Positive Data Clustering based on Generalized Inverted Dirichlet Mixture Model

Mohamed Al Mashrgy

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of DOCTOR OF PHILOSOPHY (Electrical & Computer Engineering) at

Concordia University

Montréal, Québec, Canada

May 2015

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Mohamed Al Mashrgy**

Entitled: **Positive Data Clustering based on Generalized Inverted Dirichlet Mixture Model**

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

DOCTOR OF PHILOSOPHY (Electrical & Computer Engineering)

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

Signed by the final examining committee:

Dr. Mingyuan Chen ______ Chair

Dr. Abir Jaafar Hussain ______ External Examiner

Dr. Lyes Kadem ______ External to Program

Dr. A. Ben Hamza ______ Examiner

Dr. Jamal Bentahar ______ Examiner

Dr. Nizar Bouguila ______ Thesis Supervisor

Approved by ______

Chair of Department or Graduate Program Director

______

Dean of Faculty of Engineering & Computer Science

ABSTRACT

Positive Data Clustering based on Generalized Inverted Dirichlet Mixture Model

Mohamed Al Mashrgy, Ph.D.

Concordia University, 2015

Recent advances in processing and networking capabilities of computers have caused an accumulation of immense amounts of multimodal multimedia data (image, text, video). These data are generally presented as high-dimensional vectors of features. The availability of these high-dimensional data sets has provided the input to a large variety of statistical learning applications including clustering, classification, feature selection, outlier detection and density estimation. In this context, a finite mixture offers a formal approach to clustering and a powerful tool to tackle the problem of data modeling. A mixture model assumes that the data is generated by a set of parametric probability distributions. The main learning process of a mixture model consists of the following two parts: parameter estimation and model selection (estimation the number of components). In addition, other issues may be considered during the learning process of mixture models such as the: a) feature selection and b) outlier detection. The main objective of this thesis is to work with different kinds of estimation criteria and to incorporate those challenges into a single framework.

The first contribution of this thesis is to propose a statistical framework which can tackle the problem of parameter estimation, model selection, feature selection, and outlier rejection in a unified model. We propose to use feature saliency and introduce an expectation-maximization (EM) algorithm for the estimation of the Generalized Inverted Dirichlet (GID) mixture model. By using the Minimum Message Length (MML), we can identify how much each feature contributes to our model as well as determine the number of components. The presence of outliers is an added challenge and is handled by incorporating an auxiliary outlier component, to which we associate a uniform density. Experimental results on synthetic data, as well as real world applications involving visual scenes and object classification, indicates that the proposed approach was promising, even though low-dimensional representation of the data was applied. In addition, it showed the importance of embedding an outlier component to the proposed model. EM learning suffers from significant drawbacks. In order to overcome those drawbacks, a learning approach using a Bayesian framework is proposed as our second contribution. This learning is based on the estimation of the parameters posteriors and by considering the prior knowledge about these parameters. Calculation of the posterior distribution of each parameter in the model is done by using Markov chain Monte Carlo (MCMC) simulation methods - namely, the Gibbs sampling and the Metropolis-Hastings methods. The Bayesian Information Criterion (BIC) was used for model selection. The proposed model was validated on object classification and forgery detection applications. For the first two contributions, we developed a finite GID mixture. However, in the third contribution, we propose an infinite GID mixture model. The proposed model simutaneously tackles the clustering and feature selection problems. The proposed learning model is based on Gibbs sampling. The effectiveness of the proposed method is shown using image categorization application. Our last contribution in this thesis is another fully Bayesian approach for a finite GID mixture learning model using the Reversible Jump Markov Chain Monte Carlo (RJMCMC) technique. The proposed algorithm allows for the simultaneously handling of the model selection and parameter estimation for high dimensional data. The merits of this approach are investigated using synthetic data, and data generated from a challenging namely object detection.

ACKNOWLEDGEMENTS

I would like to express my special appreciation and thanks to my advisor Dr. Nizar Bouguila, you have been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. Your advise on both research as well as on my career have been priceless.

I would also like to thank Dr. Abir Jaafar Hussain, Dr. Lyes Kadem, Dr. Jamal Bentahar, and Dr. A. Ben Hamza for serving as my committee members and for letting my defense be an enjoyable moment. Your brilliant comments and suggestions have helped me to considerably improve my research work.

This thesis was funded by the Ministry of Higher Education and Scientific Research in Libya through the Libyan - North American Scholarship Program. I would like to take this opportunity to thank them for their financial support. Moreover, I am grateful for all research facilities that have been provided to me at Concordia University to carry out this work.

A special thanks to my family back in Libya. Words can not express how grateful I am to my dear mother Muna Zaltum, uncle Dr. Mohamed Almeshragi, brother, sisters, uncles, mother-in-law, nephews, nieces and cousins for all of the sacrifices that you have made on my behalf. Your prayers for me was what sustained me thus far.

I would also like to thank all of my friends and lab-mates, including but not limited to, Taoufik Bdiri and Abdulmotaleb Zidan, with whom I have enjoyed my journey to accomplish this thesis.

At the last and not least, I would like to express my deep appreciation to my beloved wife,

Dr. Muna, and wonderful children Anas, Muheeb, and Awss, who are the most precious gift that I have received in this life and who gave me all the energy to carry on during the most difficult moments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AIC	Akaike’s Information Criterion
BIC	Bayesian Information Criterion
B-GM	Bayesian Gaussian Mixture
B-GIDM	Bayesian Generalized Inverted Dirichlet Mixture
B-IDM	Bayesian Inverted Dirichlet Mixture
DIC	Deviance Information Criterion
EM	Expectation Maximization
FPR	False Positive Rate
GMM	Gaussian Mixture Model
GMMFS	Gaussian Mixture Model with Feature selection
GMMnoFS	Gaussian Mixture Model without Feature selection
GID	Generalized Inverted Dirichlet
GIDFS	Generalized Inverted Dirichlet with Feature Selection
GIDnoFS	Generalized Inverted Dirichlet without Feature Selection
GIDFSOL	GID mixture with Feature Selection and Outlier detection
IGID	Infinite Generalized Inverted Dirichlet Mixture
IGIDFS	Infinite GID mixture with Feature Selection
GIDOL	Generalized Inverted Dirichlet with Outlier detection
HOG	Histogram of Oriented Gradient
IGIDFS	Infinite GID mixture with Feature Selection
IGIDFS	Infinite GID mixture with Feature Selection
ML	Maximum Likelihood
ML-GID	Maximum Likelihood Generalized Inverted Dirichlet
MCMC	Markov chain Monte Carlo
MDL	Minimum Descriptive Length

MML	Minimum Message Length
PDF	Propability Density Function
RJMCMC	Reversible Jump Markov Chain Monte Carlo
SIFT	Scale-Invariant Feature Transform
TPR	True Positive Rate

Chapter 1

Introduction

Due to the evolution of information technology, the amount of data generated everyday have been growing dramatically. In general, not much data was used for analytic purposes other than making reports and performing simple statistic operations. Just recently, it became clear that data is an important asset when used for analysis that help in decision-making. This data can be generated from different fields, such as bioinformatics, climate, weather, astronomy, visionary, etc. Dealing with a large amount and variety of data calls for advanced approaches of understanding, processing, and summarizing the data. In general, machine learning algorithms have been widly can be categorized into supervised or unsupervised learning. Algorithms where the training data comprise examples of the input vectors along with their corresponding target "class labels" vectors, are known as supervised learning, i.e. classification and regression problems. In fact, the labels are predetermined to verify if the prediction is correct or not. On the other hand, in unsupervised learning, the training data consist of a set of vectors without corresponding class labels. In fact, the basic task of unsupervised learning is to discover groups of similar examples within the data, where the learning process is called clustering.

The problem of clustering, broadly stated, is to group a set of vectors into homogeneous categories. This problem has attracted much attention from different disciplines as an important step in many applications. Indeed, clustering is a powerful technique for knowledge discovery, data mining, and for several theoretical (see for instance [1, 2]) and experimental studies [3–6] that have been conducted in the past. The main goal is to identify patterns and features reflecting the regularities in data. In fact, the objects with similar characteristics are clustered together and objects with dissimilar characteristics are in different clusters. Many clustering approaches have been developed which can be roughly grouped into two categories. The first category contains heuristic algorithms where no probabilistic models are explicitly assumed (e.g. K-Means). The second category contains model-based methods which make inferences via probabilistic assumptions of the data distribution. In this thesis, we are interested with model-based approaches and especially mixture models.

Mixture model approach is the most popular model-based approach to clustering and offers a great practical value in modeling heterogenous data. The use of mixture models provides a formal approach to unsupervised learning and allows the incorporation of prior knowledge into cluster analysis which results in more meaningful clusters. The main driving force behind this interest in mixture models is their flexibility and strong theoretical foundation.

1.1 Clustering using Mixture Model

Mixture models form one of the most basic classes of statistical models for data clustering. The leading idea behind mixture models is that the data, which originate from different underlying sources, can be thought of as being generated using different underlying distributions. Presently, mixture models are used in many areas which include the statistical modeling of data (i.e. pattern recognition, computer vision, signal and image analysis, and machine learning). Mixture models provide suitable models for cluster analysis if we assume that each group of observations in a data set, contain clusters from a population with a different probability distribution [7]. In fact, mixture models assume that the data is generated by a set of parametric probability distributions. Therefore, the main challenge of the clustering process is to estimate the parameters of that mixture distribution and for each sample, to discover from which distribution it is generated. We suppose that we have a sample of N vectors $\mathscr{X} = (X_1, X_2, ..., X_N)$ with $X_i = x_{i1}, ..., x_{iD}$ and that $\mathscr{X}$ is composed of M components. A Mixture model can then be represented by

$$p(X_i|\Theta) = \sum_{j=1}^{M} p_j p(X_i|\theta_j) \tag{1.1}$$

where $p(X_i|\theta_j)$ is the probability density function (PDF) describing the j^{th} component. p_j is the mixing weight of component j, representing the probability that a randomly selected X_i was generated by component j, which are non-negative quantities under the constraints $0 \leq p_j \leq 1$ $(j = 1, ..., M)$ and $\sum_{j=1}^{M} p_j = 1$. Also, the symbol Θ represents all unknown parameters of the mixture model which is defined as $\Theta = \{p_1, ..., p_M, \theta_1, ..., \theta_M\}$. In general, mixture models can be finite or infinite depending on the number of components assumed to exist in the model (M).

In mixture modeling, choosing the appropriate PDF for the given data is a crucial decision. Selection of a suitable probability distribution improves the efficiency of modeling the data. Several research works use the Gaussian distribution and the corresponding Gaussian mixture models (GMM) since it has an analytically tractable PDF, and the analysis based on it can be derived in an explicit form. Since the Gaussian distribution has an unbounded support density, it is, therefore, not the best choice to model semi-bounded or bounded support densities. Recent works, however, have shown that other kinds of data are of particular importance. This is especially true for positive data which are naturally generated by several real life applications as thoroughly discussed in [8], for example, where a statistical model based on finite inverted Dirichlet mixture has been proposed for the clustering and modeling of such data. The inverted Dirichlet offers high flexibility and ease of use for the modeling of positive data [8–11]. However, this mixture has a significant limitation. Indeed, the inverted Dirichlet typically assumes that the features inside a given vector are positively correlated - which is not always the case in several real-life applications. Given this limitation, we propose the consideration of the generalized inverted Dirichlet which has a more general covariance structure and should be more practical. Throughout this thesis, we propose to use generalized inverted Dirichlet distribution to model positive data.

Another important task in mixture modeling is parameter estimation. The approaches used for this task can be a) deterministic or b) Bayesian. In deterministic approaches, it is assumed that all parameters are fixed and unknown, and the inference is founded on the likelihood of the data (likelihood function). The Expectation Maximization (EM) [12, 13] algorithm is widely used to estimate the mixture parameters by maximizing the likelihood function. Deterministic approaches suffer from some drawbacks such as dependency on the initialization, over-fitting, etc. To overcome those drawbacks in Bayesian approaches, a posterior distribution of each parameter is estimated by considering the prior of that parameter. Bayesian learning generally based on Markov Chain Monte Carlo (MCMC) approaches.

Selecting the optimal number of components is a fundamental step in mixture modeling. Several approaches have been adopted to solve this problem, examples include Bayesian Inference Criterion (BIC) [14], Minimum Descriptive Length (MDL) [15], Akaike's information criterion (AIC) [16], Bayesian Information Criterion [17] or the Deviance Information Criterion (DIC) [18].

There are other approaches that can perform parameters estimation and model selection simultaneously as shown in [19]. For instance, it is possible to sample from a posterior where M is considered unknown using reversible jump MCMC [20, 21]

1.2 Feature Selection

The recent increase of data dimensionality poses severe challenges to many existing data mining, pattern recognition, machine learning, and artificial intelligence methods, as well as feature selection methods with respect to efficiency and effectiveness [22–28]. These high-dimensional data need to be analyzed in order to gain the full potential from the gathered information. However, it pose different challenges for clustering algorithms that require a specialized solution [29]. The problem is that not all features are important to the clustering process. In such cases, choosing a subset of the original features will often lead to better performance. Feature selection has been shown to be a crucial step in several applications such as the analysis of gene expression data [30–32], fraud detection [33], text categorization [34], and user profiling [35]. Feature selection is a large research area and different approaches have been proposed in both supervised and unsupervised settings [36–40].

1.2.1 Outlier Rejection

Outlying data (noise data) can lead to inconsistent clustering analysis. An outlier observation is defined as the data which is deviated so much from other observations as to arouse suspicion that it was generated by different mechanism [41]. Also, outlier rejection has been extensively studied using statistical approaches [42]. Indeed, outlier points might lead to biased parameter estimation, and/or incorrect results. Therefore it is important to identify them prior or during the modeling and analysis stages [43]. Outlier rejection is extremely important in data mining and pattern recognition tasks.

1.3 Contributions

The object of this thesis is to propose several novel approaches for high-dimensional positive data clustering based on Generalized Inverted Dirichlet (GID) distribution using different types of learning approaches. The contributions of this thesis are given as following:

☞ **Simultaneous Clustering, Feature Selection and Outlier Rejection for Positive Data:**
We tackle simultaneously the problems of cluster validation (i.e. model selection), feature selection, and outlier rejection when clustering positive data. The proposed statistical framework is based on the generalized inverted Dirichlet distribution that offers a more practical and flexible alternative to the inverted Dirichlet which has a very restrictive covariance structure. The learning of the parameters of the resulting model is based on the minimization of a message length objective incorporating prior knowledge.

☞ **Finite Generalized Inverted Dirichlet Mixtures Estimation Using Bayesian Learning**
We developed a mixture model which was subjected to a fully Bayesian analysis. This

analysis was based on Markov Chain Monte Carlo (MCMC) simulation methods - namely ,the Gibbs sampling and Metropolis-Hastings, which were used to compute the posterior distribution of the parameters, and on Bayesian information criterion (BIC), which was used for model selection. The adoption of this purely Bayesian learning choice was motivated by the fact that Bayesian inference allows to deal with uncertainty in a unified and consistent manner.

☞ **An Infinite Mixture Model of Generalized Inverted Dirichlet Distributions:**
We propose an infinite mixture model for the clustering of positive data. The proposed model was based on the generalized inverted Dirichlet distribution. And was developed in an elegant way that allowed for simultaneous clustering and feature selection. It was learned using a fully Bayesian approach via Gibbs sampling.

☞ **A Fully Bayesian Framework Based on Reversible Jump Markov Chain Monte Carlo (RJMCMC) for Positive Data Clustering:**
A fully Bayesian model for finite Generalized Inverted Dirichlet (GID) learning using a Reversible Jump Markov Chain Monte Carlo (RJMCMC) approach was developed. RJMCMC enabled us to deal simultaneously with both model selection and parameter estimation.

1.4 Thesis Overview

The organization of this thesis is as follows:

❑ In Chapter 1, we give a brief introduction about mixture model and some related problems (such as feature selection and outlier rejection).

❑ In Chapter 2, we propose a MML inference framework approach to learn a finite Generalize Inverted Dirichlet mixture model. Synthetic and real data, generated from challenging applications such as visual scenes and objects clustering, were used to demonstrate the feasibility and advantages of the proposed method. This work was published in *knowledge Based*

System, Journal [44].

- ❏ In Chapter 3, a fully Bayesian learning approach is proposed for the GID mixture. This approach is based on MCMC simulation. In order to verify the effectiveness of the proposed approach, we evaluated it on the basis of two challenging applications concerning object classification and forgery detection. This work was published in *Expert Systems with Applications, Journal* [45].

- ❏ In Chapter 4, we propose an infinite mixture of GID based on a fully Bayesian approach via Gibbs sampling. It allowed for simultaneous parameter estimation and feature selection. The effectiveness of the proposed work was verified using a challenging application, namely image categorization. This work was published in *Information & Communication Technology-EurAsia , Conference* [46].

- ❏ In Chapter 5, we propose a fully Bayesian estimation of GID mixture model using RJMCMC where we simultaneously were able to estimate the model parameters and select the number of components. The merits of RJMCMC for GID mixture learning was investigated using synthetic and real data extracted from an interesting application, namely object detection.

