]^{\frac{1}{N_i}} \tag{13.7}$$

The way of integrating generalized mean in the IBL mixture model forces our model to consider the neighborhood of the $i$th pixel rather than just the $i$th pixel itself in image segmentation.

Similar to [32], we define the prior distribution of $\boldsymbol{\pi}$ using a Dirichlet distribution as

$$p(\boldsymbol{\pi}) = \prod_{i=1}^{N} \mathrm{Dir}(\boldsymbol{\pi}_i|\boldsymbol{\Lambda}_i) = \prod_{i=1}^{N} \frac{\Gamma(\sum_{j=1}^{M} a\bar{Z}_{ij}^{b})}{\prod_{j=1}^{M}\Gamma(a\bar{Z}_{ij}^{b})} \prod_{j=1}^{M} \pi_{ij}^{a\bar{Z}_{ij}^{b}-1} \tag{13.8}$$

where $\Lambda_{ij} = a\bar{Z}_{ij}^{b}$. Based on [32], $\bar{Z}_{ij}$ is defined by

$$\bar{Z}_{ij} = \left(\frac{1}{N}\sum_{i=1}^{N} r_{ij}^{p}\right)^{\frac{1}{p}} + \frac{\langle \pi_{ij} \rangle}{\sigma} \tag{13.9}$$

where $\sigma$ is the parameter that controls the smoothing contribution from the expected value $\langle \pi_{ij} \rangle$. In this work, we set the value of $\sigma$ as 2.

Next, we use Gamma distribution $G(\cdot)$ as the priors for the following parameters:

$$p(\boldsymbol{\alpha}) = G(\boldsymbol{\alpha}|\mathbf{u}, \mathbf{v}) = \prod_{j=1}^{M}\prod_{d=1}^{D} \frac{\alpha_{jd}^{u_{jd}-1} e^{-v_{jd}\alpha_{jd}}}{\Gamma(u_{jd})}, \tag{13.10}$$

$$p(\boldsymbol{\alpha}) = G(\boldsymbol{\alpha}|\mathbf{g}, \mathbf{h}) = \prod_{j=1}^{M} \frac{\alpha_{j}^{g_j-1} e^{-h_j\alpha_j}}{\Gamma(g_j)}, \tag{13.11}$$

$$p(\boldsymbol{\beta}) = G(\boldsymbol{\beta}|\mathbf{s}, \mathbf{t}) = \prod_{j=1}^{M} \frac{\beta_{j}^{s_j-1} e^{-t_j\beta_j}}{\Gamma(s_j)}, \tag{13.12}$$

$$p(\boldsymbol{\lambda}) = G(\boldsymbol{\lambda}|\mathbf{c}, \mathbf{f}) = \prod_{j=1}^{M} \frac{\lambda_{j}^{c_j-1} e^{-f_j\lambda_j}}{\Gamma(c_j)}. \tag{13.13}$$

## 13.3 Model Learning via Variational Bayes

In this part, following [33], we adopt variational Bayes to learn the proposed spatially constrained IBL mixture model. The main idea of variational Bayes is based on finding a lower bound on $p(\mathcal{X}|\boldsymbol{\pi})$ via Jensen's inequality. The lower bound $\mathcal{L}(q)$ can then be defined as:

$$\log p(\mathcal{X}) = \log \int p(\mathcal{X}, \Omega) d\Omega = \log \int q(\Omega) \frac{p(\mathcal{X}}{q(\Omega)} d\Omega$$
$$\geq \int q(\Omega) \log \frac{p(\mathcal{X}, \Omega)}{q(\Omega)} d\Omega = \mathcal{L}(q), \tag{13.14}$$

where $\Omega = \{\mathcal{Z}, \Theta, \boldsymbol{\pi}\}$ represents all random and latent variables. $q(\Omega)$ is an approximation for the posterior distribution $p(\Omega|\mathcal{X})$.

Then, we adopt the mean-field assumption to restrict the approximated posterior distribution $q(\Omega)$ by factorizing it into the product of different factors as

$$q(\Omega) = q(\mathcal{Z}) q(\Theta) q(\boldsymbol{\pi}) = q(\mathcal{Z}) q(\boldsymbol{\alpha}) q(\boldsymbol{\alpha}) q(\boldsymbol{\beta}) q(\boldsymbol{\lambda}) q(\boldsymbol{\pi}) \tag{13.15}$$

Variational solutions are obtained by maximizing the lower bound $\mathcal{L}(q)$ with respect to each factor in turn. The variational posterior $q(\mathcal{Z})$ can be updated by

$$q(\mathcal{Z}) = \prod_{i=1}^{N} \prod_{j=1}^{M} r_{ij}^{Z_{ij}}, \tag{13.16}$$

where

$$r_{ij} = \frac{r_{ij}^*}{\sum_{k=1}^{M} r_{ik}^*}, \tag{13.17}$$

$$r_{ij}^* = \exp\Bigg\{ \langle \ln \pi_{ij} \rangle + \sum_{m \epsilon N_i} \frac{1}{N_i} \Bigg[ S_j + T_j + (\bar{\alpha}_j - \sum_{d=1}^{D} \bar{\alpha}_{jd}) \log(\sum_{d=1}^{D} X_{md})$$
$$+ \bar{\beta}_j \langle \log \lambda_j \rangle + \sum_{d=1}^{D} (\bar{\alpha}_{jd} - 1) \log X_{md}) - (\bar{\alpha}_j + \bar{\beta}_j) H_{mj} \Bigg] \Bigg\} \tag{13.18}$$

The expected values in (13.18) are given by

$$\bar{\alpha}_j = \frac{g_j^*}{h_j^*}, \qquad \bar{\beta}_j = \frac{s_j^*}{t_j^*}, \qquad \bar{\alpha}_{jd} = \frac{u_{jd}^*}{v_{jd}^*}, \quad \bar{\lambda}_j = \frac{c_{jd}^*}{f_{jd}^*} \tag{13.19}$$

$$\langle Z_{ij} \rangle = r_{ij}, \qquad \langle \ln \pi_{ij} \rangle = \Psi(\Lambda_{ij}^*) - \Psi(\sum_{j=1}^{M} \Lambda_{ij}^*) \tag{13.20}$$

$$H_{ij} = \Big\langle \log(\lambda_j + \sum_{d=1}^{D} X_{id}) \Big\rangle, \quad \langle \log \lambda_j \rangle = \Psi(c_j^*) - \log(f_j^*), \tag{13.21}$$

$$S_j = \left\langle \log \frac{\Gamma(\sum_{d=1}^{D} \alpha_{jd})}{\prod_{d=1}^{D} \Gamma(\alpha_{jd})} \right\rangle, \quad T_j = \left\langle \log \frac{\Gamma(\alpha_j + \beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)} \right\rangle. \tag{13.22}$$

The expectations $H_{ij}$, $S_j$, and $T_j$ cannot be found in closed form solutions, we use second-order Taylor series expansion as in [33] to calculate lower bounds.

The variational solution to the factor $q(\boldsymbol{\alpha})$ is given by

$$q(\boldsymbol{\alpha}) = \prod_{j=1}^{M} \prod_{d=1}^{D} G(\alpha_{jd} | u_{jd}^*, v_{jd}^*), \tag{13.23}$$

where we have

$$u_{jd}^* = u_{jd} + \sum_{i=1}^{N} \sum_{m=1}^{N_i} \frac{\langle Z_{mj} \rangle}{N_i} \bar{\alpha}_{jd} \Bigg[ \Psi(\sum_{d=1}^{D} \bar{\alpha}_{jd}) - \Psi(\bar{\alpha}_{jd}) + \Psi'(\sum_{d=1}^{D} \bar{\alpha}_{jd}) \sum_{l \neq d}^{D} (\langle \log \alpha_{jl} \rangle - \log \bar{\alpha}_{jl}) \bar{\alpha}_{jl} \Bigg], \tag{13.24}$$

$$v_{jd}^* = v_{jd} - \sum_{i=1}^{N} \sum_{m=1}^{N_i} \frac{\langle Z_{mj} \rangle}{N_i} \left[ \log X_{id} - \log \left( \sum_{d=1}^{D} X_{id} \right) \right], \tag{13.25}$$

with expected values

$$\langle \log \alpha_{jl} \rangle = \Psi(u_{jl}^*) - \Psi(v_{jl}^*). \tag{13.26}$$

The optimal solution to the factor $q(\boldsymbol{\alpha})$ is obtained by

$$q(\boldsymbol{\alpha}) = \prod_{j=1}^{M} G(\alpha_j | g_j^*, h_j^*), \tag{13.27}$$

where hyperparameters $g_j^*$ and $h_j^*$ are given by

$$g_j^* = g_j + \sum_{i=1}^{N} \sum_{m=1}^{N_i} \frac{\langle Z_{mj} \rangle}{N_i} \Big[ \Psi(\bar{\alpha}_j + \bar{\beta}_j) - \Psi(\bar{\alpha}_j) + \bar{\beta}_j \Psi'(\bar{\alpha}_j + \bar{\beta}_j)(\langle \log \beta_j \rangle - \log \bar{\beta}_j) \Big] \bar{\alpha}_j, \tag{13.28}$$

$$h_j^* = h_j - \sum_{i=1}^{N}\sum_{m=1}^{N_i}\frac{\langle Z_{mj}\rangle}{N_i}\log\left(\sum_{d=1}^{D}X_{id}\right) + \sum_{i=1}^{N}\sum_{m=1}^{N_i}\frac{\langle Z_{mj}\rangle}{N_i}H_{ij}, \tag{13.29}$$

and

$$\langle \log\beta_j\rangle = \Psi(s_j^*) - \log(t_j^*). \tag{13.30}$$

Then, the variational optimal solution to $q(\boldsymbol{\beta})$ can be defined by

$$q(\boldsymbol{\beta}) = \prod_{j=1}^{M} G(\beta_j|s_j^*, t_j^*), \tag{13.31}$$

where

$$\begin{aligned} s_j^* = s_j + \sum_{i=1}^{N}\sum_{m=1}^{N_i}\frac{\langle Z_{mj}\rangle}{N_i}\Big[&\Psi(\bar{\alpha}_j + \bar{\beta}_j) - \Psi(\bar{\beta}_j) \\ &+\bar{\alpha}_j\Psi'(\bar{\alpha}_j + \bar{\beta}_j)(\langle\log\alpha_j\rangle - \log\bar{\alpha}_j)\Big]\bar{\beta}_j \end{aligned} \tag{13.32}$$

$$t_j^* = t_j + \sum_{i=1}^{N}\sum_{m=1}^{N_i}\frac{\langle Z_{mj}\rangle}{N_i}\Big[H_{ij} - \langle\log\lambda_j\rangle\Big], \tag{13.33}$$

The variational solution to $q(\boldsymbol{\pi})$ can be obtained by

$$q(\boldsymbol{\pi}) = \prod_{i=1}^{N}\prod_{j=1}^{M}\mathrm{Dir}(\pi_{ij}|\Lambda_{ij}^*) \tag{13.34}$$

where

$$\Lambda_{ij}^* = \langle Z_{ij}\rangle + \Lambda_{ij} \tag{13.35}$$

with

$$\langle Z_{ij}\rangle = r_{ij} \tag{13.36}$$

Consequently, the expected value of $\pi_{ij}$ in posterior distribution is given by

$$\langle\pi_{ij}\rangle = \frac{\Lambda_{ij}^*}{\sum_{k=1}^{M}\Lambda_{ik}^*} \tag{13.37}$$

Finally, the optimal solution to $q(\boldsymbol{\lambda})$ is given by

$$q(\boldsymbol{\lambda}) = \prod_{j=1}^{M} G(c_j | f_j^*, t_j^*), \tag{13.38}$$

where the hyperparameters are given by

$$c_j^* = c_j + \sum_{i=1}^{N} \sum_{m=1}^{N_i} \frac{\langle Z_{mj} \rangle}{N_i} \bar{\beta}_j, \tag{13.39}$$

$$f_j^* = f_j + \sum_{i=1}^{N} \sum_{m=1}^{N_i} \frac{\langle Z_{mj} \rangle}{N_i} \frac{\bar{\alpha}_j + \bar{\beta}_j}{\bar{\lambda}_j + \sum_{d=1}^{D} X_{id}}. \tag{13.40}$$

In our case, the lower bound $\mathcal{L}(q)$ can then be calculated by

$$\begin{aligned} \mathcal{L}(q) &= \sum_{\mathcal{Z}} \int q(\mathcal{Z}, \Theta, \boldsymbol{\pi}) \log \left\{ \frac{p(\mathcal{X}, \mathcal{Z}, \Theta, \boldsymbol{\pi})}{q(\mathcal{Z}, \Theta), \boldsymbol{\pi}} \right\} d\Theta \\ &= \langle \log\, p(\mathcal{X} | \mathcal{Z}, \Theta, \boldsymbol{\pi}) \rangle + \langle \log\, p(\mathcal{Z} | \boldsymbol{\pi}) \rangle + \langle \log\, p(\boldsymbol{\pi}) \rangle + \langle \log\, p(\Theta) \rangle \\ &\quad - \langle \log q(\mathcal{Z}) \rangle - \langle \log q(\boldsymbol{\pi}) \rangle - \langle \log\, q(\Theta) \rangle. \end{aligned} \tag{13.41}$$

The complete algorithm of estimating the parameters of the proposed spatially constrained IBL mixture model with variational Bayes can be summarized as follows

**Algorithm 1**

1: Initialize the number of components $M$.
2: Initialize values of hyperparameters $u_{jd}, v_{jd}, g_j, h_j, s_j, t_j, c_j, f_j$.
3: Initialize the values of $r_{ij}$ by using K-means.
4: **repeat**
5: Variational E-step:
Calculate the expected values (13.19), (13.20), (13.21), (13.22) and (13.26).
6: Variational M-step:
Update the variational solutions by using (13.16), (13.23), (13.27), (13.31), (13.34) and (13.38).
7: **until** convergence is reached
8: Calculate the $\langle \pi_{ij} \rangle$ using (13.37).

## 13.4 Experimental Results

In our experiments, we use both simulated and real brain MRI images to validate the proposed image segmentation method that is based on the spatially constrained IBL mixture model (denoted by IBLMM-SC). We compare IBLMM-SC with other well-formulated image segmentation methods based on mixture models, such as the conventional Gaussian mixture model (GMM) [39], the fast and robust spatially constrained Gaussian mixture model (FRSCGMM) [18], the spatially constrained Dirichlet mixture model (SC-DMM) [30], the spatially constrained Student's t mixture model (SMM-SC) [40], the spatially constrained inverted Dirichlet mixture model (IDMM-SC) [31], and the spatially constrained Beta-Liouville mixture model (BLMM-SC) [32].

Our experiments can be divided into two parts: First, we test our method on simulated brain MRI images with different levels of noise. Second, we test our method with real MRI brain images. We employ the misclassification ratio [41] (MCR) to measure the performance of the segmentation result which is defined by

$$\text{MCR} = \frac{\text{number of misclassified pixels}}{\text{total number of pixels}} \times 100 \tag{13.42}$$

### *13.4.1 Synthetic MRI Brain Images*

In this experiment, we test the proposed image segmentation method on a publicly available simulated brain database (SBD), namely the BrainWeb[1] [42]. The main target of this application is to segment the simulated brain MRI image into three segments including: cerebrospinal fluid (CSF), gray matter (GM), and white matter (WM). It is noteworthy that all non-brain tissues are removed from original images as a preprocessing step.

To begin with, we use a simulated image with 9% noise (index = 50) as shown in Fig. 13.1a, the size of the image is 181 × 217. The segmentation results by different segmentation methods are demonstrated in Fig. 13.1 with the ground truth that is shown in Fig. 13.1b. The segmentation results of the proposed method and other tested ones are shown in Fig. 13.1c–i. Based on these results, we can see that the proposed IBLMM-SC can provide more visually appealing segmentation result with less noise as shown in Fig. 13.1c, and with the lowest MCR value 4.87% among all tested methods.

Next, we conduct experiment on another simulated MRI image with 7% noise (index = 100) as demonstrated in Fig. 13.2a. The ground truth of the segmentation result can be seen in Fig. 13.2b. The segmentation result of the proposed method is shown in Fig. 13.2c. As compared with other tested methods as illustrated

[1] http://brainweb.bic.mni.mcgill.ca/brainweb/.

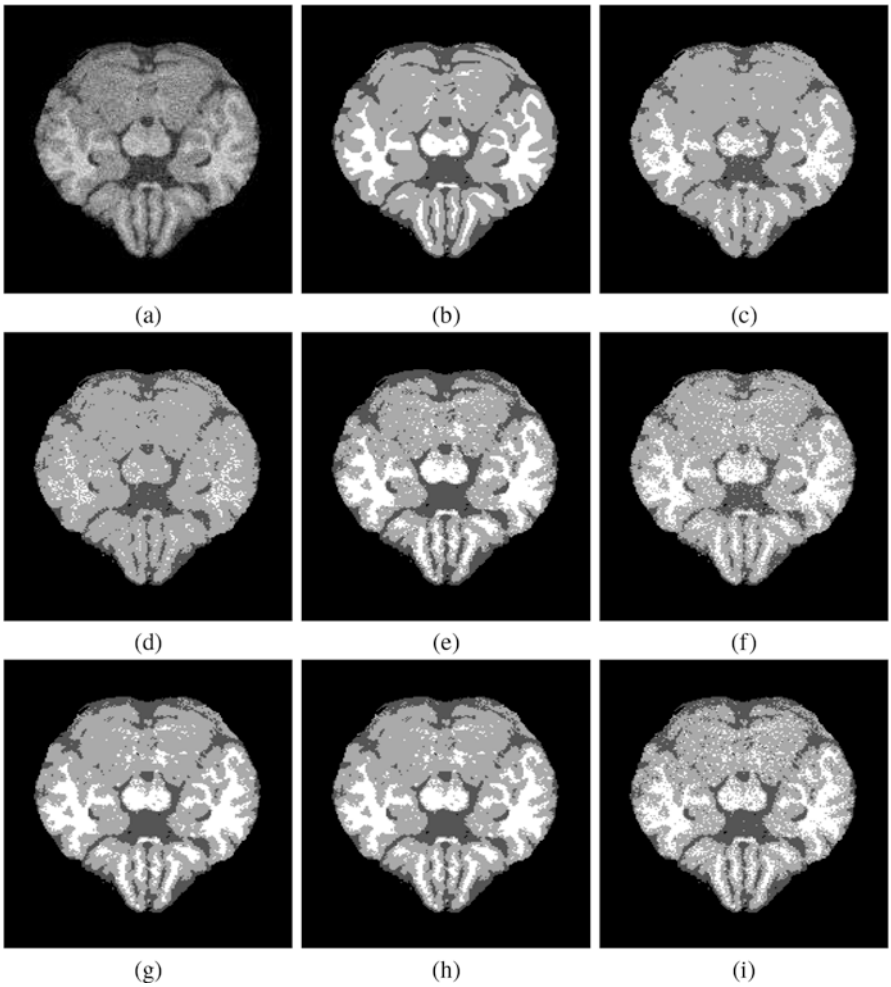



**Fig. 13.1** (**a**) Original image with 9% noise. (**b**) Ground truth. (**c**) IBLMM-SC (MCR = 4.87%). (**d**) GMM (MCR = 8.10%). (**e**) FRSCGMM (MCR = 6.95%). (**f**) SMM (MCR = 7.23%). (**g**) DMM-SC (MCR = 5.23%). (**h**) IDMM-SC (MCR = 5.38%) (**i**) BLMM (MCR = 8.87%)

in Fig. 13.2d–i, the proposed IBLMM-SC can provide better segmentation both qualitatively (with less noise) and quantitatively (with the smallest MCR rate 2.12%).

To better validate our segmentation method, we conduct experiments on all simulated images with 7%, 9% noise levels. The average segmentation results under different noise levels are shown in Table 13.1. From this table, it is clear that the

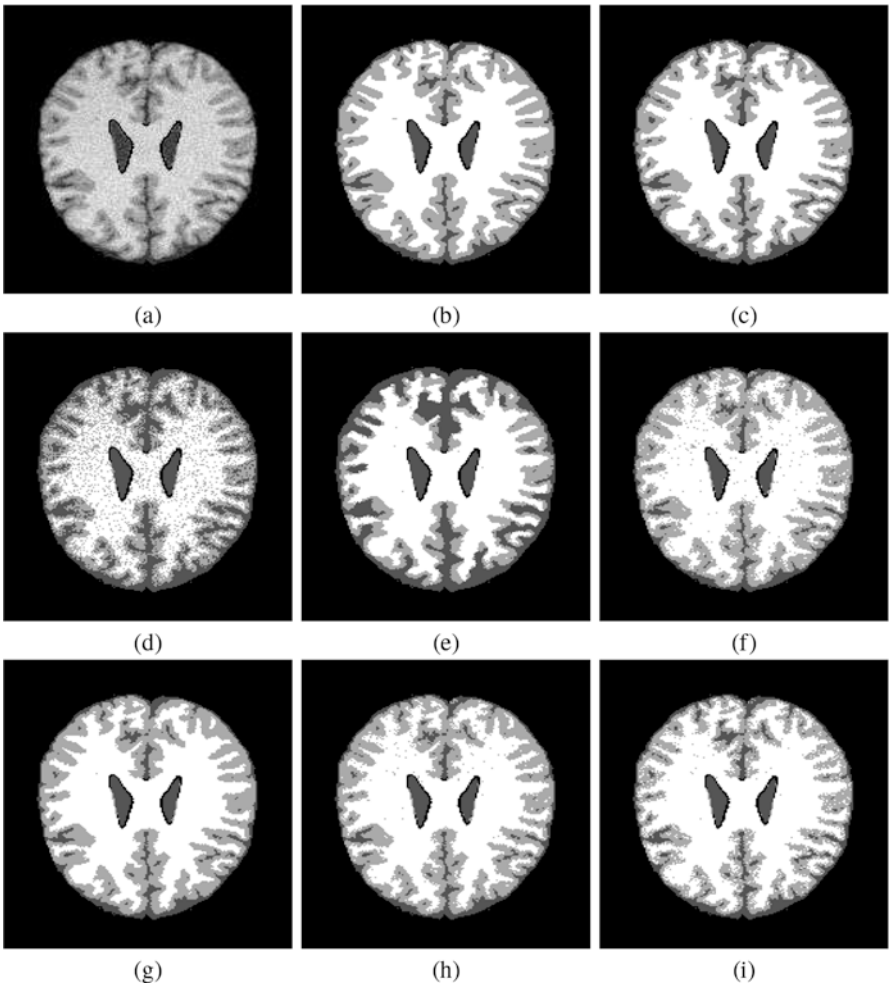



**Fig. 13.2** (**a**) Original image with 7% noise. (**b**) Ground truth. (**c**) The proposed IBLMM-SC (MCR = 2.12%). (**d**) GMM (MCR = 13.20%). (**e**) FRSCGMM (MCR = 8.37%). (**f**) SMM (MCR = 5.21%). (**g**) DMM-SC (MCR = 3.87%). (**h**) IDMM-SC (MCR = 4.07%). (**i**) BLMM (MCR = 4.44%)

proposed IBLMM-SC is more robust to noise and has obtained the best performance among all applied methods in terms of the lowest MCR for both tested noise levels.

**Table 13.1** The average MCR(%) for comparison with other segmentation methods with 7% and 9% noise levels

| Method | Noise=7% | Noise=9% |
|---|---|---|
| GMM | 10.54 | 13.89 |
| FRSCGMM | 8.22 | 8.7 |
| SMM | 7.53 | 9.23 |
| DMM-SC | 6.20 | 7.23 |
| IDMM-SC | 6.84 | 8.23 |
| BLMM | 4.64 | 5.87 |
| IBLMM-SC | **3.62** | **4.73** |

Bold values indicate the best performance in terms of the lowest MCR rates

**Table 13.2** The average MCR (%) for comparison with other methods for the IBSR02 data set and the whole IBSR data set

| Method | IBSR02 | Whole IBSR |
|---|---|---|
| GMM | 13.25 | 15.83 |
| FRSCGMM | 6.48 | 8.70 |
| SMM | 7.62 | 10.05 |
| DMM-SC | 5.65 | 7.58 |
| IDMM-SC | 5.17 | 8.03 |
| BLMM | 4.80 | 6.14 |
| IBLMM-SC | **3.71** | **5.35** |

Bold values indicate the best performance in terms of the lowest MCR rates

### *13.4.2 Real MRI Brain Images*

In this section, we conduct experiments on real medical images using the Internet Brain Segmentation Repository (IBSR),[2] which contains real magnetic resonance brain image data along with manually guided expert segmentation results. The MCR is used in this experiment to measure the segmentation performance. In the first experiment, we show the segmentation performance of the proposed method using two real MRI brain images as demonstrated in Figs. 13.3a and 13.4a, respectively. The first tested image can be seen in Fig. 13.3a with segmentation ground truth that is shown in Fig. 13.3b. In the preprocessing step, we removed all non-brain tissues. The segmentation results obtained by all methods are shown in Fig. 13.3c–i. According to these results, we can observe that the proposed IBLMM-SC can provide better performance with the lowest MCR (2.22%) than other tested methods.

The second tested image is shown in Fig. 13.4a with the ground truth segmentation result can be seen in Fig. 13.4b. The segmentation results of different methods are provided in Fig. 13.4c–i. Once again, the best segmentation performance was obtained by the proposed IBLMM-SC among all methods in terms of the lowest MCR value (3.96%).

[2] http://www.nitrc.org/projects/ibsr/.

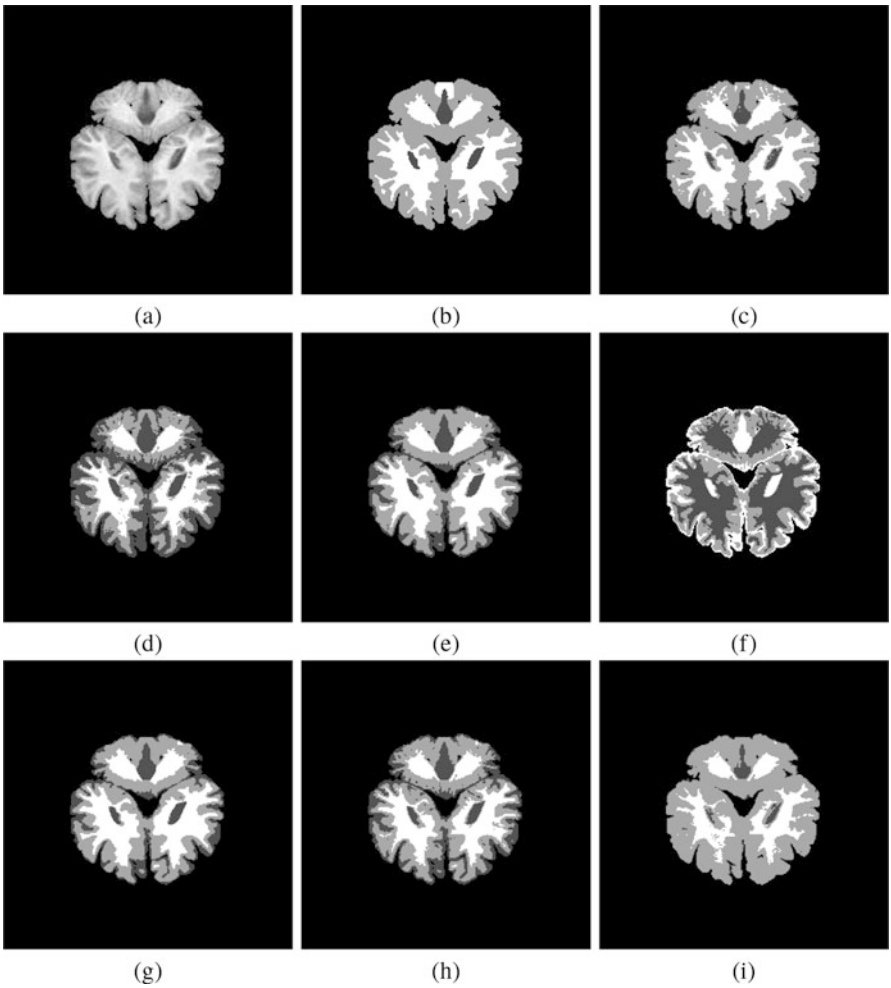



**Fig. 13.3** (**a**) Original image. (**b**) Ground truth. (**c**) IBLMM-SC (MCR = 2.22%). (**d**) GMM (MCR = 9.80%). (**e**) FGFCM (MCR = 6.10%). (**f**) SMM (MCR = 5.92%). (**g**) DMM-SC (MCR = 5.49%). (**h**) IDMM-SC (MCR = 6.56%). (**i**) BLMM-SC (MCR = 3.53%)

Next, we test our image segmentation method using all real MRI brain images in IBSR02 and the whole IBSR data set by reporting average performance shown in Table 13.2. Clearly, the proposed method has obtained the best segmentation results with the lowest MCR values for all tested data sets.