- ❏ In Chapter 6, we summarize our contributions and present some potential future works.

Chapter 2

Robust Clustering and Unsupervised Feature Selection using Generalized Inverted Dirichlet Mixture Models

The discovery, extraction and analysis of knowledge from data generally depends upon the use of unsupervised learning methods, in particular, clustering approaches. Much recent research in clustering and data engineering has focused on the use of finite mixture models. They perform well when given uncertain data and they learn by example. The adoption of these models becomes a challenging task in the presence of outliers and in the case of high-dimensional data, which necessitates the deployment of feature selection techniques. In this chapter we simultaneously address the problems of cluster validation (i.e. model selection), feature selection, and outlier rejection when clustering positive data. The proposed statistical framework was based on the generalized inverted Dirichlet distribution. The learning of the parameters, of the resulting model, was based on the minimization of a message length objective that incorporated prior knowledge. We used synthetic data and real data generated from challenging applications, namely visual scenes and object clustering, to demonstrate the feasibility and advantages of the proposed method.

2.1 Introduction

Finite mixture models are the most popular model-based approaches for clustering and offer great practical value in modeling heterogenous data. The use of finite mixture models provides a formal approach to unsupervised learning and allows the incorporation of prior knowledge into cluster analysis which results in the creation of more meaningful clusters. One of the more challenging and important aspects of knowledge discovery in general, and of cluster analysis in particular, is the weighting and selection of variables [47–49]. Feature selection has been shown to be a crucial step in several applications such as the analysis of gene expression data [30–32], fraud detection [33], text categorization [34], and user profiling [35]. This is especially true for applications generating high-dimensional data which modeling has been the subject of extensive research efforts in the past [23–28]. Indeed, it is well-known that irrelevant variables generally hurt clustering models performance and cause significant over-fitting [50–52]. Feature selection is a large research area and different approaches have been proposed in both supervised and unsupervised settings. Of particular interest for us, will be simultaneous clustering and feature selection using finite mixture models which has received some attention recently (see, for instance, [36, 53] and references therein). The main motivation of these recent works was the fact that the feature selection problem was highly related to the model selection task and therefore, had to be performed simultaneously [38–40].

The estimation of the number of components (i.e. model selection or cluster validity) in the case of finite mixture models has been studied theoretically and experimentally in different ways (see, for instance, [54]). The most popular approaches have been based on the consideration of penalized likelihood methods. These approaches add a penalizing term to the likelihood such as a minimum description length (MDL) [55] or a minimum message length (MML) [56] criteria. In particular, MML-based learning has been the subject of extensive research recently and has been used for simultaneous clustering and feature selection [36] within statistical framework adopting a finite Gaussian mixture. Another kind of data, that has been generated from different real applications, is positive data. Such type of data is obviously not Gaussian since it has a semi-bounded density support. In this chapter, we propose to use a Generalized Inverted Dirichlet (GID) mixture for modeling positive data. GID sample features can be represented as conditionally independent by transforming GID into a product of inverted Beta distribution. Therefore, the conditional independence assumption among features commonly used by researchers, when assuming a diagonal covariance matrix in the case of the Gaussian for instance [36], in modeling high-dimensional data, becomes a fact for GID data sets without loss of accuracy. This interesting advantage of the GID is used in this work to develop a statistical model for simultaneous high-dimensional positive data clustering and feature selection. In contrast with previous simultaneous clustering and feature selection approaches, another principal focus of the developed model is outlier detection, which is a crucial problem in many real applications. Indeed, it is well-known that the clustering performance can be very sensitive to outliers [57–60] and that the resulting clustering models and selected features may be inaccurate in noisy environments [61, 62].

This chapter is organized as follows: in Section 2.1, a brief introduction is given. Then, our simultaneous clustering, feature selection, and outliers detection model is discussed in detail in Section 2.2. In Section 2.3, we develop a MML-based approach to learn the proposed model by choosing its appropriate priors and computing the related Fisher information. Extensive simulations and experiments are conducted in Section 2.4 to show the merits of the proposed work. Finally, in Section 2.5, a summary of this chapter is given.

2.2 The Generalized Inverted Dirichlet Finite Mixture Model

2.2.1 The Generalized Inverted Dirichlet Distribution

A D-dimensional positive vector $\vec{Y} = (Y_1, Y_2, \ldots, Y_D) \in \mathbb{R}_+^D$ is said to follow an inverted Dirichlet distribution if its density is given by [11]:

$$p(\vec{Y}|\vec{\alpha}) = \frac{\Gamma(|\vec{\alpha}|)}{\prod_{d=1}^{D+1}\Gamma(\alpha_d)} \frac{\prod_{d=1}^{D} Y_d^{\alpha_d - 1}}{(1+\sum_{d=1}^{D} Y_d)^{|\vec{\alpha}|}} \tag{2.1}$$

where $Y_d > 0$ where $d = 1\ldots, D$, $\vec{\alpha} = (\alpha_1, \ldots, \alpha_{D+1})$ is the vector of parameters, for $|\vec{\alpha}| = \sum_{d=1}^{D+1} \alpha_d$, $\alpha_d > 0$. This distribution is the multivariate extension of the 2-parameters inverted Beta distribution which is given by:

$$p_{IBeta}(Y_1|\alpha_1, \alpha_2) = \frac{\Gamma(\alpha_1+\alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)} \frac{Y_1^{\alpha_1 - 1}}{(1+Y_1)^{(\alpha_1+\alpha_2)}} \tag{2.2}$$

The mixed moments are given by:

$$E(\prod_{d=1}^{D} Y_d^{r_d}) = \frac{\Gamma(\alpha_{D+1} - r_+)}{\Gamma(\alpha_{D+1})} \prod_{d=1}^{D} \frac{\Gamma(\alpha_d + r_d)}{\Gamma(\alpha_d)} \tag{2.3}$$

for $\alpha_{D+1} > r_+ = \sum_{d=1}^{D} r_d$, and does not exist, otherwise. In particular, the mean and variance of the inverted Dirichlet satisfy the following conditions:

$$E(Y_d) = \frac{\alpha_d}{\alpha_{D+1} - 1} \tag{2.4}$$

$$Var(Y_d) = \frac{\alpha_d(\alpha_d + \alpha_{D+1} - 1)}{(\alpha_{D+1} - 1)^2(\alpha_{D+1} - 2)} \tag{2.5}$$

and the covariance between Y_d and Y_l is:

$$Cov(Y_d, Y_l) = \frac{\alpha_d \alpha_l}{(\alpha_{D+1}-1)^2(\alpha_{D+1}-2)} \tag{2.6}$$

Thus, any two random variables in $\vec{Y}$, are positively correlated when $\alpha_{D+1} > 2$, which is actually the case of the inverted Dirichlet. In real practical cases, however, the correlation may be negative and then the inverted Dirichlet becomes an inappropriate choice. Lingappaiah [63] has generalized the inverted Dirichlet distribution as follows:

$$p(\vec{Y}|\vec{\theta}) = \prod_{d=1}^{D} \frac{\Gamma(\alpha_d+\beta_d)}{\Gamma(\alpha_d)\Gamma(\beta_d)} \frac{Y_d^{\alpha_d-1}}{(1+\sum_{l=1}^{d} Y_l)^{\gamma_d}} \tag{2.7}$$

where $\vec{\theta} = (\alpha_1, \beta_1, \ldots, \alpha_D, \beta_D)$ and $\gamma_d = \beta_d + \alpha_d - \beta_{d+1}$, with $\beta_{D+1} = 0$. It is noteworthy that it is straightforward to verify that the generalized inverted Dirichlet (GID) has a more general covariance structure than the inverted Dirichlet and that it is reduced to an inverted Dirichlet with parameters $(\alpha_1, \ldots, \alpha_D, \beta_1)$ if we set $\gamma_1 = \gamma_2 = \ldots = \gamma_{D-1} = 0$ [63].

2.2.2 The Generalized Inverted Dirichlet Finite Mixture Model

Let us consider a data set $\mathscr{Y}$ of D-dimensional positive vectors where $\mathscr{Y} = (\vec{Y}_1, \vec{Y}_2, \ldots, \vec{Y}_N)$. We assume that $\mathscr{Y}$ is governed by a weighted sum of M GID component densities with parameters $\Theta = (\vec{\theta}_1, \vec{\theta}_2, \ldots, \vec{\theta}_M, p_1, p_2, \ldots, p_M)$, where $\vec{\theta}_j$ is the parameter vector of the jth component and $\{p_j\}$ are the mixing weights which are positive and sum to one such that:

$$p(\vec{Y}_i|\Theta) = \sum_{j=1}^{M} p_j p(\vec{Y}_i|\vec{\theta}_j) \tag{2.8}$$

where $p(\vec{Y}_i|\vec{\theta}_j)$ is the GID distribution with parameters $\vec{\theta}_j = (\alpha_{j1}, \beta_{j1}, \alpha_{j2}, \beta_{j2}, \ldots, \alpha_{jD}, \beta_{jD})$. In mixture-based clustering, each data point $\vec{Y}_i$ is assigned to all classes with different posterior probabilities $p(j|\vec{Y}_i) \propto p_j p(\vec{Y}_i|\vec{\theta}_j)$. The GID distribution allows the factorization of the posterior probability as (see Appendix A):

$$p(j|\vec{Y}_i) \propto p_j \prod_{l=1}^{D} p_{IBeta}(X_{il}|\theta_{jl}) \tag{2.9}$$

where we have set $X_{i1} = Y_{i1}$ and $X_{il} = \frac{Y_{il}}{1+\sum_{k=1}^{l-1} Y_{ik}}$ for $l > 1$. $p_{IBeta}(X_{il}|\theta_{jl})$ is an inverted Beta distribution with parameters $\theta_{jl} = (\alpha_{jl}, \beta_{jl})$, $l = 1, \ldots, D$. Thus, the clustering structure underlying $\mathscr{Y}$ is the same as the one underlying $\mathscr{X} = (\vec{X}_1, \ldots, \vec{X}_N)$, where $\vec{X}_i = (X_{i1}, \ldots, X_{iD})$, for $i = 1, \ldots, N$. It is governed by the following mixture model with conditionally independent features:

$$p(\vec{X}_i|\Theta) = \sum_{j=1}^{M} p_j \prod_{l=1}^{D} p_{IBeta}(X_{il}|\theta_{jl}) \tag{2.10}$$

In general, when formulating mixture models, we introduce latent allocation vectors $\vec{Z}_i = (Z_{i1}, \ldots, Z_{iM})$ which indicate to which mixture component the vector $\vec{X}_i$ belongs to, such that $Z_{ij} \in \{0, 1\}$, $\sum_{j=1}^{M} Z_{ij} = 1$, and $Z_{ij} = 1$ if $\vec{X}_i$ belongs to component j, and 0, otherwise. $\mathscr{Z} = \{\vec{Z}_1, \ldots, \vec{Z}_N\}$ is known as the set of "membership vectors" of the mixture model. Thus, the distribution of $\vec{X}_i$ given the class label $\vec{Z}_i$ is

$$p(\vec{X}_i|\Theta, \vec{Z}_i) = \prod_{j=1}^{M} \left(\prod_{l=1}^{D} p_{IBeta}(X_{il}|\theta_{jl}) \right)^{Z_{ij}} \tag{2.11}$$

It is noteworthy that the mixture model in Eq. 2.10 supposes that all features X_{il} have the same weight and thus, are equally important for the clustering task. Generally, this assumption is not realistic since some of the features might be irrelevant and then compromise the clustering structure. The goal of the next subsection is to improve our mixture model to take into account this important issue.

2.2.3 Feature Selection

Feature selection allows to take into account the fact that different features contribute differently to the clustering structure according to their degree of relevance which we should determine. Then, it has the potential to improve modeling and generalization capabilities if performed reliably and properly. Simultaneous clustering and feature selection (i.e. unsupervised feature selection) is one of the most difficult problems in data mining and machine learning. Indeed, the selection of features in this case is performed without a priori knowledge about the data labels. An interesting unsupervised feature selection/weighting formulation has been previously proposed in [36] in the case of finite Gaussian mixture-based clustering and has been applied successfully to other models (see, for instance, [53, 64]). The main idea is to suppose that a given feature X_{il} is generated from a mixture of two univariate distributions. The first one is assumed to generate relevant features and is different for each cluster. The second one is common to all clusters (i.e. independent from class labels) and assumed to generate irrelevant features. In our case, this idea can be formulated as follows:

$$p(\vec{X}_i|\Theta^*) = \sum_{j=1}^{M} p_j \prod_{l=1}^{D} \left[\rho_l p_{IBeta}(X_{il}|\theta_{jl}) + (1-\rho_l) p_{IBeta}(X_{il}|\lambda_l) \right] \tag{2.12}$$

where $\Theta^* = \{\Theta, \{\rho_l\}, \{\lambda_l\}\}$ is the set of all our unsupervised feature selection model parameters. ρ_l represents the probability that feature X_{il} is relevant for clustering, and $p_{IBeta}(X_{il}|\lambda_l)$ is an inverted Beta distribution[1], with parameter $\lambda_l = (\alpha_{\lambda|l}, \beta_{\lambda|l})$, which is common to all clusters and designed to generate irrelevant features. The model in Eq. 2.12 is clearly a generalization of our GID mixture since it is reduced to Eq. 2.10 if we assume that $\rho_l = 1$ for $l = 1, \ldots, D$. In the following subsection we shall see how we can make this model more robust by taking the presence of potential outliers in the data when taking clustering into account.

[1] It is noteworthy that it is possible also to consider that irrelevant features are generated by a mixture of inverted Beta distributions rather than only one distribution.

2.2.4 Outlier Detection

In the previous model, given by Eq. 2.12, examples are assumed to be noise-free which make them vulnerable to the presence of outliers. Indeed, real world application examples are generally afflicted with noise (or outliers) and it is important to deal with such outliers[2] [69]. Outlier detection has attracted great interest in the literature and many techniques have been proposed in the past (see, for instance, [70–74]). Here, we approach the outlier detection problem by incorporating an auxiliary outlier component, to which we associate a uniform density as done for instance in [37, 75–77], into the model:

$$p(\vec{X}_i|\Theta^{**}) = \sum_{j=1}^{M} p_j \prod_{l=1}^{D} \left[\rho_l p_{IBeta}(X_{il}|\theta_{jl}) + (1-\rho_l) p_{IBeta}(X_{il}|\lambda_l)\right] + p_{M+1} U(\vec{X}_i) \tag{2.13}$$

where $\Theta^{**} = (\Theta^*, p_{M+1})$ and $p_{M+1} = 1 - \sum_{j=1}^{M} p_j$ is the probability that $\vec{X}_i$ was not generated by the central model and $U(\vec{X}_i)$ is a uniform distribution common for all data to isolate vectors which are not generated by any of the M components forming the mixture model (i.e. the outliers). The main goal of the previous model is to make the unsupervised feature selection process robust to outliers [3] and is clearly a generalization of the model in Eq. 2.12 which can be verified by assuming that $p_{M+1} = 0$. It can be viewed also as a data dynamic weighting [79] approach which increases the importance of inliers and decreases the importance of outliers in clustering during the iterations.

2.3 Model Learning

2.3.1 MML-Based Learning

In this section, we develop an approach and a detailed algorithm to learn the parameters of our model in Eq. 2.13. By learning, we mean both the estimation of the parameters and the selection of the model's complexity (i.e. determination of the number of mixture components which best

[2] The problem of outliers detection has been called also anomaly detection, chance discovery [65], novelty detection [66], exception mining [67], and rare classes mining [68].

[3] A model which is not greatly altered by outliers can be also said to be resistant [78].

describes the data). The estimation of the parameters of finite mixture models has been generally based on the maximum likelihood (ML) approach. In our case, this is equivalent to the maximization of the following log-likelihood function:

$$\log p(\mathscr{X}|\Theta^{**}) = \sum_{i=1}^{N} \log \left[\sum_{j=1}^{M} p_j \prod_{l=1}^{D} \left[\rho_l p_{IBeta}(X_{il}|\theta_{jl}) + (1-\rho_l) p_{IBeta}(X_{il}|\lambda_l)\right] + p_{M+1} U(\vec{X}_i) \right] \tag{2.14}$$

Unfortunately ML estimation cannot be deployed for model selection. A better approach is to use the MML, which has been widely adopted in the past to learn finite mixture models. The MML principle can viewed as a paradigm for determining a model structure [56]. It aims to strike a balance between model complexity and goodness of fit. It is a model selection criterion that states that the best model to represent a set of given data is the one that requires the minimum amount of information to transmit the data efficiently from a sender to a receiver. Moreover, it prevents us from over-fitting the data by embodying the well-known Occam's Razor principle, thus we shall adopt it here. The message length is based on the following two parts: one representing the parameter values of the model and one encoding the data using the model itself. The message length can be written as follows [56]:

$$MML(M) = -\log p(\Theta^{**}) + \frac{1}{2}\log|I(\Theta^{**})| + \frac{c}{2}(1+\log\frac{1}{12}) - \log p(\mathscr{X}|\Theta^{**}) \tag{2.15}$$

where $p(\Theta^{**})$ and $|I(\Theta^{**})|$ denote the prior distribution and the Fisher information, respectively. The constant $c = M + 3D + 2DM$ is the total number of free parameters in our model. Both the optimal number of components, $\hat{M}$, and the best estimate, $\hat{\Theta}^{**}$, correspond to the lowest message length. The efficiency of the MML approach is highly dependent on the choice of prior distributions which we will tackle, along with the computation of the Fisher information, in the next subsection.