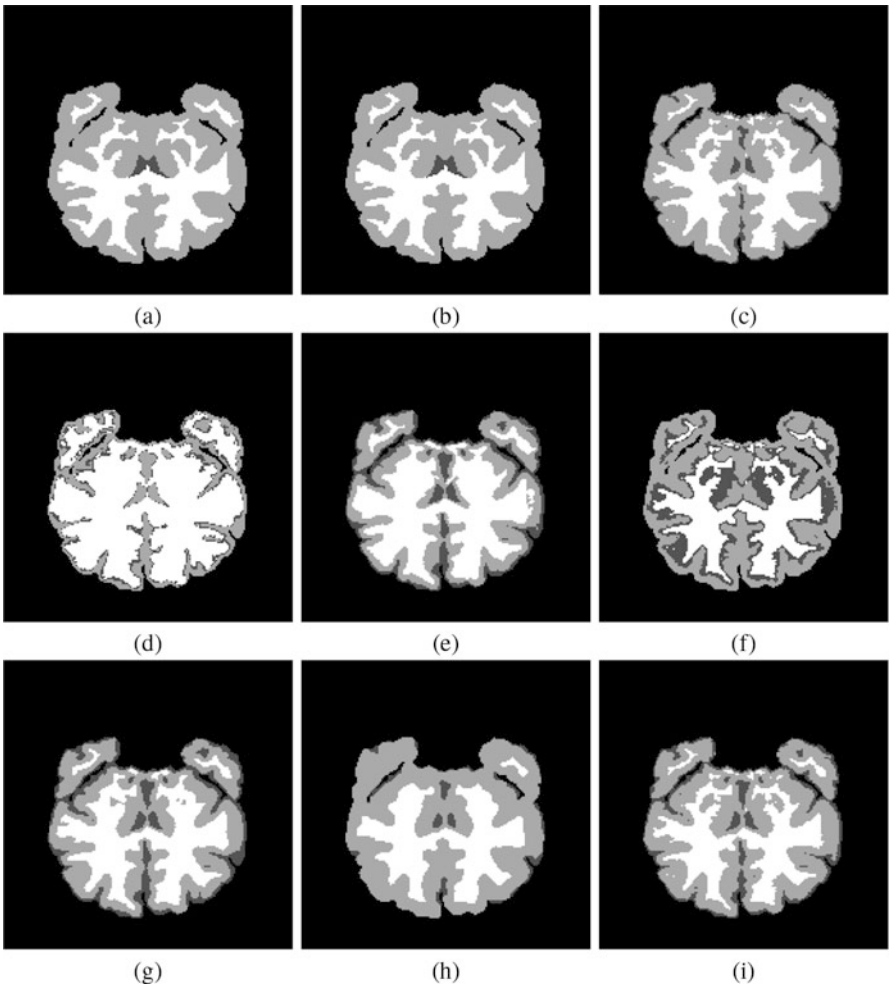



**Fig. 13.4** (**a**) Original image. (**b**) Ground truth. (**c**) IBLMM-SC (MCR = 3.96%). (**d**) GMM (MCR = 14.68%). (**e**) FGFCM (MCR = 9.12%). (**f**) SMM (MCR = 8.91%). (**g**) DMM-SC (MCR = 7.26%). (**h**) IDMM-SC (MCR = 4.93%). (**i**) BLMM-SC (MCR = 6.82%)

## 13.5 Conclusion

In this chapter, we proposed a new image segmentation method for segmenting medical MRI brain images. The proposed method is based on finite inverted Beta-Liouville mixtures which demonstrated better performance than commonly used mixture models (such as Gaussian mixture model) in image segmentation. To improve the robustness of our image segmentation method against noise, the spatial relationship among nearby pixels was incorporated into our model by using

generalized means. In order to learn the proposed spatially constrained mixture model, a variational Bayes inference algorithm was developed, such that model parameters can be efficiently estimated in closed form. Both simulated and real brain MRI data were used to validate our model.

**Acknowledgements** The completion of this work was supported by the National Natural Science Foundation of China (61876068, 61572205), the Natural Science Foundation of Fujian Province (2018J01094), and the Promotion Program for young and middle-aged teacher in Science and Technology Research of Huaqiao University (ZQNPY510).

## References

1. Huang, X., Tsechpenakis, G.: Medical image segmentation. Inf. Discov. Electron. Health Rec. **10**, 251–289 (2009)
2. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 878–885 (2005)
3. Samaria, F., Young, S.: Hmm-based architecture for face identification. Image Vis. Comput. **12**(8), 537–543 (1994)
4. Pham, D.L., Xu, C., Prince, J.L.: A survey of current methods in medical image segmentation. Annu. Rev. Biomed. Eng. **2**(2000), 315–337 (2000)
5. Kalpathy-Cramer, J., Hersh, W.: Medical image retrieval and automatic annotation: OHSU at ImageCLEF 2007. In: Advances in Multilingual and Multimodal Information Retrieval, pp. 623–630 (2008)
6. Fabijańska, A.: Variance filter for edge detection and edge-based image segmentation. In: Viith International Conference on Perspective Technologies and Methods in Mems Design (2011)
7. Silva, L., Bellon, O.R.P., Gotardo, P.F.U.: Edge-based image segmentation using curvature sign maps from reflectance and range images. In: International Conference on Image Processing (2001)
8. Adamek, T., O'Connor, N.E.: Stopping region-based image segmentation at meaningful partitions. In: Semantic and Digital Media Technologies International Conference on Semantic Multimedia (2007)
9. Fauqueur, J., Boujemaa, N.: Region-based image retrieval: fast coarse segmentation and fine color description. J. Vis. Lang. Comput. **15**(1), 69–95 (2004)
10. Monteiro, F.C., Campilho, A.: Watershed framework to region-based image segmentation. In: International Conference on Pattern Recognition (2008)
11. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
12. Zabih, R., Kolmogorov, V.: Spatially coherent clustering using graph cuts. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2004)
13. Mei, Y.C., Wei, L.K., Wei, Y.K., Angeline, L., Teo, K.T.K.: Graph-based image segmentation using k-means clustering and normalised cuts. In: Fourth International Conference on Computational Intelligence (2012)
14. Cai, W., Chen, S., Zhang, D.: Fast and robust fuzzy -means clustering algorithms incorporating local information for image segmentation. Pattern Recogn. **40**(3), 825–838 (2007)
15. Flury, B.: Algorithms for clustering data: Anil K. Jain and Richard C. Dubes Prentice Hall advanced reference series in computer science Prentice Hall, Englewood cliffs, NJ (1988). J. Stat. Plan. Inference **21**(1), 137–138 (1989)
16. Blekas, K., Likas, A., Galatsanos, N.P., Lagaris, I.E.: A spatially constrained mixture model for image segmentation. IEEE Trans. Neural Netw. **16**(2), 494–498 (2005)

17. Nikou, C., Likas, A.C., Galatsanos, N.P.: A bayesian framework for image segmentation with spatially varying mixtures. IEEE Trans. Image Process. **19**(9), 2278–2289 (2010)
18. Nguyen, T.M., Wu, Q.M.J.: Fast and robust spatially constrained Gaussian mixture model for image segmentation. IEEE Trans. Circuits Syst. Video Technol. **23**(4), 621–635 (2013)
19. Nguyen, T.M., Wu, Q.M.J.: Gaussian-mixture-model-based spatial neighborhood relationships for pixel labeling problem. IEEE Trans. Syst. Man Cybern. B Cybern. **42**(1), 193–202 (2012)
20. Fan, W., Sallay, H., Bouguila, N.: Online learning of hierarchical Pitman–Yor process mixture of generalized Dirichlet distributions with feature selection. IEEE Trans. Neural Netw. Learn. Syst. **28**(9), 2048–2061 (2017)
21. Fan, W., Bouguila, N., Ziou, D.: Variational learning for finite Dirichlet mixture models and applications. IEEE Trans. Neural Netw. Learn. Syst. **23**(5), 762–774 (2012)
22. Bdiri, T., Bouguila, N.: Positive vectors clustering using inverted Dirichlet finite mixture models. Expert Syst. Appl. **39**(2), 1869–1882 (2012)
23. Bdiri, T., Bouguila, N.: Bayesian learning of inverted Dirichlet mixtures for SVM kernels generation. Neural Comput. Appl. **23**(5), 1443–1458 (2013)
24. Bdiri, T., Bouguila, N., Ziou, D.: A statistical framework for online learning using adjustable model selection criteria. Eng. Appl. Artif. Intell. **49**, 19–42 (2016)
25. Mashrgy, M.A., Bdiri, T., Bouguila, N.: Robust simultaneous positive data clustering and unsupervised feature selection using generalized inverted Dirichlet mixture models. Knowl. Based Syst. **59**, 182–195 (2014)
26. Bdiri, T., Bouguila, N., Ziou, D.: Variational bayesian inference for infinite generalized inverted Dirichlet mixtures with feature selection and its application to clustering. Appl. Intell. **44**(3), 507–525 (2016)
27. Fan, W., Bouguila, N., Liu, X.: A hierarchical Dirichlet process mixture of GID distributions with feature selection for spatio-temporal video modeling and segmentation. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, pp. 2771–2775 (2017)
28. Fan, W., Bouguila, N.: Online learning of a Dirichlet process mixture of Beta-Liouville distributions via variational inference. IEEE Trans. Neural Netw. Learn. Syst. **24**(11), 1850–1862 (2013)
29. Fan, W., Bouguila, N.: Expectation propagation learning of a Dirichlet process mixture of Beta-Liouville distributions for proportional data clustering. Eng. Appl. Artif. Intell. **43**, 1–14 (2015)
30. Hu, C., Fan, W., Du, J., Zeng, Y.: Model-based segmentation of image data using spatially constrained mixture models. Neurocomputing **283**, 214–227 (2018)
31. Fan, W., Hu, C., Du, J., Bouguila, N.: A novel model-based approach for medical image segmentation using spatially constrained inverted Dirichlet mixture models. Neural Process. Lett. **47**(2), 619–639 (2018)
32. Hu, C., Fan, W., Du, J.X., Nan, X.: Spatially variant mixture model for natural image segmentation. J. Electron. Imaging **26**(4), 043005 (2017)
33. Hu, C., Fan, W., Du, J., Bouguila, N.: A novel statistical approach for clustering positive data based on finite inverted Beta-Liouville mixture models. Neurocomputing **333**, 110–123 (2019)
34. Hui, Z., Wu, Q.M.J., Nguyen, T.M.: Image segmentation by Dirichlet process mixture model with generalised mean. Iet Image Process. **8**(2), 103–111 (2013)
35. Attias, H.: A variational Bayes framework for graphical models. In: Proceedings of Advances in Neural Information Processing Systems (NIPS), pp. 209–215 (1999)
36. Jordan, M., Ghahramani, Z., Jaakkola, T., Saul, L.: An introduction to variational methods for graphical models. Mach. Learn. **37**, 183–233 (1999)
37. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
38. Li, S.Z.: Markov Random Field Modeling in Image Analysis. Springer, London (2009)
39. Zhu, Y., Fujimura, K.: Driver face tracking using Gaussian mixture model(gmm). In: IEEE Intelligent Vehicles Symposium (2003)

40. Nguyen, T.M., Wu, Q.M.J.: Robust student's-t mixture model with spatial constraints and its application in medical image segmentation. IEEE Trans. Med. Imaging **31**(1), 103–116 (2012)
41. Zhang, Y., Brady, M., Smith, S.: Segmentation of brain MR images through a hidden markov random field model and the expectation-maximization algorithm. IEEE Trans. Med. Imaging **20**(1), 45–57 (2002)
42. Cocosco, C.A., Kollokian, V., Kwan, R.K.-S., Pike, G.B., Evans, A.C.: Brainweb: Online interface to a 3D MRI simulated brain database. NeuroImage **5**, 425 (1997)

# Chapter 14
# Flexible Statistical Learning Model for Unsupervised Image Modeling and Segmentation


**Ines Channoufi, Fatma Najar, Sami Bourouis, Muhammad Azam, Alrence S. Halibas, Roobaea Alroobaea, and Ali Al-Badi**

**Abstract** We propose in this work to improve the tasks of image segmentation and modeling through an unsupervised flexible learning approach. Our focus here is to develop an alternative mixture model based on a bounded generalized Gaussian distribution, which is less sensitive to over-segmentation and offers more flexibility in data modeling than the Gaussian distribution which is certainly not the best approximation for image segmentation. A maximum likelihood- (ML) based

I. Channoufi
Université de Tunis El Manar, Ecole Nationale d'Ingénieurs de Tunis, LR-SITI Laboratoire Signal, Image et Technologies de l'Information, Tunis, Tunisie
e-mail: ines.channoufi@esprit.tn

F. Najar
Laboratoire RISC Robotique Informatique et Systémes Complexes, Université de Tunis El Manar, ENIT, Tunis, Tunisie
e-mail: fatma.najjar@enit.utm.tn

S. Bourouis (✉)
Taif University, Taif, Saudi Arabia

Université de Tunis El Manar, LR-SITI Laboratoire Signal, Image et Technologies de l'Information, Tunis, Tunisie
e-mail: s.bourouis@tu.edu.sa

M. Azam
Department of Electrical and Computer Engineering (ECE), Concordia University, Montreal, QC, Canada
e-mail: mu_azam@encs.concordia.ca

A. S. Halibas · A. Al-Badi
Gulf College, Al Maabelah, Muscat, Oman
e-mail: alrence@gulfcollege.edu.om; aalbadi@gulfcollege.edu.om

R. Alroobaea
Taif University, Taif, Saudi Arabia
e-mail: r.robai@tu.edu.sa



N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6_14

algorithm is applied for estimating the resulted model parameters. We investigate here the integration of both a spatial information (a prior information between neighboring pixels) and a minimum description length (MDL) principle into the model learning step in order to deal with the major problems of finding the optimal number of classes and also selecting the best model that describes accurately the dataset. Therefore, the proposed model has the advantage to maintain the balance between model complexity and goodness of fit. Obtained results on a large database of medical MR images confirm the effectiveness of the proposed approach and demonstrate its superior performance compared to some conventional methods.

## 14.1 Introduction

Data modeling and image segmentation are two of the most difficult and challenging problems in image processing. Their importance is highlighted by the large number of applications that include image or video segmentation step. Among these applications, content-based image retrieval [16, 50], visual scene interpretation [60], object recognition [57], remote sensing [6, 35, 59, 74], and so forth. In these applications, the adoption of a specific segmentation algorithm is the most difficult problem to enhance the final results. For this reason, various approaches have been proposed in literature and adopted in the above-mentioned applications. Among the different approaches which have been widely used, statistical models [6, 13, 22, 25, 30, 45, 67, 70, 71] are often used in image segmentation according to their simplicity when describing images features and their ability for data classification. Among these statistical-based techniques, the Gaussian mixture model (GMM) has been widely used and it has shown its importance through many applications from machine learning, pattern recognition, and image processing [9, 34, 43, 67]. An advantage of GMM is that it requires a small number of parameters for learning which can be accurately estimated by adopting the expectation maximization (EM) algorithm [23, 68] to maximize the log-likelihood function. However, this kind of model is often very sensitive to outliers and is certainly not the best approximation for image segmentation. As well, for many problems, the tail of the Gaussian distribution is shorter than what is required. In order to enhance the robustness and accuracy of this model, the generalized Gaussian mixture model (GGMM) has been proposed as an alternative solution to overcome the above limitations [20, 24, 36, 39, 46–48]. The GGMM has been successfully used according to its flexibility to model data with different shapes. It has been used in image and video coding [6, 43], texture discrimination and retrieval [20], and so on. This distribution has one more parameter $\lambda$ than the Gaussian distribution which controls the tails of the distribution and decides whether the latter is peaked or flat. The Gaussian distribution is considered as a particular case for the generalized Gaussian distribution, where $\lambda=2$. Thus, the generalized Gaussian mixture model provides a flexibility to fit the shape

of the data better than the Gaussian mixture model. However, it is important to note that conventional Gaussian-based models (GMM and GGMM) have a common issue related to their supports which are unbounded. In fact, in many applications, the observed data are digitalized and have bounded support and this statement can be exploited to select the appropriate model shape. Fortunately, the bounded Gaussian distribution and the bounded generalized Gaussian distribution have been developed in [5, 17–19, 38, 49] as an alternative generalized model which includes Gaussian model (GMM), Laplacian model (LMM), and generalized Gaussian model (GGMM) as special cases. This new distribution has the advantage to fit different shapes of observed non-Gaussian data defined within a bounded support. Moreover, it is possible to model the observed data in each component of the model with different bounded support regions. On the other hand, several developed approaches in the literature for image segmentation are facing another common problem that concerns the automatic determining of adequate components (clusters or classes) which best describes the data. In order to deal with this issue, some approaches have been suggested with relative success (see, for instance, [11, 43, 56]). In particular, the spatial information was used as a prior information about the expected number of regions in [11]. The minimum description length (MDL) criterion was also used for the same purpose in [10, 43, 56]. Motivated by all these observations, we introduce in this work an alternative flexible mixture model based on the bounded generalized Gaussian distribution (BGGMM) which incorporates a spatial information and the MDL penalty to overcome the previous limitations and to improve image segmentation and data modeling tasks. It is noteworthy that the proposed framework has never been proposed before.

The remainder of this paper is organized as follows. Our statistical model is proposed in Sect. 14.2. The model parameters estimation and the complete segmentation algorithm are described in the next section. In Sect. 14.4, we present and discuss the experimental results. Finally, we end with conclusions of this work in Sect. 14.5.

## 14.2 Bounded Generalized Gaussian Mixture Model

Let $\mathcal{X}$ be an image characterized by a set of pixels $\mathcal{X} = \{X_1, \ldots, X_N\}$, where $N$ is the number of pixels. Since an image is composed of several regions, so it could be described using a mixture model with $K$ components:

$$p(X_i|\Theta) = \sum_{j=1}^{K} \pi_j f(X_i|\theta_j) \tag{14.1}$$

provided that $\pi_j \geq 0$ and $\sum_{j=1}^{K} \pi_j = 1$. In Eq. (14.1), $f(X_i|\theta_j)$ is the probability density function associated with the region $j$, $\theta_j$ represents the set of parameters defining $j$th component, $\pi_j$ are the mixing proportions, $\Theta = \{\theta_1, \ldots, \theta_K, \pi_1, \ldots, \pi_K\}$ is the complete set of parameters to characterize the mixture model, and $K \geq 1$ is the number of components in the mixture model [43]. With a mixture of $K$ components, likelihood of data $\mathcal{X}$ with $N$ independent and identically distributed data points (pixels in our case) can be expressed as:

$$p(\mathcal{X}|\Theta) = \prod_{i=1}^{N} \sum_{j=1}^{K} \pi_j f(X_i|\theta_j) \tag{14.2}$$

The main aim of using statistical modeling consists of adopting a model which can describe accurately the statistical properties of the underlying source. In order to model the distributions, the mixture models are based on the probability density function $f(X_i|\theta_j)$ mentioned in Eqs. (14.1) and (14.2). The problem here is the choice of mixture probability density functions. In general, the Gaussian distribution is considered one of the most appropriate used distributions. This model has been widely adopted in machine learning, pattern recognition, and speech processing applications. In Gaussian mixture model (GMM), $f(X_i|\theta_j)$ is Gaussian distribution as follows:

$$f(X_i|\theta_j) = \frac{1}{(2\pi {\sigma_j}^2)^{1/2}} \exp\left\{-\frac{1}{2{\sigma_j}^2}(X_i - \mu_j)^2\right\} \tag{14.3}$$

where $\theta_j = (\mu_j, \sigma_j)$ represents the set of parameters defining $j$th component. Note that $\mu$ and $\sigma$ are the mean and standard deviation, respectively. However, in order to develop this mixture model and to control the tail of the distribution, another proposed distribution could provide better modeling capabilities which is based on the integration of the shape parameter $\lambda$. This distribution is called generalized Gaussian distribution (GGD). For its flexibility to model data with different shapes, this model has been widely used, especially in signal processing, speech modeling, and image and video coding. In generalized Gaussian mixture model (GGMM), the probability density function $f(X_i|\theta_j)$ is GGD as:

$$f(X_i|\theta_j) = \frac{\lambda_j \sqrt{\frac{\Gamma(3/\lambda_j)}{\Gamma(1/\lambda_j)}}}{2\sigma_j \Gamma(1/\lambda_j)} \exp\left(-A(\lambda_j)\left|\frac{X_i - \mu_j}{\sigma_j}\right|^{\lambda_j}\right) \tag{14.4}$$

with

$$A(\lambda_j) = \left[\frac{\Gamma(3/\lambda_j)}{\Gamma(1/\lambda_j)}\right]^{\lambda_j/2} \tag{14.5}$$

where $\Gamma(.)$ represents gamma function and $\theta_j = (\mu_j, \sigma_j, \lambda_j)$ is the set of parameters of $j$th component. Note that $\mu$, $\sigma$, and $\lambda$ are the mean, the standard deviation, and the shape parameters, respectively. The above-mentioned distributions are unbounded with support range $(-\infty, +\infty)$. However, many real applications have their data within a bounded support regions. In order to deal with this problem, the bounded generalized Gaussian mixture model (BGGMM) is proposed. This model has the flexibility to fit different shapes of observed data. If we represent the probability distribution given in Eq. (14.4) as notation $f_{ggd}(X_i|\theta_j)$, then proposed bounded generalized Gaussian distribution (BGGD) for BGGMM is given as follows:

$$f(X_i|\theta_j) = \frac{f_{ggd}(X_i|\theta_j)\mathrm{H}(X_i|\Omega_j)}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j)d\mathrm{u}} \tag{14.6}$$

where $\mathrm{H}(X_i|\Omega_j)$ is called indicator function which defines $\partial_j$ to be the bounded support region in $\Re$ for each component $\Omega_j$ as:

$$\mathrm{H}(X_i|\Omega_j) = \begin{cases} 1 & \text{if} \quad X_i \in \partial_j \\ 0 & \text{Otherwise} \end{cases} \tag{14.7}$$

The term $\int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j)d\mathrm{u}$ in Eq. (14.6) is the normalization constant which indicates the share of $f_{ggd}(X_i|\theta_j)$ which belongs to the support region $\partial_j$. As presented in [49], the proposed distribution in Eq. (14.6) allows more flexibility in order to fit the data (pixels in our case) with the limited range more efficiently than a Gaussian distribution or a generalized Gaussian distribution. By employing bounded generalized Gaussian distribution (From Eq. (14.6)) in Eq. (14.2), the complete data likelihood function for BGGMM can be described as follows:

$$p(\mathcal{X}|\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{K} \pi_j \left\{ \frac{f_{ggd}(X_i|\theta_j)\mathrm{H}(X_i|\Omega_j)}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j)d\mathrm{u}} \right\} \tag{14.8}$$

where the complete set of parameters to fully characterize the mixture is described by $\Theta = (\mu_1, \ldots, \mu_K, \sigma_1, \ldots, \sigma_K, \lambda_1, \ldots, \lambda_K, \pi_1, \ldots, \pi_K)$. As mentioned in Eq. (14.8), each component in the proposed model has the ability to model the data with different bounded support regions $\partial_j$. In order to compare the mathematical expressions according to this model with those proposed with GMM or GGMM we can conclude that for example if we set $\lambda_j = 2$ and $H(x_i|\Omega_j) = 1$ for each $\Omega_j$, the method will be similar to GMM. Therefore, we could say that BGGMM is a generalization of GMM and GGMM models (Fig. 14.1).

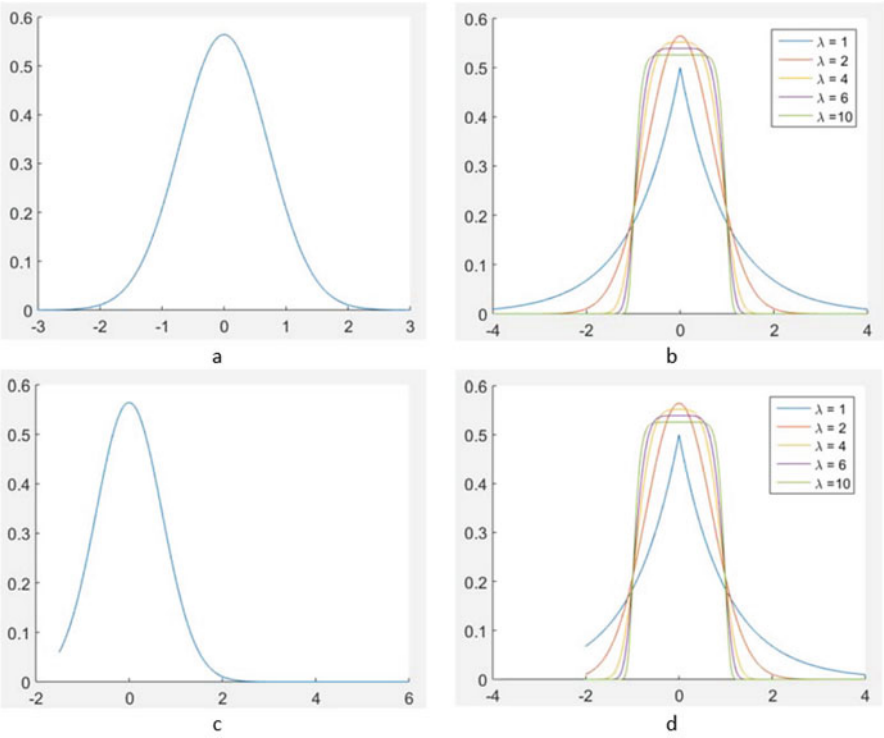



**Fig. 14.1** Illustration of different distributions: (**a**) GMM distribution; (**b**) GGMM distribution with different values of $\lambda = 1, 2, 4, 6, 10$; (**c**) BGMM distribution, the observed data in the interval $(-1.5,6)$; (**d**) BGGMM distribution with different values of $\lambda = 1, 2, 4, 6, 10$ and the observed data in the interval $(-2,4)$

## 14.3 BGGMM Learning Using EM, SI and MDL for Image Segmentation

In the following, we present our unsupervised learning method for image segmentation. In particular, we incorporate both spatial information (SI) and MDL criterion into the EM algorithm to estimate the model's parameters and to find the optimal number of model components.