2.3.2 Prior Distribution $p(\Theta^{**})$ and Fisher Information $|I(\Theta^{**})|$

In order to be able to factorize both $p(\Theta^{**})$ and $|I(\Theta^{**})|$, we use a common assumption by considering independence among the different groups of parameters $\vec{P} = (p_1, \dots, p_M, p_{M+1})$, θ_{jl}, λ_l and ρ_l. We begin by selecting the prior distribution $p(\Theta^{**})$:

$$p(\Theta^{**}) = p(\vec{P}) \prod_{l=1}^{D} \left[p(\vec{\rho}_l) p(\lambda_l) \prod_{j=1}^{M} p(\theta_{jl}) \right] \tag{2.16}$$

The Dirichlet distribution is frequently used as a prior to the vector of mixing parameters, $\vec{P}$, due to its flexibility and suitability in modeling proportional vectors. The same choice is valid also for the $\vec{\rho}_l$ parameter for which we consider a one-dimensional Dirichlet (i.e. Beta distribution). In both cases, we set the hyperparameters to 0.5, which gives us the following Jeffrey's priors [80]:

$$p(\vec{P}) \propto \prod_{j=1}^{M+1} \frac{1}{\sqrt{p_j}} \qquad p(\vec{\rho}_l) \propto \frac{1}{\sqrt{\rho_{l_1} \rho_{l_2}}} \tag{2.17}$$

where $\vec{\rho}_l = (\rho_{l_1}, \rho_{l_2})$, $\rho_{l_1} = \rho_l$ and $\rho_{l_2} = 1 - \rho_{l_1}$. For θ_{jl}, we know experimentally that $\alpha_{jl} \in [0, e^{6}\frac{\hat{\alpha}_{jl}+\hat{\beta}_{jl}}{\hat{\alpha}_{jl}}]$ and $\hat{\beta}_{jl} \in [0, e^{6}\frac{\hat{\alpha}_{jl}+\hat{\beta}_{jl}}{\hat{\beta}_{jl}}]$, where the hat symbol on parameters indicates estimated quantities. By considering $\hat{A}^{\theta_{jl}} = e^{6}\frac{(\hat{\alpha}_{jl}+\hat{\beta}_{jl})^2}{\hat{\alpha}_{jl}\hat{\beta}_{jl}}$, it is clear that the parameters $(\alpha_{jl}, \beta_{jl})$ are defined on the simplex $\{(\alpha_{jl}, \beta_{jl}) : \alpha_{jl} + \beta_{jl} < \hat{A}^{\theta_{jl}}\}$. Thus, the two-dimensional Dirichlet distribution can be considered as a good choice to model prior knowledge about θ_{jl}. In particular, we consider a Dirichlet distribution with hyper-parameters set to 0 (i.e. Laplace prior):

$$p(\theta_{jl}) = \frac{\hat{A}^{\theta_{jl}}}{\alpha_{jl}\beta_{jl}(\hat{A}^{\theta_{jl}} - \alpha_{jl} - \beta_{jl})} \tag{2.18}$$

It is noteworthy that the development done for θ_{jl} is valid for λ_l as well. Thus, we obtain the following prior

$$p(\lambda_l) = \frac{\hat{A}^{\lambda_l}}{\alpha_{\lambda|l}\beta_{\lambda|l}(\hat{A}^{\lambda_l} - \alpha_{\lambda|l} - \beta_{\lambda|l})} \tag{2.19}$$

where $\hat{A}^{\lambda_l} = e^{6\frac{(\hat{\alpha}_{\lambda|l}+\hat{\beta}_{\lambda|l})^2}{\hat{\alpha}_{\lambda|l}\hat{\beta}_{\lambda|l}}}$.

The Fisher information $|I(\vec{P})|$ and $|I(\vec{\rho}_l)|$ are computed by taking the determinant of the information matrices of the multinomial distributions, having parameters $\vec{P}$ and $\vec{\rho}_l$, respectively:

$$|I(\vec{P})| = N^M \prod_{j=1}^{M+1} \frac{1}{p_j} \qquad |I(\vec{\rho}_l)| = \frac{N}{\rho_{l_1}\rho_{l_2}} \tag{2.20}$$

The Fisher information of the parameters θ_{jl} and λ_l are computed by considering the log-likelihood of each feature taken separately. Let $\mathscr{X}_l^j$ (resp. $\mathscr{X}_l$) be the set of one-dimensional observations obtained by considering the lth feature of the N_{jl} (resp. N_l) data points of the jth relevant component (resp. background common component). The second derivative of the negative log-likelihood function, corresponding to $\mathscr{X}_l^j$, is given by:

$$-\frac{\partial^2 \log(p(\mathscr{X}_l^j|\theta_{jl}))}{\partial\theta_{jl}^2} = -\sum_{i=1}^{N_{jl}} \frac{\partial^2}{\partial\theta_{jl}^2}\left[\log p_j + \log\rho_{l_1} + \log p_{IBeta}(X_{il}|\theta_{jl})\right] \tag{2.21}$$

which gives us the following equation for the Fisher information of θ_{jl}:

$$|I(\theta_{jl})| = N_{jl}^2|\Psi'(\alpha_{jl})\Psi'(\beta_{jl}) - \Psi'(\alpha_{jl}+\beta_{jl})(\Psi'(\alpha_{jl})+\Psi'(\beta_{jl}))| \tag{2.22}$$

where $\Psi'(.)$ is the second derivative of the logarithm of the Gamma function. Using the same approach, we obtain the Fisher information of λ_l:

$$|I(\lambda_l)| = N_l^2|\Psi'(\alpha_{\lambda|l})\Psi'(\beta_{\lambda|l}) - \Psi'(\alpha_{\lambda|l}+\beta_{\lambda|l})(\Psi'(\alpha_{\lambda|l})+\Psi'\beta_{\lambda|l}))| \tag{2.23}$$

The message length is obtained by substituting $p(\Theta^{**})$ and $|I(\Theta^{**})|$ into equation 2.15 (see Appendix B):

$$\begin{aligned} MML(M) = {} & \frac{M+3D+2MD}{2}\log(N) + D\sum_{j=1}^{M}\log p_j + M\sum_{l=1}^{D}\log(\rho_{l_1}) + \sum_{l=1}^{D}\log(\rho_{l_2}) \\ & + \frac{1}{2}\Bigg[\sum_{j=1}^{M}\sum_{l=1}^{D}\bigg(\log|\Psi'(\alpha_{jl})\Psi'(\beta_{jl}) - \Psi'(\alpha_{jl}+\beta_{jl})(\Psi'(\alpha_{jl})+\Psi'(\beta_{jl}))|)\bigg) \\ & + \sum_{l=1}^{D}\bigg(\log|\Psi'(\alpha_{\lambda|l}) + \Psi'(\beta_{\lambda|l}) - \Psi'(\alpha_{\lambda|l}+\beta_{\lambda|l})(\Psi'(\alpha_{\lambda|l})+\Psi'\beta_{\lambda|l}))|\bigg)\Bigg] \\ & + \sum_{l=1}^{D}\bigg(\log\alpha_{\lambda|l} + \log\beta_{\lambda|l} + \log(\hat{A}^{\lambda_l} - \alpha_{\lambda|l} - \beta_{\lambda|l})\bigg) - \log p(\mathscr{X}|\Theta^{**}) \\ & + \sum_{j=1}^{M}\sum_{l=1}^{D}\bigg(\log\alpha_{jl} + \log\beta_{jl} + \log(\hat{A}^{\theta_{jl}} - \alpha_{jl} - \beta_{jl})\bigg) + \frac{c}{2}(1+\log\frac{1}{12}) \end{aligned} \tag{2.24}$$

2.3.3 Parameters Estimation

The estimation of the parameters is based on the minimization of the message length of the data set $\mathscr{X}$ given by Eq. 2.24. The minimization must be done under the constraints $\sum_{j=1}^{M+1} p_j = 1$ and $\rho_{l_1} + \rho_{l_2} = 1$, which we can take into account by introducing Lagrange multipliers Λ_1 and Λ_2. Thus, we have to optimize the following new objective:

$$L(\Theta^{**}, \mathscr{X}) = -MML(M) + \Lambda_1(1 - \sum_{j=1}^{M+1} p_j) + \Lambda_2(1 - \rho_{l_1} - \rho_{l_2}) \tag{2.25}$$

We estimate $\vec{P}$ by setting to zero the derivatives of $L(\Theta^{**}, \mathscr{X})$ with respect to p_j and Λ_1 as following:

$$\frac{\partial L(\Theta^{**}, \mathscr{X})}{\partial p_j} = \begin{cases} \frac{\partial \log p(\mathscr{X}|\Theta^{**})}{\partial p_j} - \frac{D}{p_j} - \Lambda_1 = 0 = \frac{1}{p_j}\big(\sum_{i=1}^{N} p(j|\vec{X}_i) - D\big) - \Lambda_1 & \text{if } j = 1, \dots, M \\ \frac{\partial \log p(\mathscr{X}|\Theta^{**})}{\partial p_j} - \Lambda_1 = 0 = \frac{1}{p_j}\big(\sum_{i=1}^{N} p(j|\vec{X}_i)\big) - \Lambda_1 & \text{if } j = M+1 \end{cases} \tag{2.26}$$

$$\frac{\partial L(\Theta^{**}, \mathscr{X})}{\partial \Lambda_1} = 1 - \sum_{j=1}^{M+1} p_j = 0 \tag{2.27}$$

since the $\{p_j\}$ are positive, their update formula were computed in the M-step as:

$$p_j \propto \begin{cases} \max(\sum_{i=1}^{N} p(j|\vec{X}_i) - D, 0) & \text{if } j = 1, \dots, M \\ \sum_{i=1}^{N} p(j|\vec{X}_i) & \text{if } j = M+1 \end{cases} \tag{2.28}$$

where $p(j|\vec{X}_i)$ is computed in the E-step as:

$$p(j|\vec{X}_i) \propto \begin{cases} p_j \prod_{l=1}^{D} \left[\rho_l p_{IBeta}(X_{il}|\theta_{jl}) + (1-\rho_l) p_{IBeta}(X_{il}|\lambda_l)\right] & \text{if } j = 1, \dots, M \\ p_j U(\vec{X}_i) & \text{if } j = M+1 \end{cases} \tag{2.29}$$

The update formulae for ρ_{l_1} can be obtained in a similar way as (see Appendix C):

$$\frac{1}{\rho_{l_1}} = 1 + \frac{\max(\sum_{i=1}^{N} \sum_{j=1}^{M} p(j|\vec{X}_i) \frac{\rho_{l_2} p_{IBeta}(X_{il}|\lambda_l)}{\rho_{l_1} p_{IBeta}(X_{il}|\theta_{jl}) + \rho_{l_2} p_{IBeta}(X_{il}|\lambda_l)} - 1, 0)}{\max(\sum_{i=1}^{N} \sum_{j=1}^{M} p(j|\vec{X}_i) \frac{\rho_{l_1} p_{IBeta}(X_{il}|\theta_{jl})}{\rho_{l_1} p_{IBeta}(X_{il}|\theta_{jl}) + \rho_{l_2} p_{IBeta}(X_{il}|\lambda_l)} - M, 0)} \tag{2.30}$$

In order to estimate the θ_{jl} and λ_l parameters, we will use Fisher's scoring methods. These methods are based on the first (Gradient), and mixed derivatives (Hessian) of the function $L(\Theta^{**}, \mathscr{X})$. Given the initial estimation of the parameters, Fisher's scoring approach can be used to update them. Updating θ_{jl} and λ_l in the M-step is based on the following equations:

$$\hat{\theta}_{jl}^{t+1} = \hat{\theta}_{jl}^{t} - \big(\frac{\partial^2}{\partial \theta_{jl}^2} L(\Theta^{**}, \mathscr{X})\big)^{-1}_{\theta_{jl} = \hat{\theta}_{jl}^{t}} \times \big(\frac{\partial}{\partial \theta_{jl}} L(\Theta^{**}, \mathscr{X})\big)_{\theta_{jl} = \hat{\theta}_{jl}^{t}} \tag{2.31}$$

$$\hat{\lambda}_{l}^{t+1} = \hat{\lambda}_{l}^{t} - \big(\frac{\partial^2}{\partial \lambda_{l}^2} L(\Theta^{**}, \mathscr{X})\big)^{-1}_{\lambda_{l} = \hat{\lambda}_{l}^{t}} \times \big(\frac{\partial}{\partial \lambda_{l}} L(\Theta^{**}, \mathscr{X})\big)_{\lambda_{l} = \hat{\lambda}_{l}^{t}} \tag{2.32}$$

The computation of the gradient and Hessian of $L(\Theta^{**}, \mathscr{X})$ is detailed in Appendix D and the complete learning process is given in Algorithm 1. The proposed algorithm first detects outlying data. Then, only the inliers are considered to determine the model of the data (i.e. number of clusters, relevant features, and optimal parameters). It is noteworthy that the initialization using the Fuzzy C-Means algorithm and the method of moments is performed by supposing that the initial number of clusters is $M+1$. We may also note that our learning process was actually based on the

Algorithm 1 Learning of the GID mixture model with simultaneous feature selection and outliers rejection.

```
Input: $D$-dimensional data set $\mathscr{X} = \{\vec{X}_1, \dots, \vec{X}_N\}$, $M_{max}$, $M_{min}$.
Output: $M$, $\Theta^{**}$.
{Initialization}
  $M = M_{max}$, $\rho_{l1} = \rho_{l2} = 0.5$.
  Apply Fuzzy C-Means Algorithm.
  Apply the method of moments.
while $M \geq M_{min}$ do
    repeat
      for all $1 \leq j \leq M+1$ {[E Step]} do
        Compute the posteriors $p(j|\vec{X}_i)$ for $i = 1, \dots, N$ using equation 2.29.
      end for
      {[M step]}
      Update $\vec{P}$ using equation 2.28.
      for all $1 \leq l \leq D$ do
        Update $\rho_{l_1}$ using equation 2.30.
        Update $\lambda_l$ using using Fisher's scoring method.
        for all $1 \leq j \leq M$ do
          Update $\theta_{jl}$ using Fisher's scoring method.
        end for
      end for
      if $p_j = 0$ then prune the $jth$ component $\theta_j$, $M = M - 1$.
      if $\rho_l = 0$ then prune the $lth$ feature.
    until convergence
    Record $\Theta^{**}$, $M$ and MML of the model.
    Remove the $jth$ component $(\theta_{jl}, l = 1, \dots, D)$ of the mixture with the smallest weight.
    M= M-1.
end while
return $\Theta^{**}$, $M$ with the lowest message length.
```

expectation-maximization (EM) approach [81] where both the E- and M-steps have a complexity of $O(NMD)$.

2.4 Experimental Results

In this section, extensive experiments were conducted in order to evaluate the benefits of using the proposed model. Our experiments involved synthetic data sets as well as real-world challenging applications that concern unsupervised scenes and objects categorization. The main goal of the application on synthetic data was to show the merits of our model and its ability to efficiently perform simultaneous clustering, feature selection, and outlier rejection, in the case of positive data. The goal of the real life applications was to compare the performances of the GID mixture and the widely used Gaussian mixture model (GMM) when feature selection and outlier rejection are integrated into it in the same way. It is noteworthy that the learning of the GMM in this case has been based on the same methodology described in the previous section to learn the GID mixture. It is also worth noting, that we have used a common approach for the definition of the uniform distribution in our model. The used approach supposes that the data follow a single component model averaged over all the observations [76] which, in our case, uses the following: $U(\vec{X}) = \frac{1}{N}\sum_{i=1}^{N}\prod_{d=1}^{D}\big(\hat{\rho}_l p(X_{il}|\hat{\theta}_l) + (1-\hat{\rho}_l)p(X_{il}|\hat{\lambda}_l)\big)$, where the parameters $\hat{\rho}_l$, $\hat{\theta}_l$ and $\hat{\lambda}_l$ are estimated using ML. Of course, other choices are possible, but the main advantage of the used uniform distribution, which was found appropriate according to our experiments, is the fact that it takes into account that outliers should be sparsely distributed, which is generally the case for real data sets.

2.4.1 Evaluation on Synthetic Data

In this experiment, we first start by evaluating the performance of the proposed GID mixture without feature selection and outlier rejections. Second, we consider the case where clustering and feature selection are performed simultaneously (GIDFS). The main goal here is to evaluate the ability of the algorithm in selecting features when no outliers are present. Since the proposed model extracts features of a GID sample in a space where features are independent, we generated the relevant features of our data in the transformed space from a mixture of inverted beta distributions, and the irrelevant ones from one inverted beta. Using this generation approach, three 3-dimensional data sets are synthesized from 2-, 3-, and 4-components GID mixtures. Table 2.1 shows the parameters used for generating these data sets. Also, eight 'noisy' features are appended to these sets which increases the dimensionality of the data to 11 dimensions. The eight irrelevant features were generated from an inverted beta with parameters $(\alpha, \beta) = (3, 15)$. Table 2.2 shows

Table 2.1: Parameters used to generate the three synthetic data sets (n_j represents the number of elements in cluster j).