### *14.3.1 Integration of the Spatial Information*

In this work, we incorporate the spatial information (SI) as a prior information between neighboring pixels in our developed model. It is useful as an implicit prior information about the probable number of regions as indicated in [59]. For each

pixel $X_i \in \mathcal{X}$ its immediate neighbor $\widehat{X}_i \in \mathcal{X}$, which we call the peer of $X_i$ and is supposed to have arisen from the same cluster of $X_i$. Thus, we can use this spatial information as indirect information in order to estimate the number of clusters, since we suppose that the peers stay in the same cluster. As though each region is composed by similar adjacent pixels, when we integrate spatial information, this enables us to find out accurate and smooth segments. For example, if a large value is fixed for $K$, there would be a conflict with the provided spatial information. This means that a true segment is wrongly divided into two sub-segments which have to be merged to a new segment that we should re-estimate its related parameters. Thus, the number of segments will decrease to reach its true number.

### *14.3.2 Mixture Model's Parameters Estimation Using EM*

In this section, we develop the equations that learn the parameters of the BGGMM using the common EM approach. First, we suppose that the number of components M is known. Many approaches have been developed in order to deal with mixture models parameters estimation [43]. One of the most popular and used estimation methods is the maximum likelihood approach in which the main idea is to find parameters that maximize the joint probability density function. This task can be performed using the expectation maximization (EM) algorithm which is widely used in the case of missing data. In our case, the missing data are the knowledge of the pixel classes. Let $\mathcal{X}$ and the set of peers $\widehat{\mathcal{X}} = \left\{\widehat{X}_1, \ldots, \widehat{X}_N\right\}$ be our observed data. The set of group indicators for all pixels $\mathcal{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_N\}$ correspond to the unobserved data, where $\mathbf{Z}_i = \{Z_{i1}, \ldots, Z_{iK}\}$ is the missing group indicator and $Z_{ij}$ is equal to one if $X_i$ belongs to the same cluster $j$ as $\widehat{X}_i$ and zero otherwise. By taking into account the spatial information and missing group indicator, complete data likelihood function referring to Eqs. (14.2) and (14.8) can be described as follows:

$$p(\mathcal{X}, \widehat{\mathcal{X}}, Z|\Theta) = \prod_{i=1}^{N}\prod_{j=1}^{K}\left[\pi_j f(X_i|\theta_j)\pi_j f(\widehat{X}_i|\theta_j)\right]^{Z_{ij}} \tag{14.9}$$

Thus, the complete log-likelihood function can be written as:

$$\begin{aligned} L(\mathcal{X}, \widehat{\mathcal{X}}, Z|\Theta) &= \log\left\{\prod_{i=1}^{N}\prod_{j=1}^{K}\left[\pi_j f(X_i|\theta_j)\pi_j f(\widehat{X}_i|\theta_j)\right]^{Z_{ij}}\right\} \\ &= \log\left\{\prod_{i=1}^{N}\prod_{j=1}^{K}\left[\pi_j B(X_i)\pi_j B(\widehat{X}_i)\right]^{Z_{ij}}\right\} \end{aligned} \tag{14.10}$$

where: $B(X_i) = \left\{ \frac{f_{ggd}(X_i|\theta_j)\mathrm{H}(X_i|\Omega_j)}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j)d\mathrm{u}} \right\}$ and $B(\widehat{X}_i) = \left\{ \frac{f_{ggd}(\widehat{X}_i|\theta_j)\mathrm{H}(\widehat{X}_i|\Omega_j)}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j)d\mathrm{u}} \right\}$.

Now, in order to adjust the parameters $\Theta = \{\pi_j, \mu_j, \sigma_j, \lambda_j\}$, we should maximize the likelihood function in Eq. (14.10) which is equivalent to maximize the following:

$$L(\mathcal{X}, \widehat{\mathcal{X}}, Z|\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{K} Z_{ij}\Big(2\log\pi_j + \log f_{ggd}(X_i|\theta_j) + \log f_{ggd}(\widehat{X}_i|\theta_j) - 2\log\int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j) + \log\mathrm{H}(X_i|\Omega_j) + \log\mathrm{H}(\widehat{X}_i|\Omega_j)\Big) \tag{14.11}$$

Using the EM algorithm, the parameters $\Theta = \{\pi_j, \mu_j, \sigma_j, \lambda_j\}$ will be estimated according to two different steps. In E-step, conditional expectation of the complete data log-likelihood is calculated as:

$$\begin{aligned} E[L(\mathcal{X}, \widehat{\mathcal{X}}, Z|\Theta)] &= Q(\mathcal{X}, \widehat{\mathcal{X}}, Z|\Theta) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{K} p(j|X_i, \widehat{X}_i, \theta_j^{(t)})(2\log\pi_j + \log f_{ggd}(X_i|\theta_j) \\ &\quad + \log f_{ggd}(\widehat{X}_i|\theta_j) - 2\log\int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j) \\ &\quad + \log\mathrm{H}(X_i|\Omega_j) + \log\mathrm{H}(\widehat{X}_i|\Omega_j)) \end{aligned} \tag{14.12}$$

where $p(j|X_n, \widehat{X}_n, \theta_j)$ denotes the posterior probability which indicates the probability that $X_i$ and $\widehat{X}_i$ are assigned to cluster $j$:

$$p(j|X_i, \widehat{X}_i, \theta_j) = \frac{\pi_j f(X_i|\theta_j)\pi_j f(\widehat{X}_i|\theta_j)}{\sum_{j'=1}^{K} \pi_{j'} f(X_i|\theta_{j'})\pi_{j'} f(\widehat{X}_i|\theta_{j'})} \tag{14.13}$$

The next step is the maximization (M) step, in which Eq. (14.12) will be maximized which leads to the estimation of parameters of mixture model. In order to present conveniently, we divide this section into four subsections.

#### 14.3.2.1 Mean Parameter Estimation

We consider the derivation of log-likelihood given in Eq. (14.12) with respect to $\mu_j$ at (t+1) iteration step as follows:

$$\frac{\partial Q\left(\mathcal{X},\widehat{\mathcal{X}},Z|\Theta\right)}{\partial\mu_j}=-A(\lambda_j)\frac{\lambda_j}{{\sigma_j}^{\lambda_j}}\sum_{i=1}^{N}p\left(j|X_i,\widehat{X}_i,\theta_j\right)\Big\{|X_i-\mu_j|^{\lambda_j-2}\left(-X_i+\mu_j\right) \tag{14.14}$$

$$+|\widehat{X}_i-\mu_j|^{\lambda_j-2}(-\widehat{X}_i+\mu_j)$$

$$-2\frac{\int_{\partial_j}f_{ggd}(\mathrm{u}|\theta_j)\mathrm{sign}(\mu_j-\mathrm{u})|\mathrm{u}-\mu_j|^{\lambda_j-1}d\mathrm{u}}{\int_{\partial_j}f_{ggd}(\mathrm{u}|\theta_j)d\mathrm{u}}\Big\}$$

By approximating operations, the above given function in Eq. (14.14) is rewritten as:

$$\frac{\partial Q(\mathcal{X},\widehat{\mathcal{X}},Z|\Theta)}{\partial\mu_j}=-A(\lambda_j)\frac{\lambda_j}{{\sigma_j}^{\lambda_j}}\sum_{i=1}^{N}p(j|X_i,\widehat{X}_i,\theta_j)\left|X_i-\mu_j\right|^{\lambda_j-2} \tag{14.15}$$

$$\times\left\{(-X_i+\mu_j)+(-\widehat{X}_i+\mu_j)-\frac{2R_j}{\left|X_i-\mu_j\right|^{\lambda_j-2}}\right\}$$

where

$$R_j=\frac{\sum_{m=1}^{M}\mathrm{sign}({\mu_j}^{(t)}-S_{mj})\left|S_{mj}-{\mu_j}^{(t)}\right|^{{\lambda_j}^{(t)}-1}\mathrm{H}(S_{mj}|\Omega_j)}{\sum_{m=1}^{M}\mathrm{H}(S_{mj}|\Omega_j)} \tag{14.16}$$

The term $S_{mj}\sim f_{ggd}(\mathrm{u}|\theta_j)$ represents the random variables drawn from probability distribution $f_{ggd}(\mathrm{u}|\theta_j)$, and $M$ is number of random variables $S_{mj}$. Now, the solution of $\frac{\partial Q(\mathcal{X},\widehat{\mathcal{X}},Z|\Theta)}{\partial\mu_j}=0$ generates the solutions of $\mu_j$ at (t+1) iteration as follows:

$${\mu_j}^{(t+1)}=\frac{\sum_{i=1}^{N}p(j|X_i,\widehat{X}_i,\theta_j)(\left|X_i-\mu_j\right|^{\lambda_j-2}X_i+\left|\widehat{X}_i-\mu_j\right|^{\lambda_j-2}\widehat{X}_i+2R_j)}{\sum_{i=1}^{N}p(j|X_i,\widehat{X}_i,\theta_j)\left(\left|X_i-\mu_j\right|^{\lambda_j-2}+\left|\widehat{X}_i-\mu_j\right|^{\lambda_j-2}\right)} \tag{14.17}$$

#### 14.3.2.2 Standard Deviation Estimation

Setting the derivative of log-likelihood given in Eq. (14.12) with respect to $\sigma_j$ at (t+1) iteration step, we have:

$$\frac{\partial Q(\mathcal{X}, \widehat{\mathcal{X}}, Z|\Theta)}{\partial \sigma_j} = -{\sigma_j}^{-1} \sum_{i=1}^{N} p\left(j|X_i, \widehat{X}_i, \theta_j\right) \Bigg\{ -2 + A(\lambda_j)\lambda_j {\sigma_j}^{-\lambda_j} \times \left[ |X_i - \mu_j|^{\lambda_j} + |\widehat{X}_i - \mu_j|^{\lambda_j} \right] - 2\frac{\int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j)(-1 + A(\lambda_j)|\mathrm{u} - \mu_j|^{\lambda_j}\lambda_j {\sigma_j}^{-\lambda_j})d\mathrm{u}}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j)d\mathrm{u}} \Bigg\} \tag{14.18}$$

By approximating operations, above given function in Eq.(14.18) is rewritten as:

$$\frac{\partial Q(\mathcal{X}, \widehat{\mathcal{X}}, Z|\Theta)}{\partial \sigma_j} = -{\sigma_j}^{-1} \sum_{i=1}^{N} p(j|X_i, \widehat{X}_i, \theta_j) \times \left\{ -2 + A(\lambda_j)\lambda_j {\sigma_j}^{-\lambda_j} \left[ \left|X_i - \mu_j\right|^{\lambda_j} + \left|\widehat{X}_i - \mu_j\right|^{\lambda_j} \right] - 2G_j \right\} \tag{14.19}$$

where

$$G_j = \frac{\sum_{m=1}^{M} \left( -1 + A\left({\lambda_j}^{(t)}\right) \left|S_{mj} - {\mu_j}^{(t)}\right|^{{\lambda_j}^{(t)}} \left({\sigma_j}^{(t)}\right)^{-{\lambda_j}^{(t)}} \right) \mathrm{H}\left(S_{mj}|\Omega_j\right)}{\sum_{m=1}^{M} \mathrm{H}\left(S_{mj}|\Omega_j\right)} \tag{14.20}$$

Similarly, the solution of $\frac{\partial Q(\mathcal{X}, \widehat{\mathcal{X}}, Z|\Theta)}{\partial \sigma_j} = 0$ generates the solutions of $\sigma_j$ at the (t+1) step as follows:

$${\sigma_j}^{(t+1)} = \left( \frac{\lambda_j A(\lambda_j) \sum_{i=1}^{N} p\left(j|X_i, \widehat{X}_i, \theta_j\right) \left[ \left|X_i - \mu_j\right|^{\lambda_j} + \left|\widehat{X}_i - \mu_j\right|^{\lambda_j} \right]}{2 \sum_{i=1}^{N} p\left(j|X_i, \widehat{X}_i, \theta_j\right) (1 + G_j)} \right)^{1/\lambda_j} \tag{14.21}$$

#### 14.3.2.3 Shape Parameter Estimation

To achieve the estimation of this parameter $\lambda_j$, each iteration requires the calculation of the first and the second derivatives of the function in Eq. (14.12) with respect to the parameter $\lambda_j$ as following:

$$\lambda_j{}^{(t+1)} = \lambda_j{}^{(t)} - \frac{\partial Q(\mathcal{X}, \widehat{\mathcal{X}}, Z|\Theta)}{\partial \lambda_j} \left( \frac{\partial^2 Q(\mathcal{X}, \widehat{\mathcal{X}}, Z|\Theta)}{\partial \lambda_j^2} + \gamma \right)^{-1} \Bigg|_{\lambda_j = \lambda_j{}^{(t)}} \tag{14.22}$$

where $\gamma$ represent a scaling factor. Thus, the derivative of the error function is given by:

$$\frac{\partial Q(\mathcal{X}, \widehat{\mathcal{X}}, Z|\Theta)}{\partial \lambda_j} = -\sum_{i=1}^{N} p(j|X_i, \widehat{X}_i, \theta_j) \times \left\{ f_{ggd}{}'(X_i|\theta_j) + f_{ggd}{}'(\widehat{X}_i|\theta_j) - 2\frac{\int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j) f_{ggd}{}'(\mathrm{u}|\theta_j) d\mathrm{u}}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j) d\mathrm{u}} \right\} \tag{14.23}$$

where

$$f_{ggd}{}'(X|\theta_j) = \frac{\partial f_{ggd}(X|\theta_j)}{\partial \lambda_j} \tag{14.24}$$

The calculation of the terms $\frac{\partial^2 Q(\mathcal{X}, \widehat{\mathcal{X}}, Z|\Theta)}{\partial \lambda_j^2}$ is obtained as:

$$\begin{aligned} \frac{\partial}{\partial \lambda_j} \left( \frac{\partial Q(\mathcal{X}, \widehat{\mathcal{X}}, Z|\Theta)}{\partial \lambda_j} \right) = -\sum_{i=1}^{N} p(j|X_i, \widehat{X}_i, \theta_j) \Bigg\{ & f_{ggd}{}''(X_i|\theta_j) + f_{ggd}{}''(\widehat{X}_i|\theta_j) \\ & + 2\frac{\left( \int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j) f_{ggd}{}'(\mathrm{u}|\theta_j) d\mathrm{u} \right)^2}{(\int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j) d\mathrm{u})^2} \\ & - 2\frac{\int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j)(f_{ggd}{}'(\mathrm{u}|\theta_j))^2 + f_{ggd}{}''(\mathrm{u}|\theta_j) d\mathrm{u}}{\int_{\partial_j} f_{ggd}(\mathrm{u}|\theta_j) d\mathrm{u}} \Bigg\} \end{aligned} \tag{14.25}$$

where

$$f_{ggd}{}''(X|\theta_j) = \frac{\partial f_{ggd}{}'(X|\theta_j)}{\partial \lambda_j} \tag{14.26}$$

The computation of $f_{ggd}{}'(X|\theta_j)$, $f_{ggd}{}''(X|\theta_j)$, $f_{ggd}{}'(\mathrm{u}|\theta_j)$, and $f_{ggd}{}''(\mathrm{u}|\theta_j)$ is followed from [6].