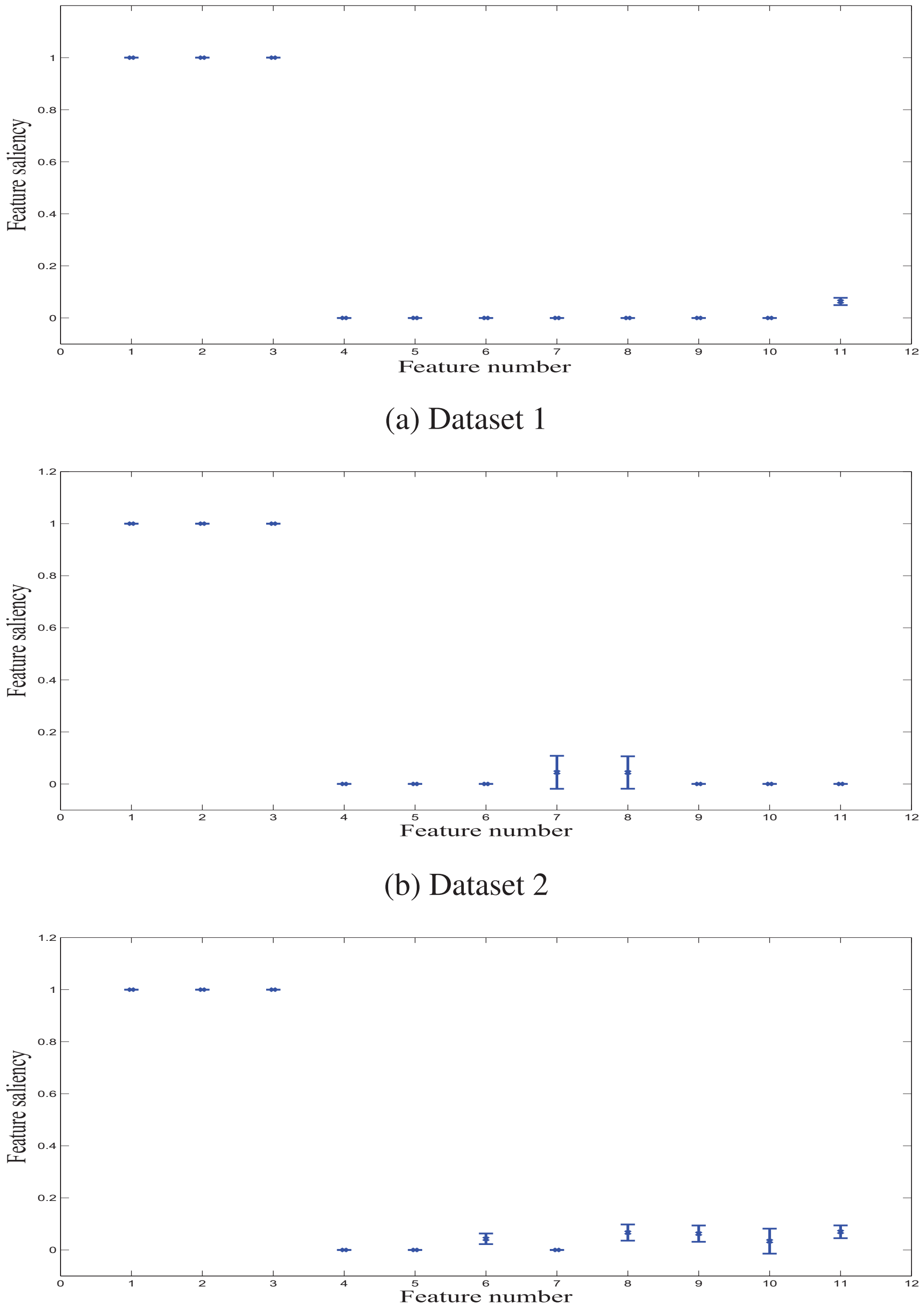
	n_j	j	α_{j1}	β_{j1}	α_{j2}	β_{j2}	α_{j3}	β_{j3}
Data set 1	300	1	40	28	33	46	18	40
	300	2	18	35	43	25	21	14
Data set 2	300	1	30	44	25	40	35	22
	300	2	16	28	17	32	21	41
	300	3	40	28	33	46	18	40
Data set 3	300	1	16	28	17	32	21	41
	300	2	18	35	43	25	21	14
	300	3	40	28	33	46	18	40
	300	4	30	44	25	40	35	22

the classification results for the three synthetic data sets using GIDFS and GID models. According to this table, using feature selection has improved the clustering performance significantly for the three generated data sets compared to the case where all features were considered relevant for the clustering process. It is noteworthy that in all cases, the MML criterion was able to select the correct number of clusters. The obtained saliencies of all 11 features for the generated synthetic data sets, when using GIDFS, are shown in Fig. 2.1. According to this figure, it is obvious that the model was able to locate the relevant features (i.e. features 1, 2, and 3) and the irrelevant ones (i.e. features from 4 to 11).

Table 2.2: Classification results for the three synthetic data sets using GIDFS and GID models. n_j represents the number of well-classified vectors.

Data set	j	$GIDFS : n_j$	Accuracy	$GID : n_j$	Accuracy
Data set 1	1	300		288	
	2	297	99.50%	282	95.00 %
Data set 2	1	295		275	
	2	296		280	
	3	300	99.00 %	289	93.78%
Data set 3	1	283		269	
	2	296		290	
	3	291		282	
	4	293	96.17%	285	93.00%

The second experiment was conducted in the presence of noise by appending 10 outliers



(a) Dataset 1

(b) Dataset 2

(c) Dataset 3

Figure 2.1: Features saliencies for the three synthetic data sets obtained using GIDFS.

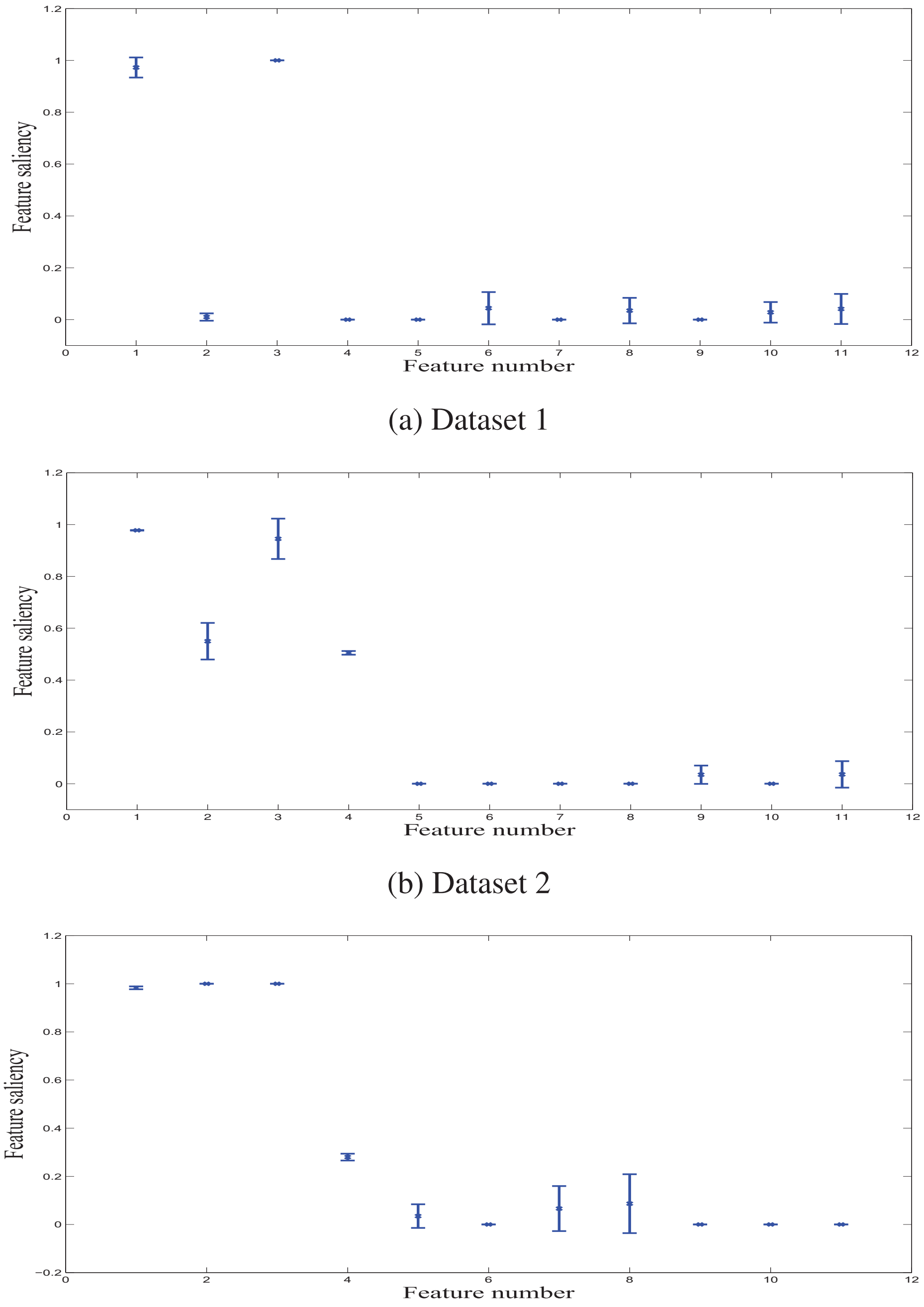
to each of the previously generated data sets. The addition of these outliers did not affect the MML criterion since we were able to find the exact number of clusters for all data sets. Table 2.3 shows the classification results obtained using both GIDFS and GID in this case. From this table, we can notice that the classification accuracy has decreased. The feature saliencies are shown in Fig. 2.2 from which we notice that the presence of outliers has compromised the feature selection process by affecting high weights to some irrelevant features and by decreasing the saliency of some relevant ones. Table 2.4 shows the classification results for the contaminated data sets by performing simultaneous clustering, feature selection, and outlier rejection (GIDFSOL), and by performing clustering without feature selection but with outlier rejection (GIDOL). In both cases, the MML criterion performed well by selecting the exact number of clusters for the three data sets. As shown in Fig. 2.3, the feature saliencies have been identified correctly by GIDFSOL. Moreover, by comparing tables 2.3 and 2.4, it is clear that the quality of clustering has been improved by taking outliers into account.

Table 2.3: Classification results for the three synthetic data sets, when contaminated with outliers, using GIDFS and GID. n_j represents the number of well-classified vectors.

Data set	j	$GIDFS : n_j$	Accuracy	$GID : n_j$	Accuracy
Data set 1	1	281		273	
	2	279	93.33%	255	88.00%
Data set 2	1	288		285	
	2	292		279	
	3	291	96.78%	277	93.44%
Data set 3	1	286		270	
	2	277		273	
	3	283		280	
	4	290	94.67%	269	91.00%

2.4.2 Images Clustering

With advances in multimedia technology, images and videos are becoming available at an explosive rate. A crucial problem is how to efficiently organize and index those multimedia data. Several approaches and techniques have been developed in the past to reach this goal. In this section, we shall focus on two challenging problems related to multimedia data organization - namely visual scenes and visual object clustering in order to evaluate the merits of our work.



(a) Dataset 1

(b) Dataset 2

(c) Dataset 3

Figure 2.2: Features saliencies for the three synthetic data sets, in the presence of outliers, obtained using GIDFS.

Visual Scenes Clustering

The visual scenes clustering application is based on the data set originally collected in [82] from which five categories were used in our experiments - namely living room, coasts, forest, mountain,

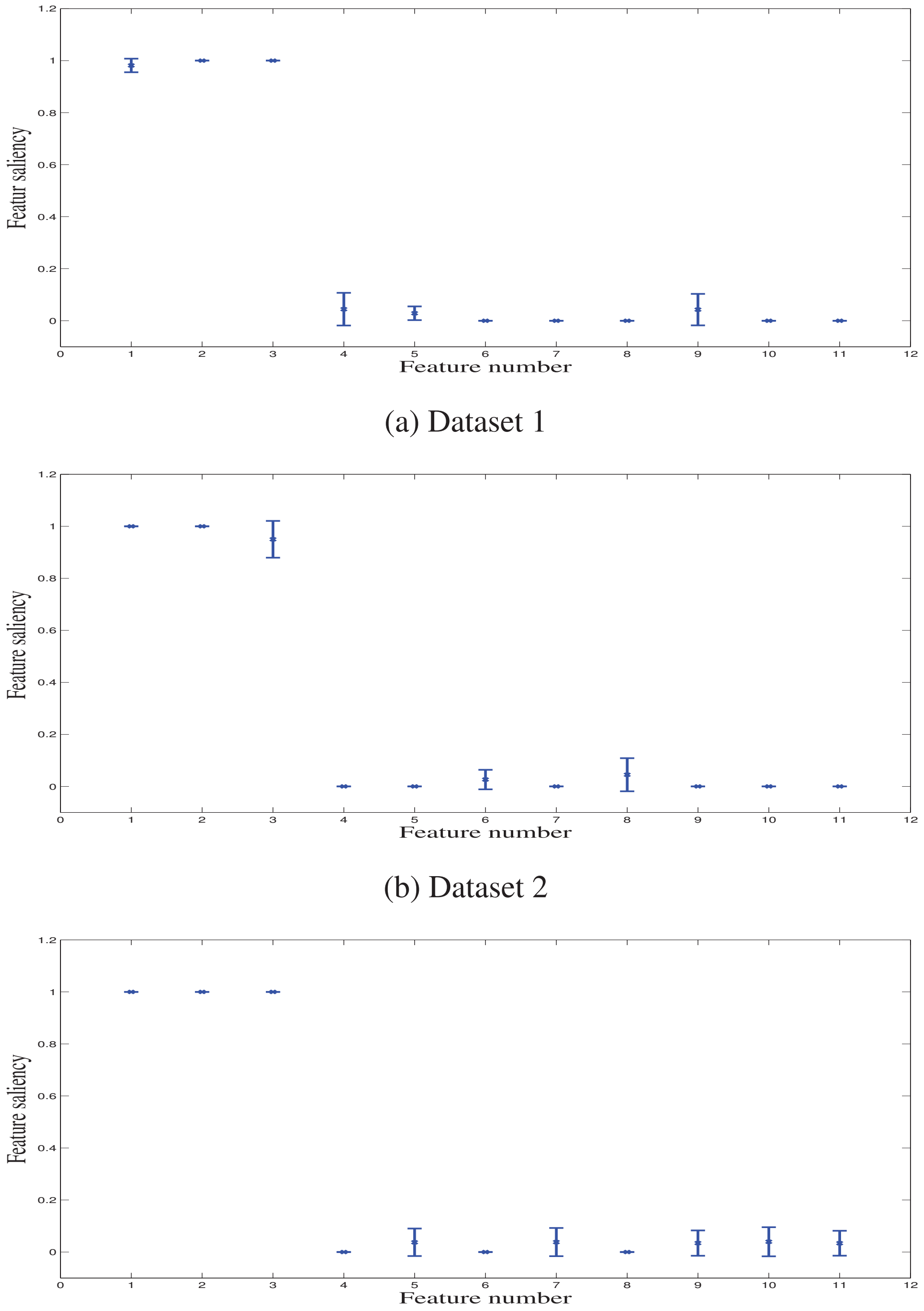
Table 2.4: Classification results for the three synthetic data sets, when contaminated with outliers, using GIDFSOL and GIDOL. n_j represents the number of well-classified vectors.

Data set	j	$GIDFSOL : n_j$	Accuracy	$GIDOL : n_j$	Accuracy
Data set 1	1	300		291	
	2	300		289	
	outlier	10	100 %	10	96.67%
Data set 2	1	290		282	
	2	296		290	
	3	296		292	
	outlier	10	98%	10	96.00%
Data set 3	1	284		280	
	2	298		290	
	3	291		285	
	4	291		279	
	outlier	10	96.42%	10	94.50%

and tall buildings (see Fig. 2.4). From each category, 280 images were considered. Moreover, 15 textural images taken from the MIT Vistex gray level texture database [4] were appended to this data set and considered as outliers (see Fig. 2.5). The main goal of this application is to investigate the effects of outliers on the performance of the clustering algorithm for positive real data sets using GID mixture model. Also, the performance of GID mixture has been compared with GMM model.

An important part of visual scenes clustering problem is feature extraction. Many global and local visual descriptors have been proposed in the past. Here, we use local Histogram of Oriented Gradient (HOG) descriptor [83] which generates positive features and which we found efficient and convenient for our applications. Experiments are conducted by considering three windows for the HOG descriptor, which allows us to represent each image by an 81-dimensional vector of features. We conducted 5-fold cross-validation experiments, then we report the number of components returned by the algorithm, features relevancies, and the confusion matrix for each

[4] http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html



(a) Dataset 1

(b) Dataset 2

(c) Dataset 3

Figure 2.3: Features saliencies for the three synthetic data sets, in the presence of outliers, obtained using GIDFSOL.

experiment.

In the first experiment, we compared the performance of GID and GMM mixture models without taking into consideration the relevancy of the features nor the presence of outliers. Table 2.5

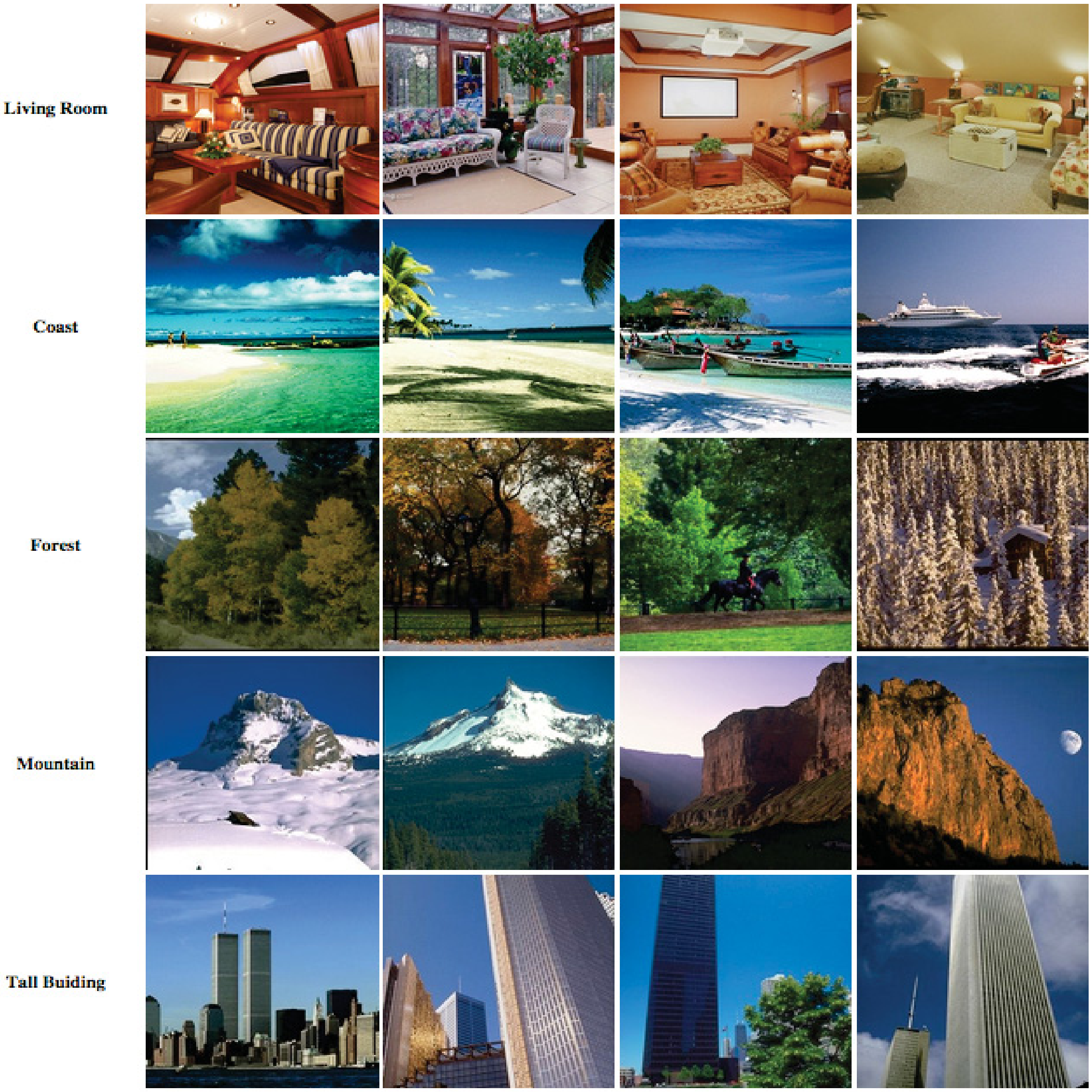


Figure 2.4: Examples of images from the 5 classes of visual scenes considered in our experiments.

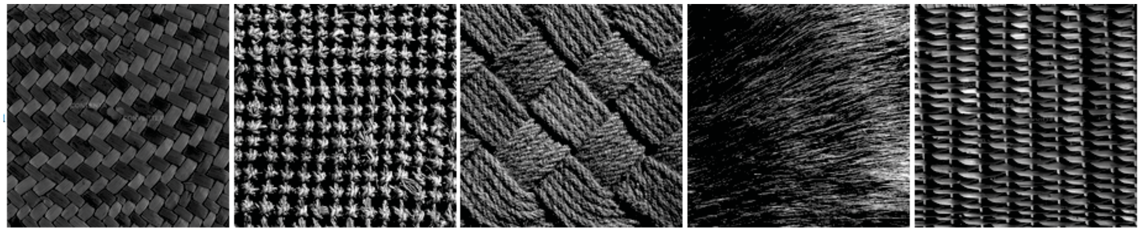


Figure 2.5: Examples of outlier images taken from the the MIT Vistex texture data set.

shows the confusion matrix when a GID model was used. According to this table, the number of misclassified images was 85, which gives us an accuracy of 71.19%. On the other hand, Table 2.5

presents the confusion matrix when the GMM was applied; the number of misclassified images is 94 which represents an accuracy of 68.13%. Both models were not able to detect the outlier images as a different class. In fact, all outliers have been assigned to the five main classes.

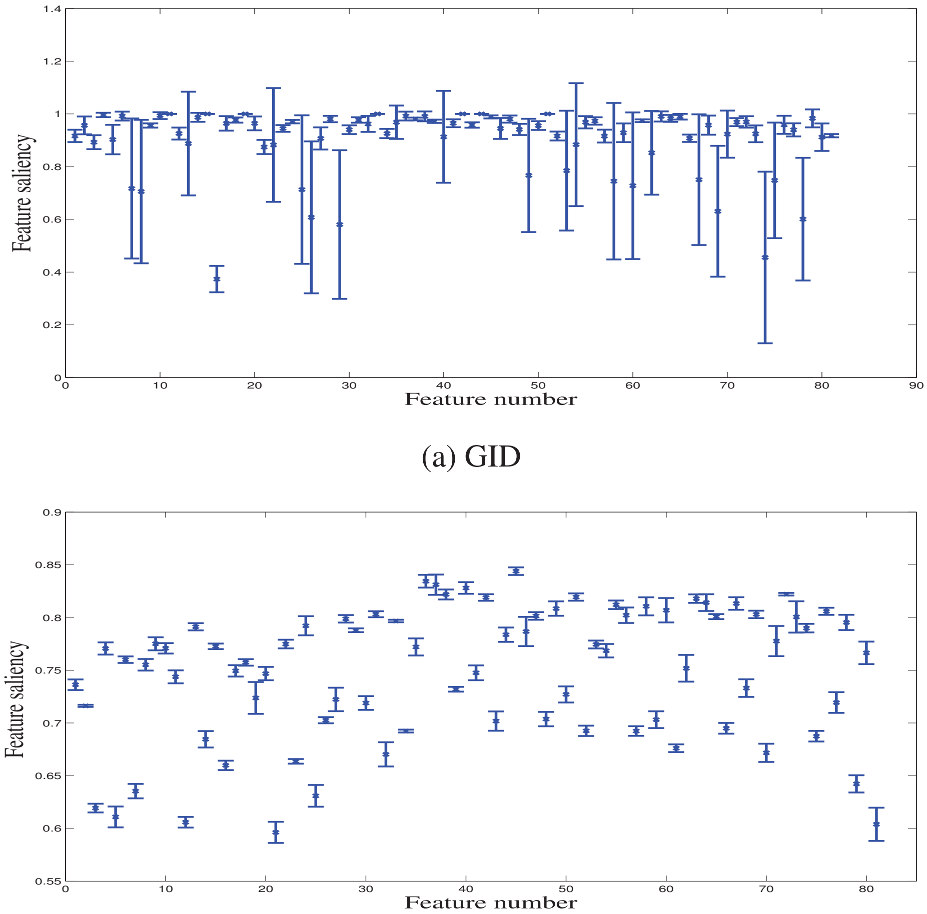
Table 2.5: The confusion matrix in the case of the visual scenes clustering problem when applying GID mixture without feature selection nor outliers detection.

	living room	coast	forest	mountain	tall building
living room	48	2	0	3	3
coast	0	46	0	10	0
forest	0	0	52	4	0
mountain	0	3	7	34	12
tall building	25	1	0	0	30
outlier class	0	1	0	14	0

Table 2.6: The confusion matrix in the case of the visual scenes clustering problem when applying GMM without feature selection nor outliers detection.

	living room	coast	forest	mountain	tall building
living room	43	5	0	5	3
coast	0	46	0	10	0
forest	0	2	50	4	0
mountain	2	3	11	31	9
tall building	23	2	0	0	31
outlier class	3	1	0	11	0

The second experiment was conducted by taking into consideration the relevancy of the features but without performing outlier detection for both the GID mixture and GMM. Figure 2.6 shows the feature saliencies obtained by both models. Table 2.7 shows the confusion matrix when GID mixture model was applied. From this table, we can see that the number of misclassified images was 74, which represents an accuracy of 74.92%. Table 2.8 shows the confusion matrix when applying GMM, which gives an accuracy of 70.17%. Comparing the results of the first and the second experiments, we conclude that the feature selection process improves the classification results. However, the outliers have not been assigned to a separate cluster. In the third experiment, we considered both the relevancy of features and the presence of outliers. Figure 2.7 shows the



(a) GID

(b) GMM

Figure 2.6: Features saliencies obtained in the case of the visual scenes clustering problem when performing feature selection without outlier detection using (a) GID mixture and (b) GMM.

Table 2.7: The confusion matrix in the case of the visual scenes clustering problem when applying GMM with feature selection but without outlier detection.

	living room	coast	forest	mountain	tall building
living room	47	1	1	2	5
coast	0	48	0	8	0
forest	0	0	50	5	1
mountain	0	2	9	44	1
tall building	22	1	0	1	32
outlier class	0	4	2	9	0

feature saliencies obtained by both models, which were able to detect the class representing the outliers. The classification accuracy has increased for both mixtures as we can clearly see in

Table 2.8: The confusion matrix in the case of the visual scenes clustering problem when applying GMM with feature selection but without outlier detection.

	living room	coast	forest	mountain	tall building
living room	44	0	1	5	6
coast	1	47	0	6	2
forest	1	0	51	3	2
mountain	4	2	15	34	1
tall building	13	1	4	2	36
outlier class	0	4	0	8	3

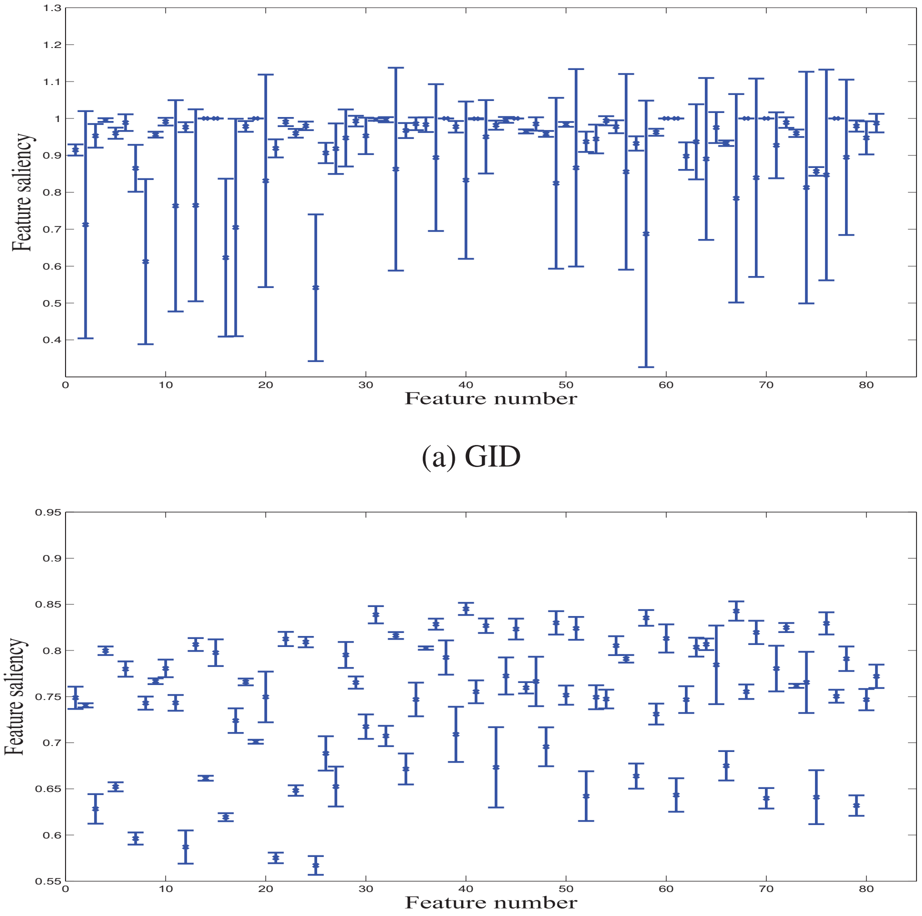
Tables 2.9 and 2.10 which show the confusion matrices when using the GID mixture model and GMM, respectively. The number of misclassified images in the case of the GID mixture was 71, which corresponds to an accuracy of 79.32% as compared to the 73.22% obtained with the GMM.

Table 2.9: The confusion matrix in the case of the visual scenes clustering problem when applying GID mixture with feature selection and outlier detection.

	living room	coast	forest	mountain	tall building	
living room	52	0	1	2	1	0
coast	0	42	0	14	0	0
forest	0	0	54	1	1	0
mountain	0	2	14	34	0	0
tall building	20	0	0	1	37	0
outlier class	0	0	0	0	0	15

Table 2.10: The confusion matrix in the case of the visual scenes clustering problem when applying GMM with feature selection and outlier detection.

	living room	coast	forest	mountain	tall building	
living room	41	2	5	3	5	0
coast	1	45	0	7	0	1
forest	0	0	51	3	1	0
mountain	4	0	18	32	0	2
tall building	19	0	2	1	34	0
outlier class	0	2	1	0	0	12



(a) GID

(b) GMM

Figure 2.7: Features saliencies obtained in the case of the visual scenes clustering problem when performing simultaneous feature selection and outlier detection using (a) GID mixture and (b) GMM.

Objects Clustering

In this application, we focused on the object clustering problem which is prevalent in many real-life applications such as object detection and recognition, automated visual inspection, and video surveillance [84]. Indeed, our proposed model was evaluated using ETH-80 [5] data set from which we considered four categories (pear, car, cup, and dog), each of which contains 410 images that were cropped, so that they contained only the object without any border area. In addition, they were resized to a size of 128 × 128 pixels. Examples of objects from each category are displayed

[5] http://www.d2.mpi-inf.mpg.de/Datasets/ETH80

in Fig. 2.8. The same texture images used in the previous application (see Figure 2.5) were appended to this objects data set and were considered as outliers. As in the previous application, we considered HOG features with 3 windows to describe our images.

Tables 2.11 and 2.12 show the confusion matrices for this data set when applying the GID mixture and the GMM without performing neither feature selection nor outlier detection. According to these matrices, the number of misclassified objects in the case of the GID mixture was 77 (i.e. an accuracy of 77.55%), and the number of misclassified objects in the case of the GMM was 85

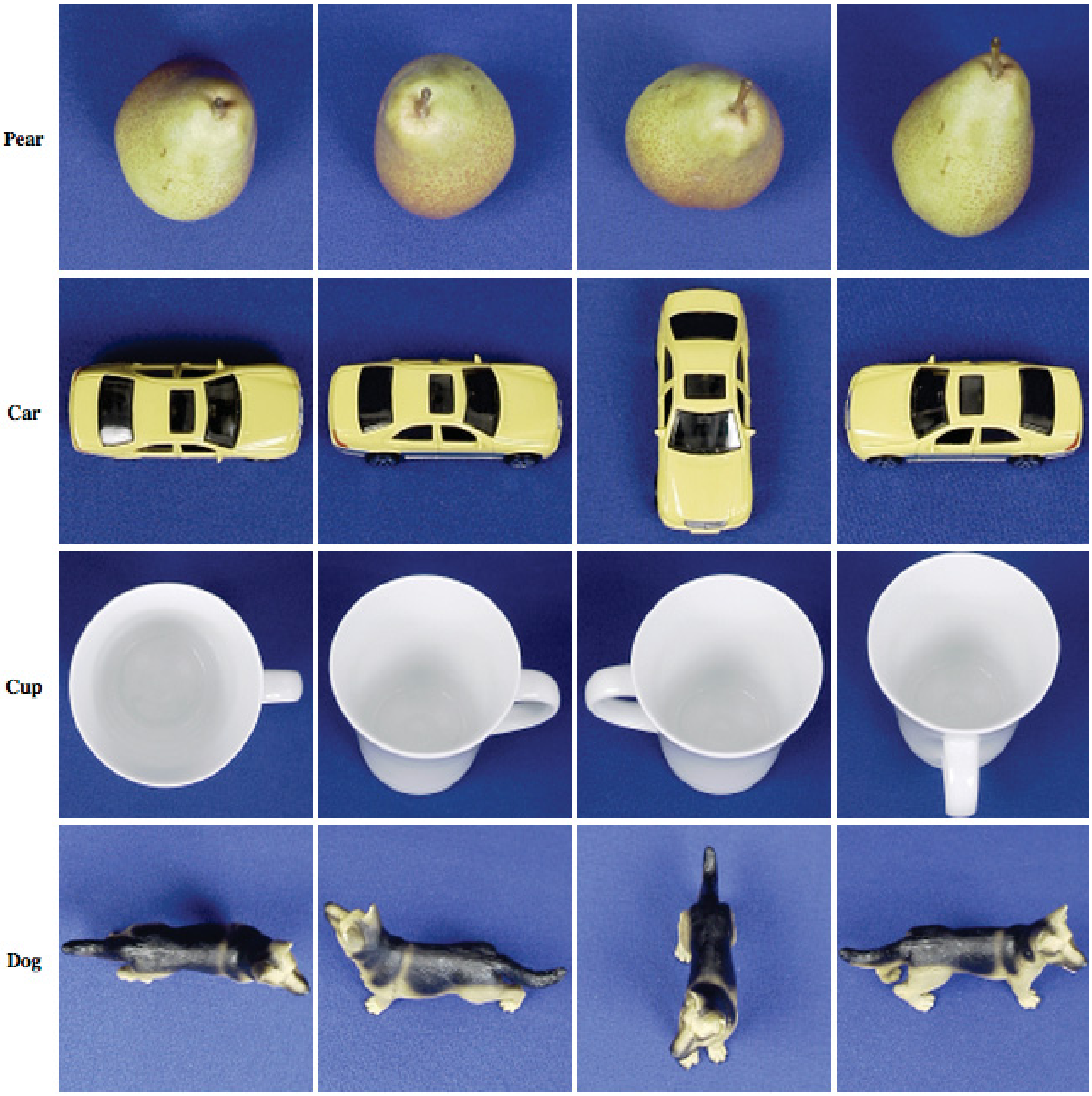


Figure 2.8: Examples of objects from the four considered categories.