#### 14.3.2.4 Prior Probability Estimation

The updated estimate of the prior probability $\pi_j$ which is positive and sum to one ($\sum_{j=1}^{K} \pi_j = 1$ ) is given as follows:

$$\pi_j^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} p(j|X_i, \widehat{X}_i, \theta_j)^{(t)} \tag{14.27}$$

### *14.3.3 Model Selection Using MDL*

We recall that the maximum likelihood (ML) favors generally for higher values of the number of components, and this issue leads easily to over fitting. Thus, in order to estimate the number of components, we have used the minimum description length (MDL) criterion given by [10, 56]. Therefore, the optimal number of components in the developed mixture model BGGMM is obtained by minimizing the following function:

$$\text{MDL} \approx -\log\ p(X|\Theta) + \frac{1}{2} N_l\ \log(N) \tag{14.28}$$

where $N_l = M(2d+1)$ denotes the number of free parameters in the mixture model.

### *14.3.4 The Proposed Complete Algorithm*

We summarize here the main steps of the proposed algorithm used for the bounded generalized Gaussian mixture model's parameters estimation and model selection.

## 14.4 Experimental Results

### *14.4.1 Experiment Design*

The main goal of this section is to evaluate the performance of the proposed bounded generalized Gaussian mixture model with spatial information and MDL (BGGMM + SI + MDL) as compared to some conventional Gaussian-based methods such as Gaussian-based (GMM), generalized Gaussian-based (GGMM), and bounded Gaussian-based (BGMM). Evaluation is performed on the basis of several real world images. To perform a comparison between image segmentation approaches, different measures are proposed in literature that allow us to determine the quality

***Model learning using EM + SI + MDL***

1. **Input:** Image to be segmented $\mathcal{X}$
2. **Initialization:**
   - Model's parameters $\Theta$ are initialized with the K-means algorithm.
   - Choose an initial high value for K = Kmax (number of regions).
3. **While** ($K \leq K_{max}$) **Do**
4. **Repeat** (Update parameters by alternating the following steps)
   - **Expectation-Step:**
   - Compute the posterior probabilities according to Eq. (14.13).
   - **Maximization-Step:**
   - Update the means $\mu_j$ using Eq. (14.17)
   - Update the standard deviation $\sigma_j$ using Eq. (14.21)
   - Update the parameter $\lambda_j$ using Eq. (14.22)
   - Update the prior distribution $\pi_j$ using Eq. (14.27)

   **While** (algorithm not converging)
   - Calculate the associated MDL criterion using Eq. (14.28).
   - Remove the component j with the smallest $\pi_j$ (K = K - 1)

   **End While**
- Select the optimal model $M^*$ such that: $M^* = \arg\min_M MDL(M)$. Then, return the model parameters with the optimal model.
- The segmented image is determined according to the optimal model.

of the segmentation results according to one or more reference segmentation. In our case, we propose to use the following measures: sensitivity, specificity, accuracy, recall, F1-measure, and MCC (Matthews correlation coefficient). They are often used in the context of image segmentation and classification in order to quantify the quality of the segmentation result.

- Sensitivity = $\frac{TP}{TP+FN}$
- Specificity = $\frac{TN}{TN+FP}$
- Precision = $\frac{TP}{TP+FP}$
- Accuracy = $\frac{TP+TN}{TP+FP+TN+FN}$
- Recall = $\frac{TP}{TP+FN}$
- F1-measure = $\frac{2*TP}{(2*TP+FP+FN)}$
- MCC = $\frac{TP*TN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}}$

### *14.4.2 Experiment 1: Real World Image Segmentation*

We start by evaluating the performance of our proposed approach using various examples from real world images which are publicly provided by the Berkeley segmentation dataset (BSD) [42]. The Berkeley benchmark is a public database consists of 300 images of a wide variety of natural scenes. Ground truth segmentations of these images are also provided. Quantitative performances are obtained based on the ground truth and using the accuracy, the precision, and the boundary displacement error (BDE) metrics. The latter metric measures the average displacement error of one boundary pixels and its closest boundary pixels in the other segmentation [27]. Figure 14.2 shows obtained results for some samples chosen randomly from the BSD dataset. A comparative study between different models is depicted in Tables 14.1 and 14.2. According to these results, it is clear that our approach denoted by (BGGMM + SI + MDL) outperforms other methods. Indeed, the accuracy value is about 92.9% for BGGMM + SI + MDL against 88.6% for GGMM and 84.3% for GMM. In addition, the minimum boundary displacement error is found with our method. It should be emphasized also that more accurate number of clusters (or regions) are obtained with our model thanks to the integration of the spatial

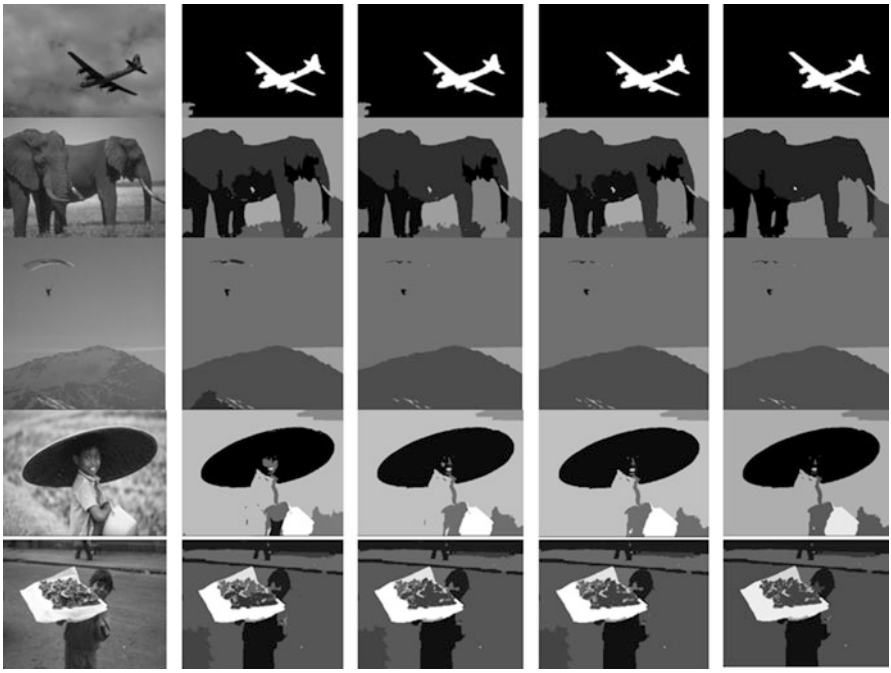


**Fig. 14.2** Image segmentation results of some samples selected from the BSD dataset. First column: Original Image, second, third fourth, and fifth columns correspond to results obtained using GMM, GGMM, BGMM, and BGGMM+SI+MDL respectively

**Table 14.1** Results obtained for five different images chosen randomly from the BSD dataset (see Fig. 14.2)

| Method | Precision | Accuracy | BDE | M | $\hat{M}$ |
|---|---|---|---|---|---|
| GMM | 88.70 | 90.11 | 0.241 | 3 | 2 |
| GGMM | 91.23 | 92.58 | 0.234 | 3 | |
| BGMM | 90.21 | 91.13 | 0.229 | 3 | |
| BGGMM+SI+MDL | 95.97 | 94.43 | 0.109 | 2 | |
| GMM | 78.21 | 77.53 | 0.321 | 5 | 3 |
| GGMM | 86.54 | 82.36 | 0.280 | 5 | |
| BGMM | 88.08 | 83.48 | 0.266 | 5 | |
| BGGMM+SI+MDL | 91.36 | 89.76 | 0.237 | 4 | |
| GMM | 76.10 | 90.13 | 0.296 | 6 | 5 |
| GGMM | 88.61 | 94.90 | 0.209 | 6 | |
| BGMM | 88.96 | 95.01 | 0.213 | 6 | |
| BGGMM+SI+MDL | 89.88 | 95.30 | 0.183 | 4 | |
| GMM | 82.16 | 78.14 | 0.344 | 7 | 5 |
| GGMM | 85.34 | 81.02 | 0.281 | 6 | |
| BGMM | 86.29 | 81.97 | 0.274 | 6 | |
| BGGMM+SI+MDL | 89.20 | 82.34 | 0.246 | 5 | |
| GMM | 79.76 | 80.07 | 0.401 | 14 | 10 |
| GGMM | 85.31 | 83.22 | 0.385 | 13 | |
| BGMM | 84.90 | 81.47 | 0.391 | 13 | |
| BGGMM+SI+MDL | 91.19 | 87.21 | 0.291 | 11 | |

$M$ and $\hat{M}$ denote the obtained and the real number of regions, respectively

**Table 14.2** Average metrics for all images in the BSD dataset produced by the algorithms: GMM, GGMM, BGMM, and BGGMM+SI+MDL

| Method | Precision | Accuracy | BDE |
|---|---|---|---|
| GMM | 81.39 | 84.30 | 31.06 |
| GGMM | 86.23 | 88.64 | 28.67 |
| BGMM | 87.20 | 87.62 | 27.04 |
| BGGMM+SI+MDL | 91.33 | 92.96 | 20.39 |

information and the MDL criterion into the learning model (see two last columns in Table 14.1). In other word, these constraints help to avoid over-segmentation and provide more accurate segmentation results.

### *14.4.3 Experiment 2: MR Brain Images Segmentation*

Precise brain tissues segmentation in magnetic resonance (MR) images has been the topic of extensive research in the past and is a crucial step in several applications

such as surgical and radiotherapy planning, analysis of neuroanatomical variability, image guided therapy, abnormality detection, and many other for medical studies [2, 14, 53, 58, 64]. In particular, some results are presented for segmenting the white matter (WM), gray matter (GM), cerebrospinal fluid (CSF), and modeling the cortex anatomy. Various promising works have been proposed that offering a diversity of methods such as atlas-based methods [4, 7, 15, 29, 31, 55, 63, 69], variational approaches [28, 32, 33, 61, 65, 72], and pattern classification-based techniques [1, 3, 4, 28, 44, 55, 66]. Several methods suppose that the anatomical tissue intensity can be modeled as a mixture of Gaussians. Unfortunately, this assumption leads to erroneous results since these tissues have overlapping spectral properties. Despite the good results obtained in the literature; the achievement of MRI brain segmentation task has proven problematic due to the poor contrast, the inhomogeneity, and the unknown noise. In this section, we experimentally evaluate our proposed model on real MR brain images provided by the Internet Brain Segmentation Repository (IBSR) database. These images and their manual segmentations are provided and publicly available[1] by the Center for Morphometric Analysis at Massachusetts General Hospital. The IBSR provided 20 real T1-weighted coronal MRI scans (3D volumes) of normal subjects with gray/white/cerebral-spinal-fluid/other expert segmentations. Each volume is of size 63 scans, and each scan is of size $256 \times 256$ pixels. In our study, we propose to identify the three main structures: white matter (WM), gray matter (GM), and cerebrospinal fluid (CSF). Thus, we choose as initial value for the number of classes K = 4 (WM, GM, CSF, other). If a pixel does not belong to WM nor to GM or CSF, it will be affected to the cluster "other". We note also that in some scans the CSF does not appear in the image, so the number of classes is automatically reduced to 3 instead of 4 and all pixels in this image will be assigned to WM or to GM or to "other" component. Some samples of original images (chosen randomly from the dataset IBSR) and their segmentation obtained by our method (BGGMM+SI+MDL), and other algorithms GGMM and GMM are depicted in Fig. 14.3. Quantitative measures are also determined on the basis of the above metrics (presented in the previous section) and given in Table 14.3 for different methods. Under the assumption cited in [73], the accuracy value gives a score of one for perfect agreement and 0 for complete disagreement, and any accuracy value above 0.7 indicates a strong agreement. According to this study, it is clear that our model is capable to provide strongly acceptable results compared to the ground truth and also it is able to provide better results w.r.t the rest of methods (GGMM and GMM). On the other hand, values obtained with GGMM are more better than GMM's values. Sometimes, the results provided by our model are equal or slightly higher than the other two models. We interpret and justify this result by the fact that only the value of the grayscale is considered as information and no more other features are taken into account in the segmentation process. Thereby, more relevant features are definitely needed to improve the expected results for many cases especially when the image is blurred and highly textured.

[1] http://www.cma.mgh.harvard.edu/ibsr/.