(i.e. an accuracy of 75.22%). Both models were not able to detect the outlying images which were distributed in different classes. We have also investigated the case where feature selection was performed without outlier detection as shown in Tables 2.7 and 2.14. We obtained classification accuracies of 79.88% and 76.97% for the GID mixture and GMM, respectively. Figure 2.9 shows the feature saliencies resulting from both models. The classification accuracies have clearly increased, but again both approaches were unable to cluster the outliers into a different class.

Table 2.11: Confusion matrix for the objects clustering application using GID mixture.

	pear	car	cup	dog
pear	73	0	0	9
car	0	54	0	28
cup	0	12	61	9
dog	2	2	0	78
outlier class	0	0	3	12

Table 2.12: Confusion matrix for the objects clustering application using GMM.

	pear	car	cup	dog
pear	74	0	0	8
car	0	56	0	26
cup	2	14	53	13
dog	4	3	0	75
outlier class	0	14	1	0

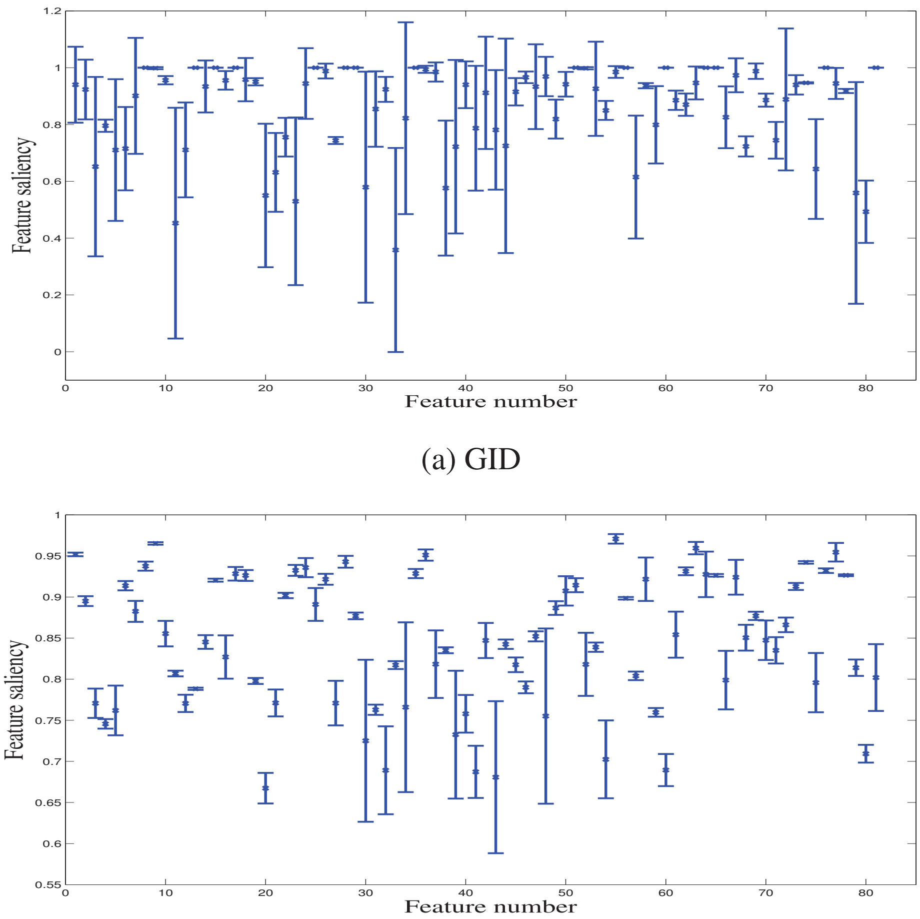
Table 2.13: Confusion matrix for the objects clustering application using GID mixture with feature selection.

	pear	car	cup	dog
pear	75	0	0	7
car	0	57	0	25
cup	0	12	63	7
dog	1	2	0	79
outlier class	0	0	3	12

In the last experiment, we considered the scenario where feature selection (see Figure 2.10) and outlier rejection have been performed simultaneously. Both mixture models were able to

Table 2.14: Confusion matrix for the objects clustering application using GMM with feature selection.

	pear	car	cup	dog
pear	76	0	0	6
car	0	58	0	24
cup	0	17	55	10
dog	4	3	0	75
outlier class	0	4	0	11

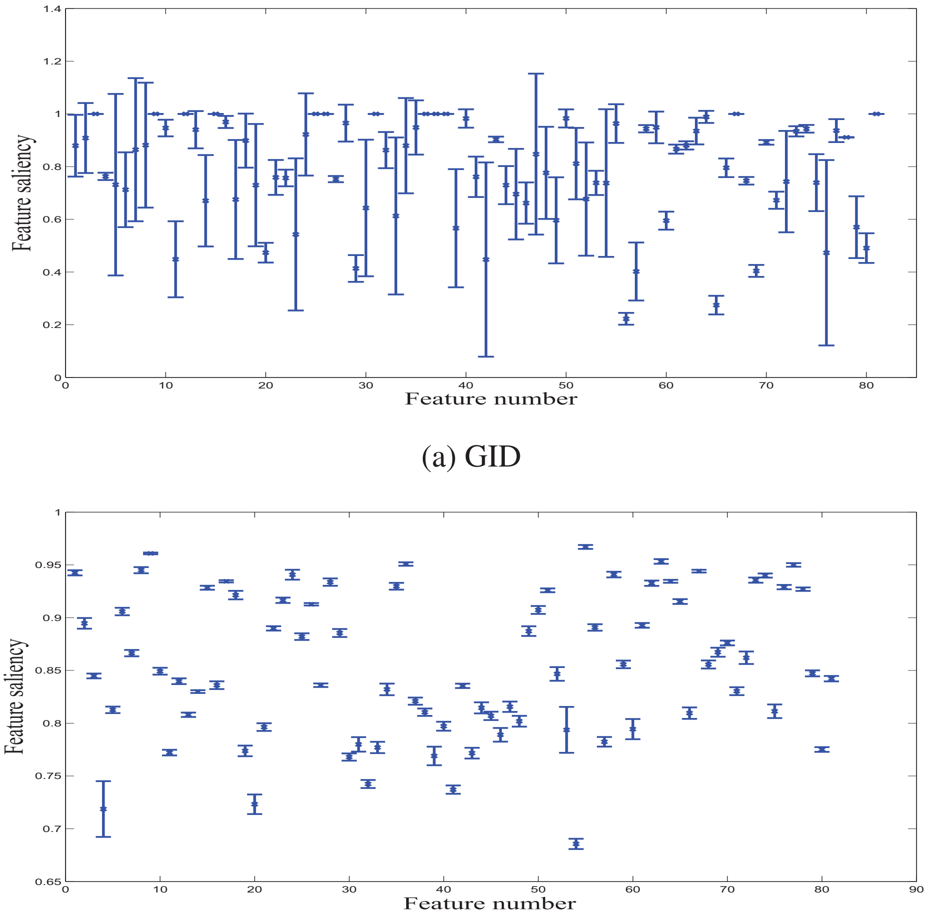


(a) GID

(b) GMM

Figure 2.9: Features saliencies obtained in the case of the object clustering problem when performing feature selection without outlier detection using (a) GID mixture and (b) GMM.

detect the outlier data and the exact number of categories, as shown in Tables 2.15 and 2.16, with accuracies of 85.71% and 83.96%, respectively.



(a) GID

(b) GMM

Figure 2.10: Features saliencies obtained in the case of the object clustering problem when performing feature selection with outlier detection using (a) GID mixture and (b) GMM.

Table 2.15: Confusion matrix for the objects clustering application using GID mixture with simultaneous feature selection and outlier detection.

	pear	car	cup	dog	outlier class
pear	76	0	0	6	0
car	0	64	0	18	0
cup	0	15	60	7	0
dog	0	3	0	79	0
outlier class	0	0	0	0	15

2.5 Summary

Finite mixture offers a formal approach to clustering and a powerful tool to tackle the problem of data modeling. It is often applied in signal and image processing, as well as machine learning

Table 2.16: Confusion matrix for the object clustering application using GMM with simultaneous feature selection and outlier detection.

	pear	car	cup	dog	outlier class
pear	78	0	0	4	0
car	1	63	0	14	5
cup	3	15	61	6	0
dog	6	5	0	71	0
outlier class	0	0	0	0	15

and data mining applications. Some special concerns when considering mixture models are the choice of the components densities, determining the number of components, the selection of the relevant features, and the detection of potential outliers. In this chapter, we introduce a statistical framework in which all these problems are addressed simultaneously in the case of non-Gaussian data, specifically, positive data. The developed statistical framework was based on the finite GID mixture model and learned via the minimization of a message length objective by deploying an EM framework. The strength of our learning approach is that it allows a compromise between goodness of fit and a model's complexity. Empirical results which involve generated data as well as real applications concerning visual scenes and objects classification, show that the proposed approach is promising.

In the next chapter, we investigate the development of a purely Bayesian inference alternative, for the learning of finite generalized inverted Dirichlet mixture. The main goal is to overcome problems related to local convergence, dependency on initialization, and model selection.

Chapter 3

Bayesian Learning of Finite Generalized Inverted Dirichlet Mixtures

Our focus in this chapter is to develop a Bayesian framework which is based on the generalized inverted Dirichlet distribution for modeling positive data. The proposed mixture model is subjected to a fully Bayesian analysis based on Markov chain Monte Carlo (MCMC) simulation methods. These methods, the Gibbs sampling and Metropolis-Hastings, were used to compute the posterior distribution of the parameters. The Bayesian information criterion (BIC) was used for model selection. The adoption of this purely Bayesian learning choice was motivated by the fact that the Bayesian inference handle uncertainty in a unified and consistent manner. We evaluated our approach on the basis of generated data and two challenging applications concerning object classification and forgery detection.

3.1 Introduction

Learning finite mixture models involves the following two important tasks: estimation and selection. Mixture estimation refers to the estimation of parameters given the data. The maximum likelihood (ML) has been widely used to perform this task. It is generally implemented using the expectation maximization (EM) algorithm which treats the estimation as a special case of estimation when using incomplete data. The EM algorithm and its several extensions attempt to estimate a single "best" model, which is not always realistic since the data may suggest many "good" models. Moreover, the EM has some drawbacks, like convergence to local maxima due to its dependence on the initialization step [81]. An efficient alternative technique, that we shall adopt in this work, in the case of generalized inverted Dirichlet mixture, is to consider the average result computed over several models. This can be done through Bayesian approaches that have become a popular choice for computer vision applications in general [85] and the estimation of mixture models in particular [86].

It is well-known that the Bayesian inference provides better generalization capabilities, when learning mixture models, than the ML method, that tends to generally under- or over-fit the data. Unlike frequency approaches in general, and ML in particular, which can learn from a single model, Bayesian approaches can compute distributions over all possible parameter values with respect to the posterior distribution [87]. It can also automatically implement Ockhams Razor by integrating out irrelevant variables in a given model. This permits it to give preference for simple models that sufficiently explain the data without additional unnecessary complexity. Markov Chain Monte Carlo (MCMC) methods have been used extensively in computational statistics, over the last few years, to perform Bayesian inference [88–90]. For an in-depth overview of MCMC techniques, the reader is referred to [91]. Model selection is a pivotal issue in mixture-based modeling. This subject has been under extensive study over the last few years. Within a Bayesian setting model, this problem had been handled using several approaches [92] such as the reversible jump sampler [21], Bayesian deviance criterion [93], Bayes factor approximation [94], and Birth-and-Death MCMC [95]. In this chapter, we consider the Bayes factor for the selection of finite generalized inverted Dirichlet mixture models, which has become a well established powerful model selection tool.

The following is a brief overview of the chapter. In Section 3.1, the generalized inverted Dirichlet mixture model is briefly introduced. In Section 3.2, we motivate and develop the priors and posteriors needed in our learning framework. The complete learning approach is presented in Section 3.3. In Section 3.4, we experimentally examine the effectiveness of our model. Finally, in Section 3.5, a summary of this chapter is given.

3.2 Finite Generalized Inverted Dirichlet Mixture Model

3.2.1 The Mixture Model

Let $p(\vec{X}|\vec{\theta})$ be a density function of a D-dimensional inverted generalized Dirichlet distribution whose parameters are $\vec{\theta} = (\alpha_1, \beta_1, \ldots, \alpha_D, \beta_D)$. An inverted generalized Dirichlet mixture $p(\vec{X}|\Theta)$ of a D-dimensional vector $\vec{X} \in \mathbb{R}_+^D$ is defined by:

$$p(\vec{X}|\Theta) = \sum_{j=1}^{M} p_j p(\vec{X}|\vec{\theta}_j) \tag{3.1}$$

where M is an integer which represents the number of components and $\{p_j\}$ is the set of mixing parameters which are positive and sum to one. $\Theta = \{p_j, \vec{\theta}_j = (\alpha_{j1}, \beta_{j1}, \ldots, \alpha_{jD}, \beta_{jd})$
$\}_{j=1}^{M}$ represents the set of the model's parameters and $p(\vec{X}|\vec{\theta}_j)$ is given by [63]:

$$p(\vec{X}|\vec{\theta}_j) = \prod_{d=1}^{D} \frac{\Gamma(\alpha_{jd} + \beta_{jd})}{\Gamma(\alpha_{jd})\Gamma(\beta_{jd})} \frac{X_d^{\alpha_{jd}-1}}{(1 + \sum_{l=1}^{d} X_l)^{\gamma_{jd}}} \tag{3.2}$$

where $\gamma_{jd} = \beta_{jd} + \alpha_{jd} - \beta_{jd+1}$ for $d = 1, \ldots, D$ with $\beta_{jD+1} = 0$. It is noteworthy that it is straightforward to verify that the generalized inverted Dirichlet (GID) has a more general covariance structure than the inverted Dirichlet. Furthermore, it can be reduced to an inverted Dirichlet with parameters $(\alpha_{j1}, \ldots, \alpha_{jD}, \beta_{j1})$ if we set $\gamma_{j1} = \gamma_{j2} = \ldots = \gamma_{jD-1} = 0$ [63].

Let $\mathscr{X} = (\vec{X}_1, \ldots, \vec{X}_N)$ be a set of N independently and identically distributed vectors taken from our mixture model. By introducing hidden vectors $\mathscr{Z} = \{\vec{Z}_1, \ldots, \vec{Z}_N\}$, where $\vec{Z}_i = (Z_{i1}, \ldots, Z_{iM})$, representing the components from which each vector $\vec{X}_i$ is generated such that $Z_{ij} = 1$ if $\vec{X}_i$ is generated from component j, and is equal to 0, otherwise (i.e. $\sum_{j=1}^{M} Z_{ij} = 1$) We can then write the likelihood corresponding to the GID mixture model as follows:

$$p(\mathscr{X}, \mathscr{Z}|\Theta) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[p_j p(\vec{X}_i|\vec{\theta}_j)\right]^{Z_{ij}} \tag{3.3}$$

3.2.2 Bayesian Inference

Bayesian inference has been used successfully for many problems where the main goal is to infer the parameters of a model of interest. It is widely used to handle missing data [96] problems in general and finite mixture learning in particular. In Bayesian learning of finite mixture models, we first need to set the prior distribution $p(\Theta)$ on the mixture parameters. Then, the posterior distribution is computed from the data and the prior is selected as follows:

$$p(\Theta|\mathscr{X}, \mathscr{Z}) = \frac{p(\mathscr{X}, \mathscr{Z}|\Theta)p(\Theta)}{\int p(\mathscr{X}, \mathscr{Z}|\Theta)p(\Theta)\partial\theta} \propto p(\mathscr{X}, \mathscr{Z}|\Theta)p(\Theta) \tag{3.4}$$

The denominator in the previous equation, $\int p(\mathscr{X}, \mathscr{Z}|\Theta)p(\Theta)\partial\theta$, is actually a normalization constant known as the marginal likelihood (or Bayesian evidence). Having this posterior distribution in hand, the learning problem is now transformed into one of simulating parameters from the posterior distribution $\Theta \sim p(\Theta|\mathscr{X}, \mathscr{Z})$.