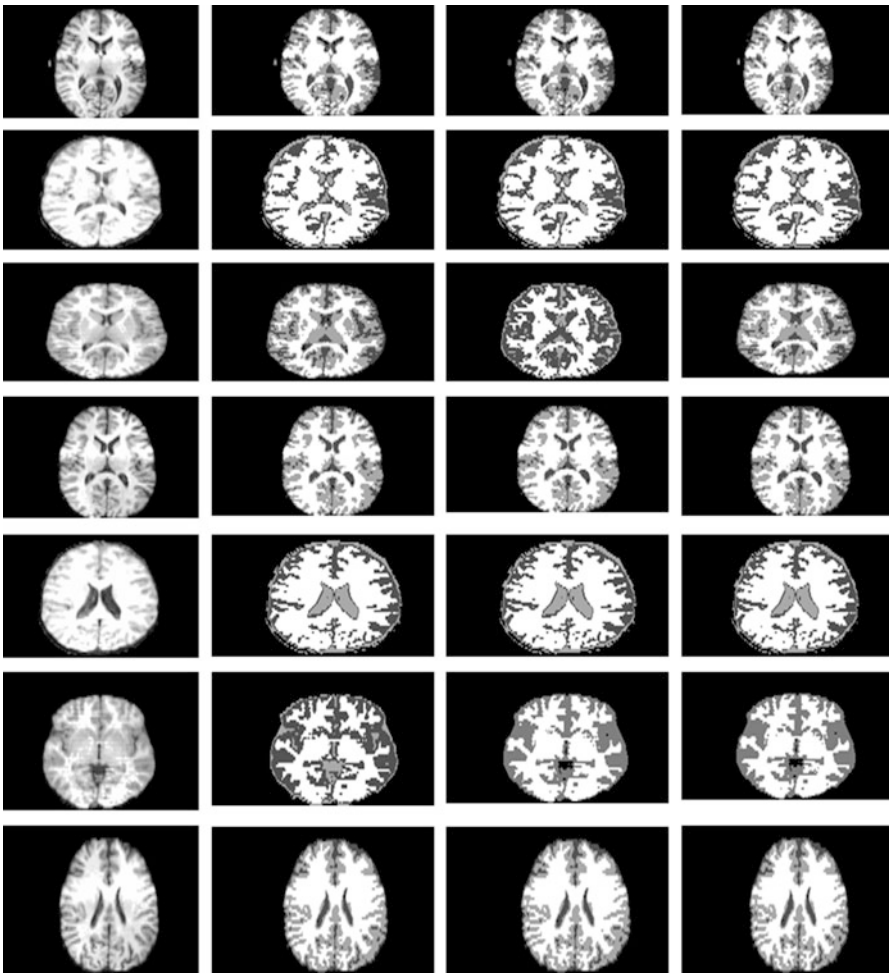


**Fig. 14.3** Segmenting main tissues in MRI images (GM, WM, CSF) for some scans from IBSR. First column: original image. Second, third, fourth, and fifth are the results with GMM, GGMM, BGGMM+SI+MDL, respectively

**Table 14.3** Average metrics for MRI brain images (IBSR dataset) segmentation produced by different algorithms: GMM, GGMM, and BGGMM+SI+MDL

| Method | Sensitivity | Specificity | Precision | Recall | Accuracy | F1 | MCC |
|---|---|---|---|---|---|---|---|
| GMM | 73.89 | 95.65 | 62.79 | 73.89 | 92.87 | 63.15 | 62.01 |
| GGMM | 75.76 | 95.79 | 63.83 | 75.76 | 93.29 | 64.81 | 63.50 |
| BGGMM +SI+MDL | 76.94 | 95.95 | 65.90 | 76.94 | 93.44 | 66.35 | 65.57 |

### 14.4.4 Experiment 3: Left Ventricle (LV) Segmentation in a Sequence of Images

Accurate segmentation of the left ventricle (LV) in cardiac magnetic resonance imaging sequences is another important application to analyze the cardiac function, to assess the myocardial mass, the stroke volume, and the ejection fraction. The left ventricle in short axis (SAX) cine MR image looks like a circular and appear bright, and all their surrounding organs are dark (i.e., lung, myocardium, and liver), as shown in Fig. 14.4. Manual segmentation is particularly impractical, non-reproducible, and time-consuming task for cardiac radiologists, so a fully automatic accurate segmentation of left ventricle is highly required and still attracting research. Nevertheless, this task faces many challenges due to the LV shape variability, the overlap between the intensity distributions, the low contrast between the myocardium and surrounding tissues, etc. In recent years, quite a number of techniques have been proposed for cardiac segmentation including: image-driven method [21], multidimensional dynamic programming [62], EM + probabilistic atlas [40], variational approaches [8, 26, 41, 51], pyramid and fuzzy clustering [54], model-based graph cut [37].

In this work, we assess the performance of our algorithm on the basis of the dataset of cardiac cine MRI images provided by the clinical database of Sunnybrook Health Sciences Centre [52] and published on the Internet.[2] It consists of 45 cases containing 12 heart failure with ischemia (HF-I) cases, 12 heart failure without ischemia (HF-NI) cases, 12 hypertrophy (HYP) cases, and nine normal (N) cases. In this experiment, we focus on identifying the ROI corresponding to the LV. To this end, we start by locating the region of interest (ROI) in the current image which is performed by selecting a rectangular that englobes the LV organ. Then, we applied the developed method to refine the final segmentation. Subsequently, the output of

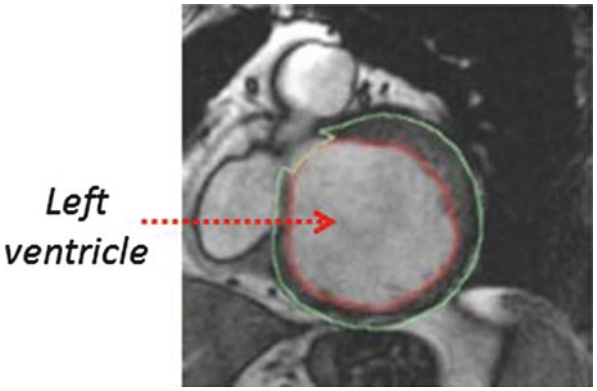



**Fig. 14.4** Illustration of a part of cardiac MR image

[2] http://sourceforge.net/projects/cardiac-mr/files.

**Table 14.4** Average metrics for Left ventricle cardiac images segmentation produced by the algorithms: GMM, GGMM and BGGMM+SI+MDL

| Method | Sensitivity | Specificity | Precision | Recall | Accuracy | F1 | MCC |
|---|---|---|---|---|---|---|---|
| GMM | 40.72 | 99.76 | 96.24 | 40.72 | 97.94 | 53.83 | 59.85 |
| GGMM | 41.73 | 99.76 | 96.26 | 41.73 | 97.96 | 55.14 | 60.90 |
| BGGMM +SI+MLL | 42.72 | 99.77 | 96.34 | 42.72 | 98.00 | 56.26 | 61.78 |

the current segmentation is used as initialization step for the following image in the sequence and so that. Some illustrations of the obtained results with GMM, GGMM, and BGGMM+SI+MDL are given in Fig. 14.5. We evaluate also all obtained results w.r.t the two other methods on the basis of the metrics described above. The average values for the dataset are depicted in Table 14.4. According to these results, we can conclude that our proposed algorithm outperforms the rest of methods. This conclusion confirms again that the integration of both spatial information and MDL criterion in the new developed mixture model based on the bounded generalized Gaussian distribution leads to more smooth and precise results.

## 14.5 Summary

In this paper, we have presented a new flexible mixture model based on the bounded generalized Gaussian distribution for data modeling and image segmentation. In order to increase the accuracy of the expected results and to estimate accurately the appropriate number of components, the proposed method integrates both a spatial information (a prior knowledge) and a minimum description length criterion (MDL). We evaluated also the performance of the proposed model based on different challenging applications such as the segmentation of MRI brain images and the cardiac left ventricle. Obtained results show the merits of our proposed framework which outperforms conventional Gaussian-based models. Future works are devoted to the involving of more relevant visual features (shape, texture, color, etc.) in the whole segmentation process to better characterizing the ROI and to improve results. Moreover, we plan to use an enhanced extension of the EM algorithm which is the ECM method [12] to avoid some problems related to EM.

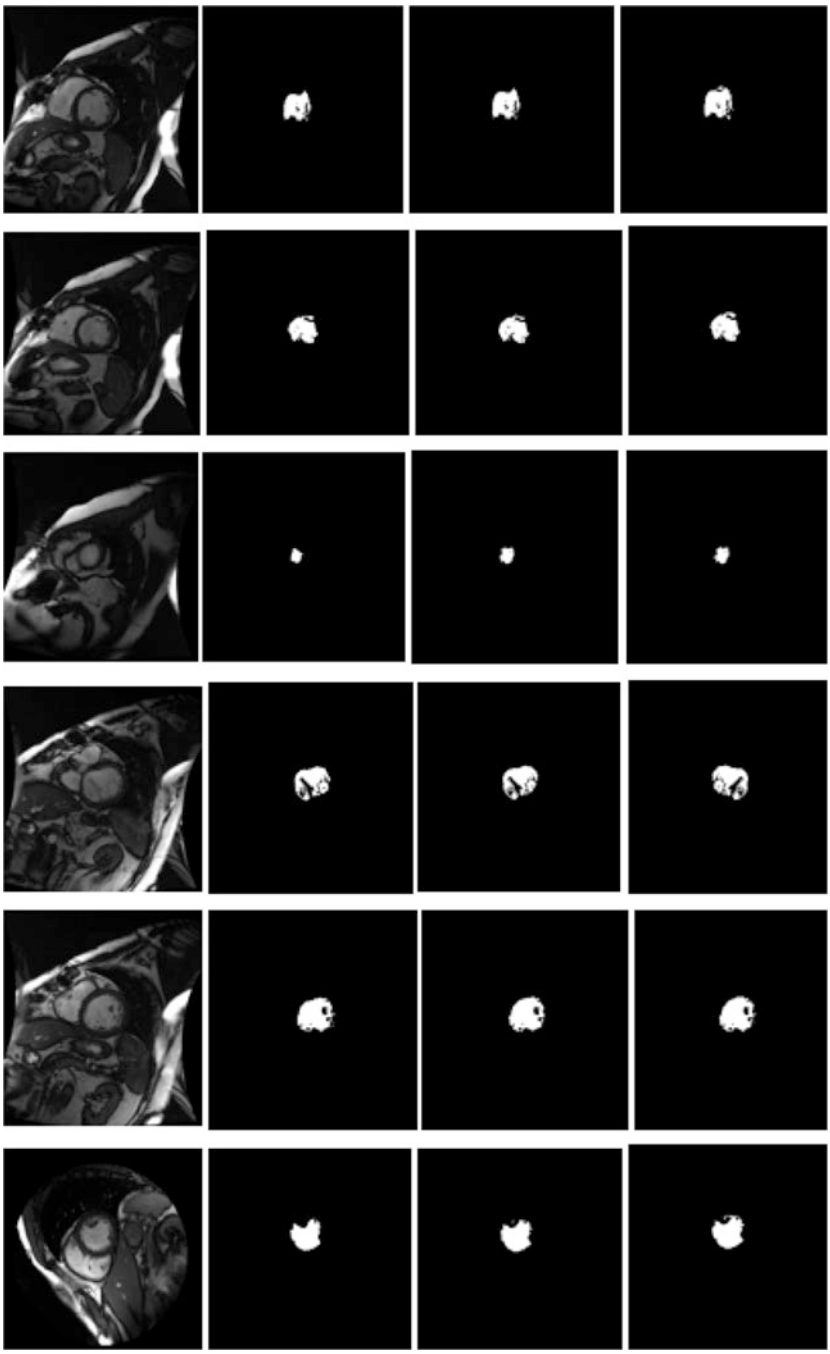


**Fig. 14.5** Segmenting the Left ventricle cardiac on some scans from the dataset in [52]. First column: Original image. Second, third, fourth, and fifth are the results with GMM, GGMM, BGGMM+SI+MDL, respectively

**Acknowledgements** This research is based on a grant received from the research council (TCR)-Oman.

## References

1. Abbasi, S., Tajeripour, F.: Detection of brain tumor in 3d MRI images using local binary patterns and histogram orientation gradient. Neurocomputing **219**, 526–535 (2017)
2. Agnello, L., Comelli, A., Ardizzone, E., Vitabile, S.: Unsupervised tissue classification of brain MR images for voxel-based morphometry analysis. Int. J. Imaging Syst. Technol. **26**(2), 136–150 (2016)
3. Ahmadvand, A., Daliri, M.R.: Improving the runtime of MRF based method for MRI brain segmentation. Appl. Math. Comput. **256**, 808–818 (2015)
4. Al-Shaikhli, S.D.S., Yang, M.Y., Rosenhahn, B.: Multi-region labeling and segmentation using a graph topology prior and atlas information in brain images. Comput. Med. Imaging Graph. **38**(8), 725–734 (2014)
5. Alhakami, W., ALharbi, A., Bourouis, S., Alroobaea, R., Bouguila, N.: Network anomaly intrusion detection using a nonparametric Bayesian approach and feature selection. IEEE Access **7**, 52181–52190 (2019)
6. Allili, M.S., Bouguila, N., Ziou, D.: Finite general Gaussian mixture modeling and application to image and video foreground segmentation. J. Electron. Imaging **17**(1), 013005–013005 (2008)
7. Asman, A.J., Landman, B.A.: Non-local statistical label fusion for multi-atlas segmentation. Med. Image Anal. **17**(2), 194–208 (2013)
8. Ayed, I.B., Li, S., Ross, I.: Embedding overlap priors in variational left ventricle tracking. IEEE Trans. Med. Imaging **28**(12), 1902–1913 (2009)
9. Bishop, C.M.: Pattern recognition. Mach. Learn. **128** (2006)
10. Bouguila, N.: Clustering of count data using generalized Dirichlet multinomial distributions. IEEE Trans. Knowl. Data Eng. **20**(4), 462–474 (2008)
11. Bouguila, N., ElGuebaly, W.: Integrating spatial and color information in images using a statistical framework. Expert Syst. Appl. **37**(2), 1542–1549 (2010)
12. Bouguila, N., Ziou, D.: On fitting finite Dirichlet mixture using ECM and MML. In: Pattern Recognition and Data Mining, Third International Conference on Advances in Pattern Recognition, ICAPR 2005, Bath, UK, August 22–25, 2005, Proceedings, Part I. pp. 172–182 (2005)
13. Bouguila, N., Ziou, D.: A probabilistic approach for shadows modeling and detection. In: IEEE International Conference on Image Processing 2005, vol. 1, pp. I–329. IEEE, Piscataway (2005)
14. Bourouis, S., Hamrouni, K.: 3D segmentation of MRI brain using level set and unsupervised classification. Int. J. Image Graph. **10**(1), 135–154 (2010)
15. Bourouis, S., Hamrouni, K., Betrouni, N.: Automatic MRI brain segmentation with combined atlas-based classification and level-set approach. In: Image Analysis and Recognition, 5th International Conference, ICIAR. pp. 770–778 (2008)
16. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: Image segmentation using expectation-maximization and its application to image querying. IEEE Trans. Pattern Anal. Mach. Intell. **24**(8), 1026–1038 (2002)
17. Channoufi, I., Bourouis, S., Bouguila, N., Hamrouni, K.: Color image segmentation with bounded generalized Gaussian mixture model and feature selection. In: 4th International Conference on Advanced Technologies for Signal and Image Processing, ATSIP 2018, Sousse, Tunisia, March 21–24, 2018. pp. 1–6 (2018)
18. Channoufi, I., Bourouis, S., Bouguila, N., Hamrouni, K.: Image and video denoising by combining unsupervised bounded generalized Gaussian mixture modeling and spatial information. Multimed. Tools Appl. **77**(19), 25591–25606 (2018)