Developing algorithms for generating observations from a posterior distribution has been one of the most active areas in statistical computing [97]. MCMC techniques have revolutionized Bayesian inferences and the algorithm of choice is the Gibbs sampler among Bayesian statisticians. Gibbs sampling begins with a random configuration which is then updated over iterations by taking advantage of the missing data. That is, to associate with each observation $\vec{X}_i$ a missing multinomial variable $\vec{Z}_i \sim \mathscr{M}(1; \hat{Z}_{i1}, \ldots, \hat{Z}_{iM})$, where:

$$\hat{Z}_{ij} = \frac{p(\vec{X}_i|\theta_j)p_j}{\sum_{j=1}^{M} p(\vec{X}_i|\theta_j)p_j} \tag{3.5}$$

As indicated by $p(\vec{p}|\mathscr{Z})$, where $\vec{p} = (p_1, \dots, p_M)$, is the density of the distribution of $\vec{p}$ given $\mathscr{Z}$, and $p(\vec{\theta}_j|\mathscr{Z}, \mathscr{X})$, is the density of the distribution of $\vec{\theta}_j$ given $\mathscr{Z}$ and $\mathscr{X}$. The standard Gibbs sampler for mixture models is based on the successive simulation of $\mathscr{Z}$, $\vec{p}$ and $\vec{\theta}_j$. As a result, the general Gibbs sampling for mixture models is as follows [91, 98]:

1. Initialization

2. Step t: For t=1,...

   (a) Generate $\vec{Z}_i^{(t)} \sim \mathscr{M}(1; \hat{Z}_{i1}^{(t-1)}, \dots, \hat{Z}_{iM}^{(t-1)})$

   (b) Generate $\vec{p}^{(t)}$ from $p(\vec{p}|\mathscr{Z}^{(t)})$

   (c) Generate $\vec{\theta}_j^{(t)}$ from $p(\vec{\theta}_j|\mathscr{Z}^{(t)}, \mathscr{X})$

3.3 Model Learning

One aspect of Bayesian modeling is that it is typically difficult, yet is crucial in influencing the chosen results of priors. The prior distributions can be viewed as our prejudice on the model's parameters. This prior parameter values can be improved in light of the information provided by the data [99]. In the following, we specify the priors, as well as the resulting posteriors, that we consider in our Bayesian framework.

3.3.1 Priors and Posteriors

A standard choice as a prior for the mixing parameters vector $\vec{p}$ is the Dirichlet distribution, since it is defined on the simplex $\{(p_1, \dots, p_M) : \sum_{j=1}^{M-1} p_j < 1\}$ [91]:

$$p(\vec{p}|\vec{\eta}) = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} p_j^{\eta_j - 1} \tag{3.6}$$

where $\vec{\eta} = (\eta_1, \dots, \eta_M)$ is the parameter vector of the Dirichlet distribution. Moreover, we have:

$$p(\mathscr{X}|\vec{p}) = \prod_{i=1}^{N} p(\vec{Z}_i|\vec{p}) = \prod_{i=1}^{N} p_1^{Z_{i1}} \dots p_M^{Z_{iM}} = \prod_{i=1}^{N} \prod_{j=1}^{M} p_j^{Z_{ij}} = \prod_{j=1}^{M} p_j^{n_j} \tag{3.7}$$

where $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_{ij}=1}$. Hence,

$$\begin{aligned} p(\vec{p}|\mathscr{X}) &\propto \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} p_j^{\eta_j - 1} \prod_{j=1}^{M} p_j^{n_j} = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} p_j^{\eta_j + n_j - 1} \\ &\propto \mathscr{D}(\eta_1 + n_1, \dots, \eta_M + n_M) \end{aligned} \tag{3.8}$$

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

In order to specify the priors for the $\vec{\theta}_j$ parameters, we consider an effective strategy that relies on the fact that the GID belongs to the exponential family of distributions. In fact, if a S-parameter density p belongs to the exponential family, then we can write it as following [100]:

$$p(\vec{X}|\theta) = H(\vec{X}) \exp(\sum_{l=1}^{S} G_l(\theta) T_l(\vec{X}) + \Phi(\theta)) \tag{3.9}$$

In this case, a conjugate prior on θ is given by [100]:

$$p(\theta) \propto \exp(\sum_{l=1}^{S} \rho_l G_l(\theta) + \kappa \Phi(\theta)) \tag{3.10}$$

where $\rho = (\rho_1, \dots, \rho_S) \in \mathbb{R}^S$ and $\kappa > 0$ are referred to as hyperparameters. The GID distribution can be written as an exponential density (See Appendix E):

$$\begin{aligned} p(\vec{X}|\vec{\theta}_j) = \exp\Bigg[&\sum_{d=1}^{D} \Big(\log\big(\Gamma(\alpha_{jd} + \beta_{jd})\big) - \log\big(\Gamma(\alpha_{jd})\big) - \log\big(\Gamma(\beta_{jd})\big) \Big) \\ &+ \sum_{d=1}^{D} \Big((\alpha_{jd} - 1) \log(X_d) \Big) - \sum_{d=D+1}^{2D-1} \Big((\beta_{jd-D} + \alpha_{jd-D} - \beta_{jd-D+1}) \\ &\times \log(1 + \sum_{l=1}^{d-l} X_l) \Big) - (\alpha_{jD} + \beta_{jD}) \log(1 + \sum_{l=1}^{D} X_l) \Bigg] \end{aligned} \tag{3.11}$$

Then, by letting

$$S = 2D$$

$$\Phi(\theta_j) = \sum_{d=1}^{D} \Big(\log\big(\Gamma(\alpha_{jd} + \beta_{jd})\big) - \log\big(\Gamma(\alpha_{jd})\big) - \log\big(\Gamma(\beta_{jd})\big) \Big)$$

$$G_d(\theta_j) = \alpha_{jd}, \quad d = 1, \dots, D$$

$$G_d(\theta_j) = \beta_{jd-D} + \alpha_{jd-D} - \beta_{jd-D+1}, \quad d = D+1, \dots, 2D-1$$

$$G_{2D}(\theta_j) = \alpha_{jD} + \beta_{jD}$$

$$T_d(\vec{X}) = \log(X_d), \quad d = 1, \dots, D$$

$$T_d(\vec{X}) = -\log(1 + \sum_{l=1}^{d-D} X_l), \quad d = D+1, \dots, 2D$$

$$H(\vec{X}) = \exp\big(- \sum_{d=1}^{D} \log(X_d) \big)$$

The prior is defined as:

$$\begin{aligned} p(\theta_j) \propto \exp\Bigg[& \sum_{d=1}^{D} \rho_d \alpha_{jd} + \sum_{d=D+1}^{2D-1} \rho_d \big(\beta_{jd-D} + \alpha_{jd-D} - \beta_{jd-D+1} \big) \\ & + \kappa \sum_{d=1}^{D} \Big(\log\big(\Gamma(\alpha_{jd} + \beta_{jd})\big) - \log\big(\Gamma(\alpha_{jd})\big) - \log\big(\Gamma(\beta_{jd})\big) \Big) + \rho_{2D}(\alpha_{jD} + \beta_{jD}) \Bigg] \\ \propto \exp\Bigg[& \kappa \sum_{d=1}^{D} \Big(\log\big(\Gamma(\alpha_{jd} + \beta_{jd})\big) - \log\big(\Gamma(\alpha_{jd})\big) - \log\big(\Gamma(\beta_{jd})\big) \Big) \\ & + \sum_{d=1}^{D} \rho_d \alpha_{jd} + \sum_{d=1}^{D} \rho_{d+D} \gamma_{jd} \Bigg] \end{aligned} \tag{3.12}$$

The prior hyperparameters are: $(\rho_1, \dots, \rho_{2d}, \kappa)$. Having this prior, $p(\theta_j)$, the posterior distribution is then (See Appendix F):

$$
\begin{aligned}
p(\theta_j|\mathscr{Z},\mathscr{X}) \propto p(\theta_j)\prod_{Z_{ij}=1} p(\vec{X}_i|\theta_j) \propto \exp\Bigg[\sum_{d=1}^{D}\alpha_{jd}\Big(\rho_d+\sum_{Z_{ij}=1}\log(X_{id})\Big) \\
+\sum_{d=1}^{D}\gamma_{jd}\Big(\rho_{d+D}-\sum_{Z_{ij}=1}\log(1+\sum_{l=1}^{d}X_{il})\Big) \\
+(\kappa+n_j)\sum_{d=1}^{D}\Big(\log\big(\Gamma(\alpha_{jd}+\beta_{jd})\big)-\log\big(\Gamma(\alpha_{jd})\big)-\log\big(\Gamma(\beta_{jd})\big)\Big)\Bigg]
\end{aligned}
\tag{3.13}
$$

According to Eqs. 3.12 and 3.13, we can clearly see that the posterior and the prior distributions have the same form. Therefore, $p(\theta_j)$ is really a conjugate prior on θ_j. In contrast to $p(\vec{p}|\mathscr{Z})$, $p(\theta_j|\mathscr{Z},\mathscr{X})$ is not standard and does not have a known form. Thus, we cope with the simulation from this distribution by considering a random-walk Metropolis-Hastings (M-H) algorithm [91].

3.3.2 Model Selection and Convergence

The model selection problem is a challenging one which does not have any completely satisfactory solutions. Suppose that there exists a set of candidate models, we need to find the "best" model that well describes the data without under- or over-fitting (i.e. with good generalization capability) [101]. The main goal here is to make a successful trade-off between the complexity and the goodness of fit. Bayes factor has been widely used in Bayesian inference [102–104] for model selection and we shall adopt it here in the case of our mixture model. The main idea is to consider the number of components for which the following approximation to the marginal likelihood is maximum:

$$
\log(p(\mathscr{X}|M)) = \log(p(\mathscr{X}|\hat{\Theta},M)) - \frac{N_p}{2}\log(N) \tag{3.14}
$$

which is the Bayesian information criterion (BIC) proposed by Schwarz [17], where $p(\mathscr{X}|\hat{\Theta},M)$ is the likelihood function taking into account that the number of components is M. N_p is the number of free parameters to be estimated and is equal to $(2d+1)M-1$, in our case. $\hat{\Theta}$ denotes the posterior mode.

The convergence of MCMC techniques, and in particular Gibbs sampling and Metropolis Hastings, has been widely studied in the literature [105]. Several systematic approaches for establishing convergence of MCMC have been proposed and one of them that we follow is the diagnostic approach proposed by Raftery and Lewis [106, 107], which has been shown to often work well in practice. This approach is based on a single long-run of the Gibbs sampler.

3.4 Experimental Results

3.4.1 Design of Experiments

In this section, extensive experiments are conducted in order to evaluate the benefits of using the proposed model. Our experiments involved synthetic data sets as well as a real-world challenging applications that concern object classification and forgery detection. The main goal of the application on synthetic data is to show the merits of the proposed model (Bayesian Generalized Inverted Mixture (B-GIDM)) and its ability to efficiently perform parameter estimation and model selection. On the other hand, for the real-world applications, the first goal of our experiments was to compare both Bayesian estimation and maximum likelihood estimation, as proposed in [44], for the GID mixture that we denote as B-GIDM and the ML-GIDM, respectively. The second goal was to compare the performance of the GID mixture with the inverted Dirichlet mixture (denoted as B-IDM) and the Gaussian mixture (denoted as B-GM) when learned in a Bayesian way. It is noteworthy that, in our applications, we held the hyperparameters η_j fixed at 1, which is a classical and a reasonable choice [91]. Moreover, according to the posterior hyperparameters,

$$\Big(\rho_1 + \sum_{Z_{ij}=1} \log(X_{i1}), \ldots, \rho_D + \sum_{Z_{ij}=1} \log(X_{iD}), \rho_{D+1} - \sum_{Z_{ij}=1} \log(1 + \sum_{t=1}^{1} X_{it}), \ldots,$$

$$\rho_{2D} - \sum_{Z_{ij}=1} \log(1 + \sum_{t=1}^{D} X_{it}), \kappa + n_j\Big)$$

the prior hyperparameters were modified by adding $T_l(\vec{X})$ or n_j to the previous values. This information, could be used to get the prior hyperparameters. Indeed, following [108,109], once the sample $\mathscr{X}$ is known, we can use it to get the prior hyperparameters. Then, we held $(\eta_1,\ldots,\eta_M)$ and $(\rho_1,\ldots,\rho_{2d},\kappa)$ fixed at: $\eta_j = 1$, $j = 1,\ldots,M$, $\rho_l = \sum_{i=1}^{N}\log(X_{il})$, $\rho_{l+d} = -\sum_{i=1}^{N}\log(1+\sum_{t=1}^{l}X_{it})$, $l = 1,\ldots,d$, $\kappa = n_j$.

3.4.2 Evaluation on Synthetic data

The reason for using synthetic data was to evaluate the performance of the proposed B-GID in terms of estimation and selection on four two-dimensional synthetic data sets. Table 3.1 shows the real and the estimated parameters for the four generated data sets. According to this table, the parameter of the model and its mixing coefficients were accurately estimated by B-GID. In addition, using BIC, the model was able to be correctly identified.

Table 3.1: Parameters of the different generated data sets. $\alpha_{j1},\beta_{bj1},\alpha_{j2},\beta_{bj2}$ and p_j are the real parameters. $\hat{\alpha}_{j1},\hat{\beta}_{bj1},\hat{\alpha}_{j2},\hat{\beta}_{bj2}$ and $\hat{p}_j$ are the estimated parameters

	j	α_{j1}	β_{j1}	α_{j2}	β_{j2}	p_j	$\hat{\alpha}_{j1}$	$\hat{\beta}_{j1}$	$\hat{\alpha}_{j2}$	$\hat{\beta}_{j2}$	$\hat{p}_j$
Data set 1	1	45	50	22	25	0.50	45.89	50.81	20.01	23.26	0.53
(N=600)	2	10	30	16	50	0.50	9.75	29.10	15.85	52.10	0.47
Data set 2	1	45	50	22	25	0.33	44.93	46.27	22.01	23.70	0.34
(N=900)	2	10	30	16	50	0.33	10.86	32.46	17.21	51.24	0.32
	3	20	4	38	5	0.33	20.53	3.79	35.15	4.43	0.34
Data set 3	1	45	50	22	25	0.30	44.97	50.69	25.22	28.49	0.31
(N=1000)	2	10	30	16	50	0.30	10.93	31.05	16.21	48.58	0.30
	3	20	4	38	5	0.20	23.10	4.72	44.93	6.03	0.21
	4	8	58	6	40	0.20	7.76	56.77	6.26	41.46	0.18
Data set 4	1	45	50	22	25	0.25	42.45	46.79	23.59	26.60	0.25
(N=1200)	2	10	30	16	50	0.25	9.86	27.98	18.36	54.30	0.25
	3	20	4	38	5	0.25	20.33	3.93	38.78	5.14	0.27
	4	8	58	6	40	0.25	7.99	56.33	6.84	42.29	0.23

3.4.3 Object Classification

Object classification and detection (i.e. distinguishing target objects from nontarget objects) have been the topic of extensive research in the past and are crucial steps in several applications such as object recognition, content-based image retrieval, and video surveillance [110–116]. Object classification task is performed by humans effortlessly, which is not the case for machines that have to be able to achieve good detection despite variable illumination conditions, orientations, positions and scales [117–119]. Although different, research efforts can be classified into two groups of approaches. The first one is concerned with the development of a robust global, such as color and texture [120], or local visual descriptors. The approaches in the second group have been devoted to the development of powerful classifiers.

Many global and local visual descriptors have been proposed in the past. Here, we use a local Histogram of Oriented Gradient (HOG) descriptor [83] which generates positive features and which we found efficient and convenient by being robust to partial occlusion and image noise [121]. Experiments are conducted by considering three windows for the HOG descriptor which allows to represent each image by an 81-dimensional vector of features. We tested our mixture model for the detection of several complex objects by considering the following data sets in our experiments. The first one is the "INRIA horses" data set [116, 122] which contains 170 positive examples (i.e. images containing horses) and 170 negative images without horses. 50 positive images and 50 negative ones have been used for training and the remaining 120 positive and 120 negative images have been used for testing. Figure 3.1 shows examples from the two groups of images. The second data set is the "Weizmann-Shotton horses" data set which is composed of 327 positive images and 327 negative ones. As for the first data set, 50 positive images and 50 negative negative images have been used for training and the rest for testing.