19. Channoufi, I., Bourouis, S., Bouguila, N., Hamrouni, K.: Spatially constrained mixture model with feature selection for image and video segmentation. In: Image and Signal Processing-8th International Conference, ICISP 2018, Cherbourg, France, July 2–4, 2018, Proceedings. pp. 36–44 (2018)
20. Choy, S.K., Tong, C.S.: Statistical wavelet subband characterization based on generalized gamma density and its application in texture retrieval. IEEE Trans. Image Process. **19**(2), 281–289 (2010)
21. Cocosco, C., Niessen, W., Netsch, T., Vonken, E., Lund, G., Stork, A., Viergever, M.: Automatic image-driven segmentation of the ventricles in cardiac cine MRI. J. Magn. Reson. Imaging **25**(2), 366–374 (2008)
22. Cohen, F.S., Cooper, D.B.: Simple parallel hierarchical and relaxation algorithms for segmenting noncausal Markovian random fields. IEEE Trans. Pattern Anal. Mach. Intell. **2**, 195–219 (1987)
23. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B Methodol. **39**(1), 1–38 (1977)
24. Do, M.N., Vetterli, M.: Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. IEEE Trans. Image Process. **11**(2), 146–158 (2002)
25. Fan, W., Sallay, H., Bouguila, N., Bourouis, S.: A hierarchical Dirichlet process mixture of generalized Dirichlet distributions for feature selection. Comput. Electr. Eng. **43**, 48–65 (2015)
26. Fradkin, M., Ciofolo, C., Mory, B., Hautvast, G., Breeuwer, M.: Comprehensive segmentation of cine cardiac MR images. Med. Image Comput. Comput. Assist. Interv. **11**, 178–185 (2008)
27. Freixenet, J., Muñoz, X., Raba, D., Martí, J., Cufí, X.: Yet another survey on image segmentation: region and boundary information integration. Comput. Vis. ECCV **2002**, 21–25 (2002)
28. Gao, G., Wen, C., Wang, H.: Fast and robust image segmentation with active contours and student's-t mixture model. Pattern Recogn. **63**, 71–86 (2017)
29. Gass, T., Székely, G., Goksel, O.: Simultaneous segmentation and multiresolution nonrigid atlas registration. IEEE Trans. Image Process. **23**(7), 2931–2943 (2014)
30. Hung, W.L., Yang, M.S., Chen, D.H.: Bootstrapping approach to feature-weight selection in fuzzy c-means algorithms with an application in color image segmentation. Pattern Recogn. Lett. **29**(9), 1317–1325 (2008)
31. Iglesias, J.E., Sabuncu, M.R.: Multi-atlas segmentation of biomedical images: A survey. Med. Image Anal. **24**(1), 205–219 (2015)
32. Ilunga-Mbuyamba, E., Avina-Cervantes, J.G., Garcia-Perez, A., de Jesus Romero-Troncoso, R., Aguirre-Ramos, H., Cruz-Aceves, I., Chalopin, C.: Localized active contour model with background intensity compensation applied on automatic MR brain tumor segmentation. Neurocomputing **220**, 84–97 (2017)
33. Ivanovska, T., Laqua, R., Wang, L., Schenk, A., Yoon, J.H., Hegenscheid, K., Völzke, H., Liebscher, V.: An efficient level set method for simultaneous intensity inhomogeneity correction and segmentation of MR images. Comput. Med. Imaging Graph. **48**, 9–20 (2016)
34. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 4–37 (2000)
35. Kelly, P.A., Derin, H., Hartt, K.D.: Adaptive segmentation of speckled images using a hierarchical random field model. IEEE Trans. Acoust. Speech Signal Process. **36**(10), 1628–1641 (1988)
36. Law, M.H., Figueiredo, M.A., Jain, A.K.: Simultaneous feature selection and clustering using mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **26**(9), 1154–1166 (2004)
37. Lin, X., Cowan, B., Young, A.: Model-based graph cut method for segmentation of the left ventricle. Conf. Proc. IEEE Eng. Med. Biol. Socpp. **3**, 3059–3062 (2005)
38. Lindblom, J., Samuelsson, J.: Bounded support gaussian mixture modeling of speech spectra. IEEE Trans. Speech and Audio Process. **11**(1), 88–99 (2003)
39. Liu, G., Wu, J., Zhou, S.: Probabilistic classifiers with a generalized Gaussian scale mixture prior. Pattern Recogn. **46**(1), 332–345 (2013)

40. Lorenzo-Valdés, M., Sanchez-Ortiz, G.I., Elkington, A., Mohiaddin, R., Rueckert, D.: Segmentation of 4d cardiac MR images using a probabilistic atlas and the EM algorithm. Med. Image Anal. **8**(3), 255–265 (2004)
41. Lynch, M., Ghita, O., Whelan, P.F.: Segmentation of the left ventricle of the heart in 3-d+t MRI data using an optimized nonrigid temporal model. IEEE Trans. Med. Imaging **27**(2), 195–203 (2008)
42. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, vol. 2. pp. 416–423, IEEE, Piscataway (2001)
43. McLachlan, G., Peel, D.: Finite Mixture Models. John, New York (2004)
44. Menze, B.H., Leemput, K.V., Lashkari, D., Riklin-Raviv, T., Geremia, E., Alberts, E., Gruber, P., Wegener, S., Weber, M.A., Székely, G., Ayache, N., Golland, P.: A generative probabilistic model and discriminative extensions for brain lesion segmentation with application to tumor and stroke. IEEE Trans. Med. Imaging **35**(4), 933–946 (2016)
45. Najar, F., Bourouis, S., Bouguila, N., Belghith, S.: A comparison between different Gaussian-based mixture models. In: 14th IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2017, Hammamet, Tunisia, October 30–Nov. 3, 2017. pp. 704–708 (2017)
46. Najar, F., Bourouis, S., Bouguila, N., Belghith, S.: A fixed-point estimation algorithm for learning the multivariate GGMM: application to human action recognition. In: 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE). pp. 1–4 (2018)
47. Najar, F., Bourouis, S., Zaguia, A., Bouguila, N., Belghith, S.: Unsupervised human action categorization using a Riemannian averaged fixed-point learning of multivariate GGMM. In: Unsupervised learning of finite full covariance multivariate generalized Gaussian mixture models for human activity recognition of Image Analysis and Recognition-15th International Conference, ICIAR 2018, Póvoa de Varzim, Portugal, June 27–29, 2018. pp. 408–415 (2018)
48. Najar, F., Bourouis, S., Bouguila, N., Belghith, S.: Unsupervised learning of finite full covariance multivariate generalized Gaussian mixture models for human activity recognition. Multimed. Tools Appl. 1–23 (2019)
49. Nguyen, T.M., Wu, Q.J., Zhang, H.: Bounded generalized gaussian mixture model. Pattern Recogn. **47**(9), 3132–3142 (2014)
50. Ozden, M., Polat, E.: A color image segmentation approach for content-based image retrieval. Pattern Recogn. **40**(4), 1318–1325 (2007)
51. Paragios, N.: A level set approach for shape-driven segmentation and tracking of the left ventricle. IEEE Trans. Med. Imaging **22**(6), 773–776 (2003)
52. Radau, P., Lu, Y., Connelly, K., Paul, G., Dick, A., Wright, G.: Evaluation framework for algorithms segmenting short axis cardiac MRI. In: The MIDAS Journal-Cardiac MR Left Ventricle Segmentation Challenge (2009)
53. Rajalakshmi, N., Lakshmi Prabha, V.: Mri brain image classification—a hybrid approach. Int. J. Imaging Syst. Technol. **25**(3), 226–244 (2015)
54. Rezaee, M.R., van der Zwet, P.M.J., Lelieveldt, B.P.E., van der Geest, R.J., Reiber, J.H.C.: A multiresolution image segmentation technique based on pyramidal segmentation and fuzzy clustering. IEEE Trans. Image Process. **9**(7), 1238–1248 (2000)
55. Ribbens, A., Hermans, J., Maes, F., Vandermeulen, D., Suetens, P.: Unsupervised segmentation, clustering, and groupwise registration of heterogeneous populations of brain MR images. IEEE Trans. Med. Imaging **33**(2), 201–224 (2014)
56. Rissanen, J.: Modeling by shortest data description. Automatica **14**(5), 465–471 (1978)
57. Roth, V., Ommer, B.: Exploiting low-level image segmentation for object recognition. In: Joint Pattern Recognition Symposium. pp. 11–20, Springer, Berlin (2006)
58. Saritha, S., Amutha Prabha, N.: A comprehensive review: Segmentation of MRI images—brain tumor. Int. J. Imaging Syst. Technol. **26**(4), 295–304 (2016)
59. Sefidpour, A., Bouguila, N.: Spatial color image segmentation based on finite non-Gaussian mixture models. Expert Syst. Appl. **39**(10), 8993–9001 (2012)

60. Tenenbaum, J.M., Barrow, H.G.: Experiments in interpretation-guided segmentation. Artif. Intell. **8**(3), 241–274 (1977)
61. Thapaliya, K., Pyun, J.Y., Park, C.S., Kwon, G.R.: Level set method with automatic selective local statistics for brain tumor segmentation in MR images. Comput. Med. Imaging Graph. **37**(7–8), 522–537 (2013)
62. Uzümcü, M., van der Geest, R., Swingen, C., Reiber, J., Lelieveldt, B.: Time continuous tracking and segmentation of cardiovascular magnetic resonance images using multidimensional dynamic programming. Invest. Radiol. **41**(1), 52–62 (2006)
63. van der Lijn, F., de Bruijne, M., Klein, S., den Heijer, T., Hoogendam, Y.Y., van der Lugt, A., Breteler, M.M.B., Niessen, W.J.: Automated brain structure segmentation based on atlas registration and appearance models. IEEE Trans. Med. Imaging **31**(2), 276–286 (2012)
64. Verma, H., Agrawal, R.K., Kumar, N.: Improved fuzzy entropy clustering algorithm for MRI brain image segmentation. Int. J. Imaging Syst. Technol. **24**(4), 277–283 (2014)
65. Wang, L., Pan, C.: Image-guided regularization level set evolution for MR image segmentation and bias field correction. Magn. Reson. Imaging **32**(1), 71–83 (2014)
66. Xia, Y., Ji, Z., Zhang, Y.: Brain MRI image segmentation based on learning local variational Gaussian mixture models. Neurocomputing **204**, 189–197 (2016), big Learning in Social Media Analytics Containing a selection of papers from the 2014 International Conference on Security, Pattern Analysis, and Cybernetics (ICSPAC2014)
67. Yang, X., Krishnan, S.M.: Image segmentation using finite mixtures and spatial information. Image Vis. Comput. **22**(9), 735–745 (2004)
68. Yang, M.S., Lai, C.Y., Lin, C.Y.: A robust EM clustering algorithm for gaussian mixture models. Pattern Recogn. **45**(11), 3950–3961 (2012)
69. Yousefi, S., Kehtarnavaz, N., Gholipour, A.: Improved labeling of subcortical brain structures in atlas-based segmentation of magnetic resonance images. IEEE Trans. Biomed. Eng. **59**(7), 1808–1817 (2012)
70. Zhang, Y., Brady, M., Smith, S.: Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE Trans. Med. Imaging **20**(1), 45–57 (2001)
71. Zhang, Z., Chen, C., Sun, J., Chan, K.L.: EM algorithms for gaussian mixtures with split-and-merge operation. Pattern Recogn. **36**(9), 1973–1983 (2003)
72. Zhou, S., Wang, J., Zhang, M., Cai, Q., Gong, Y.: Correntropy-based level set method for medical image segmentation and bias correction. Neurocomputing **234**, 216–229 (2017)
73. Zijdenbos, A.P., Dawant, B.M., Margolin, R.A., Palmer, A.C.: Morphometric analysis of white matter lesions in MR images: method and validation. IEEE Trans. Med. Imaging **13**(4), 716–724 (1994)
74. Ziou, D., Bouguila, N.: Unsupervised learning of a finite gamma mixture using MML: application to SAR image analysis. In: ICPR (2). pp. 68–71 (2004)

# Index





N. Bouguila, W. Fan (eds.), *Mixture Models and Applications*, Unsupervised and Semi-supervised Learning, https://doi.org/10.1007/978-3-030-23876-6