Figure 3.2 shows examples from the two groups of images in this data set. The third data set is the "ETHZ shape classes" which is composed of five classes of objects (bottles, swans, mugs, giraffes, and apple logos) for a total of 255 images collected from the Web and described in [123]. This data set is particularly challenging when taking into account the wide range of scales, shape

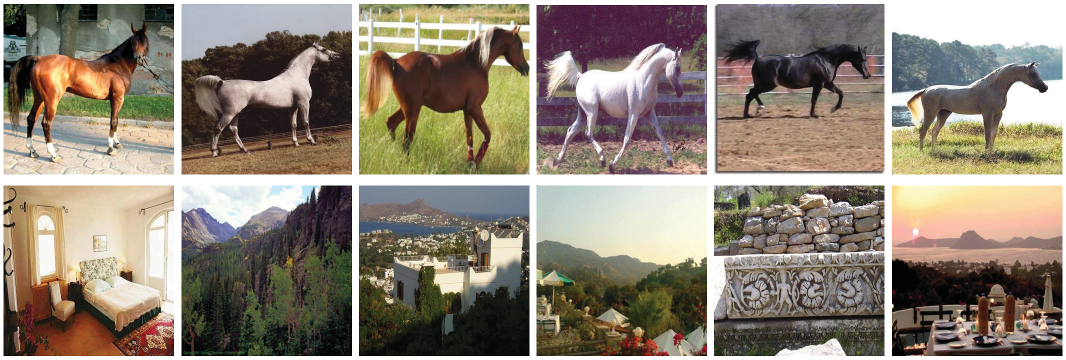


Figure 3.1: Sample images from "INRIA horses" data set. First row: positive examples. Second row: negative examples.

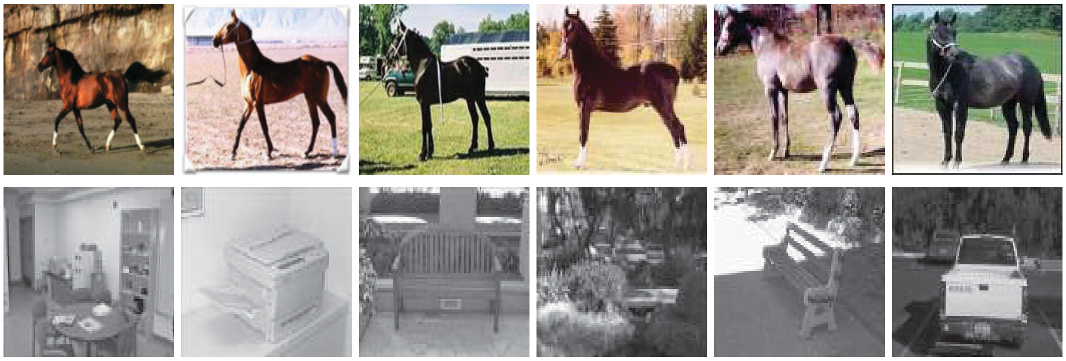
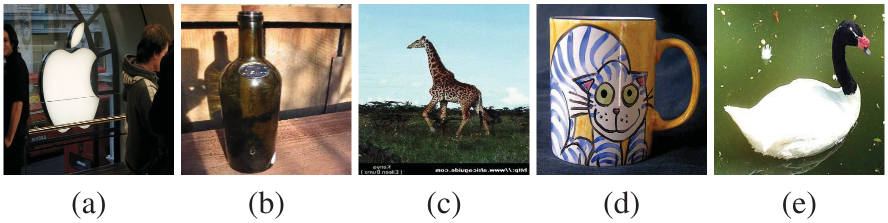


Figure 3.2: Sample images from Weizmann-Shotton horses data set. First row: positive examples. Second row: negative examples.

variations, and clutter that characterize the images it contains. Figure 3.3 displays examples of images from the different classes in this data set. Following [116], we trained one detector per class for this data set by using the first half of the positive images (40 apple logos, 48 bottles, 87 giraffes, 48 mugs, and 32 swans). The negative training images for each class in this data set were taken from the other four remaining classes where each class contributes 1/4 of the total number of images. For instance, the negative images for the class mugs were built by taking 12 images from each of the other four classes.

The detection process, in our case, was based on two steps. The first step was training which allowed the representation of each class as a finite GID mixture. For the second, classification step, each test image was assigned to the class maximizing more its log-likelihood. It is noteworthy that comparing our results to those obtained with all recent state of the art approaches, proposed for



(a) (b) (c) (d) (e)

Figure 3.3: Sample images from ETHZ shape classes data set. (a) apple logos, (b) bottles, (c) giraffes, (d) mugs, and (e) swans.

the considered data sets, is out of the scope of this work. Indeed, the main goal is to validate the approach by considering comparable mixture-based techniques. Table 3.2 summarizes the classification accuracies for the different considered data sets. According to this table, it is clear that the B-GIDM provides the best classification rates as compared to the other test models. Moreover, we can see that the inverted Dirichlet mixture performs better than the Gaussian. Future work, in the case of this application, could be devoted to the consideration of contextual information as performed in [121] to further improve the results.

Table 3.2: Classification rates (in %) for the different tested data sets using different approaches.

	B-GIDM	ML-GIDM	B-IDM	B-GM
INRIA horses	73.78	72.11	71.02	70.33
Weizmann-Shotton horses	71.01	70.42	69.56	68.77
Apple logo	84.98	84.05	83.69	81.60
Bottle	24.73	24.32	24.03	23.86
Giraffe	48.41	48.11	47.73	74.07
Mug	80.95	79.77	78.59	77.90
Swan	83.06	82.12	81.40	79.95

3.4.4 Forgery Detection

Forgery (e.g. cloning, resampling, splicing, etc.) detection is currently one of the most active research topics in image processing [124]. It can be viewed as the problem of reliably distinguishing between "doctored" images and untampered original ones. Indeed, with the advent of sophisticated multimedia editing software, that allow the manipulation of images for instance, the integrity of image content cannot be taken as granted [125–127]. This strong interest is driven by a wide spectrum of promising applications in many areas such as forensics and security (e.g. an image can contain hidden information), and journalism (e.g. using images as critical evidence) to name a few. Several techniques have been developed in an attempt to identify forged images, many of which are reviewed in [124]. Other works include the approaches in [125, 128, 129].

In this section, we apply our model to this challenging problem by focusing on the specific case of copy-move attack detection which has received some attention recently [130–135]. We follow the approach proposed in [134], which allows simultaneously to detect copy-move attacks (i.e. delete some objects from the scene and substitute them with other parts from the same scene) and to recover the geometric transformation used in cloning. This approach has been shown to be efficient. It is based on scale invariant features transform (SIFT) [136], which has been widely used in several image forensics applications [132, 133, 137], to detect points belonging to cloned areas. The choice of SIFT descriptors is motivated by the fact that they are invariant to changes in illumination, rotation, scaling and robust to occlusion and clutter. To summarize, the overall approach proceeds along three central themes. First, SIFT features are extracted and multiple keypoint matching is performed. The second theme concerns keypoint clustering and forgery detection. Then, an existing geometric transformation is estimated if a tampering is detected. Our contribution here is directed toward the second theme by applying our GID mixture model for clustering. Thus, we present our results by focusing on the detection accuracy since estimating the geometric transformation is clearly out of the scope of this work. However, it can easily be done using the approach in [134]. Indeed, we integrated the GID mixture in the agglomerative hierarchical clustering scheme used in [134] to obtain the initial clusters. The reader is referred to [134] for more details and discussions.

We considered the two data sets introduced in [134]. The first one is the MICC-F220 data set which consists of 220 images: 110 are tampered images, using 10 different attacks, and 110 are original with different resolutions varying from 722×480 to 800×600 pixels. In this data set, the size of the forged parts covers, on average, 1.2% of the image. The second data set is the MICC-F2000 data set which is composed of 2000 2048 × 1536 images (1300 original images and 700 tampered ones using 14 different attacks). In this second data set, the size of the forged parts covers on average 1.12% of an image. We follow the same experimental settings described in [134] and the same detection performance measurements. These measurements are: true positive rate (TPR), which represents the fraction of correctly identified tampered images, and false positive rate (FPR), which represents the fraction of original images not correctly identified:

$$TPR = \frac{\textit{Number of images detected as forged being forged}}{\textit{Total number of forged images}} \tag{3.15}$$

$$FPR = \frac{\textit{Number of images detected as forged being original}}{\textit{Total number of original images}} \tag{3.16}$$

Tables 3.3 and 3.4 display the results obtained for both MICC-F220 and MICC-F2000, respectively, when considering B-GIDM, ML-GIDM, B-IDM, and B-GM. The obtained results show again that the B-GIDM performs well as compared to the other tested approaches.

Table 3.3: Detection results, during testing phase, in terms of FPR and TPR in the case of the MICC-F220 data set using different clustering approaches.

	B-GIDM	ML-GIDM	B-IDM	B-GM
$FPR(\%)$	8.10	8.21	8.23	8.39
$TPR(\%)$	98.13	97.91	97.14	96.64

Table 3.4: Detection results, during testing phase, in terms of FPR and TPR in the case of the MICC-F2000 data set using different clustering approaches.

	B-GIDM	ML-GIDM	B-IDM	B-GM
$FPR(\%)$	11.26	11.53	11.67	12.01
$TPR(\%)$	93.27	93.02	92.65	91.84

3.5 Summary

In this chapter, we proposed a fully Bayesian analysis procedure based on learning Markov chain Monte Carlo (MCMC) simulation methods for GID mixture model. The estimation was based on computing the posterior of the GID parameters by developing a conjugate prior. The posterior was updated by Gibbs sampling and Metropolis-Hastings methods. Moreover, for model selection, the Bayesian information criterion was used. The obtained accuracies from two challenging problems (object classification and forgery detection) show the effectiveness of the proposed method.

It is noteworthy that several modern applications involve dynamic data. Thus, a more convenient formulation to handle this kind of data could be based on infinite mixture models. In the next chapter, we investigate infinite GID mixture models.

Chapter 4

An Infinite Mixture Model of Generalized Inverted Dirichlet Distributions for High-Dimensional Positive Data Modeling

In this chapter, we propose an infinite mixture model for the clustering of positive data. The proposed model was based on the generalized inverted Dirichlet distribution, which has a more general covariance structure than the inverted Dirichlet and has been widely used in several recent machine learning and data mining applications. The proposed mixture was developed in an elegant way that allows for simultaneous clustering and feature selection. Furthermore, it is learned using a fully Bayesian approach via the Gibbs sampling. The merits of the proposed approach are demonstrated using a challenging application, namely, image categorization.

4.1 Introduction

The important proliferation of digital content requires the development of powerful approaches for knowledge extraction, analysis, and organization. Clustering, in particular, has been widely adopted for knowledge discovery and data engineering. The main goal of any clustering algorithm is to partition a given data set into groups so that objects within a cluster are more similar than those in different clusters [138]. Many clustering techniques have been developed in the past and have been applied successfully on different data types (e.g. binary, discrete, continuous) and extracted within various applications [139–141]. Among these techniques, mixture models have played important roles in many areas including , but not limited to, image processing, computer vision, data mining, and pattern recognition. This is due to their flexibility and strong statistical foundations which offer a formal, principled way to clustering. In particular, the Gaussian mixture model has drawn considerable attention in the machine learning community and has achieved good results [142]. However, recent concentrated research efforts have shown that this mixture model may fail to provide good generalization capabilities when the per-cluster data distributions are clearly non-Gaussian, which is the case for positive data as discussed in [8, 9, 143].

The main contribution of [8, 9] was the introduction of the finite inverted Dirichlet mixture model for the clustering of positive data, which are naturally generated by many real-world applications. They have also proposed a detailed approach for the learning of the parameters of this finite mixture. In order to handle a huge number of classes and avoid over- or under-fitting problems which are central issues in learning-based techniques, the finite inverted Dirichlet mixture was extended to the infinite case in [9]. This extension was based on the consideration of Dirichlet processes which have been widely used in the case of nonparametric Bayesian approaches [144, 145]. Despite its advantages and flexibility, the inverted Dirichlet has a very restrictive covariance structure that is generally violated by data generated from real-life applications. Thus, we propose an alternative to the inverted Dirichlet, namely the generalized inverted Dirichlet (GID) that has a more general covariance structure. Our work can be viewed as a principled and natural extension to the framework developed in [9], since we consider the GID to be within an infinite mixture model by taking feature selection into account. The feature selection process was formalized by introducing a background distribution, common to all mixture components, into the infinite model to represent irrelevant features. Moreover, we developed an algorithm for the learning of the resulting model using Markov chain Monte Carlo (MCMC) sampling techniques, namely the Gibbs sampling and Metropolis-Hastings [146].

This chapter is organized as follows: Section 4.2 presents our infinite mixture model. Section 4.3 provides empirical evaluation based on the challenging problem of image categorization. Finally, in Section 4.4, a summary on this chapter is given.

4.2 The Model

In this section, we start by presenting the finite GID mixture model, then the development of its infinite counterpart is described. A feature selection approach is also proposed.

4.2.1 Finite Model

Let us consider a data set of $\mathscr{Y} = (Y_1, \ldots, Y_N)$ of N D-dimensional positive vectors, where $Y_i = (Y_{il}, \ldots, Y_{iD}), i = 1, \ldots, N$. We assume that Y_i follows a mixture of M GID distributions:

$$p(Y_i|\Theta) = \sum_{j=1}^{M} p_j p(Y_i|\Theta_j) \tag{4.1}$$

where $p(Y_i|\Theta_j)$ is a GID distribution [63]:

$$p(Y_i|\Theta) = \prod_{l=1}^{D} \frac{\Gamma(\alpha_{jl}+\beta_{jl})}{\Gamma(\alpha_{jl})\Gamma(\beta_{jl})} \frac{Y_{il}^{-\alpha_{jl}-1}}{T_{il}^{\eta_{jl}}} \tag{4.2}$$

where $T_{il} = 1 + \sum_{k=1}^{l} Y_{ik}$ and $\eta_{jl} = \alpha_{jl} + \beta_{jl} - \beta_{j(l+1)}$ with $\beta_{j(D+1)} = 0$. Each $\Theta_j = (\alpha_{j1}, \beta_{j1}, \ldots, \alpha_{jD}, \beta_{jD})$ is the set of parameters defining the jth component, and p_j is the mixing weight for that component. The p_j must satisfy the following constraints: $p_j > 0, j = 1, \ldots, M$, and $\sum_{j=1}^{M} p_j = 1$.

In finite mixture clustering [142], each vector Y_i is assigned to all classes with different posterior probabilities $p(j/Y_i) \propto p_j p(Y_i|\Theta_j)$. It is possible to show that the properties of the GID distribution allows the factorization of the posterior probabilities [44] as: $p(j|Y_i) \propto p_j \prod_l^D p_{ib}(X_{il}|\Theta_j)$, where $X_{il} = Y_{il}$ and $X_{il} = \frac{Y_{il}}{1+\sum_{k=1}^{l-1} Y_{il}}$ for $l > 1$, $p_{ib}(X_{il}|\theta_{jl})$ is an inverted Beta distribution with

$\theta_{jl} = (\alpha_{jl}, \beta_{jl}), l = 1, \dots, D:$

$$p_{ib}(X_{il}|\alpha_{jl}, \beta_{jl}) = \frac{\Gamma(\alpha_{jl} + \beta_{jl})}{\Gamma(\alpha_{jl})\Gamma(\beta_{jl})} \frac{X_{il}^{\alpha_{jl}-1}}{(1 + X_{il})^{(\alpha_{jl}+\beta_{jl})}} \tag{4.3}$$

Thus, the clustering structure underlying $\mathscr{Y}$ is the same as that underlying $\mathscr{X} = (X_1, ..., X_N)$ as described by the following mixture model with conditionally independent features:

$$p(X_i|\Theta) = \sum_{j=1}^{M} p_j \prod_{l=1}^{D} p_{ib}(X_{il}|\theta_{jl}) \tag{4.4}$$

This means that the GID mixture model has the ability to reduce complex multidimensional clustering problems to a sequence of one-dimensional ones.

4.2.2 Infinite Model

Let Z_i be a variable that indicates from which cluster each vector X_i came from (i.e $Z_i = j$ means that X_i came from component j). Therefore, $p_j = p(Z_i = j), j = 1, \dots, M$ and:

$$P(Z|P) = \prod_{j=1}^{M} p_j^{n_j} \tag{4.5}$$

where $P = (p_1, \dots, p_M)$, $Z = (Z_1, \dots, Z_N)$, $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_i=j}$ is the number of vectors in cluster j. It is common to consider a Dirichlet distribution as a prior for P, and is justified by the fact that the Dirichlet is a conjugate to the multinomial [147]:

$$p(P|\eta_1, \dots, \eta_M) = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} p_j^{\eta_j - 1} \tag{4.6}$$

where $(\eta_1, \dots, \eta_M) \in \mathbb{R}^{+M}$ are the parameters of the Dirichlet. By taking $\eta_j = \frac{\eta}{M}, j = 1, \dots, M$, where $\eta \in \mathbb{R}^+$, we obtain:

$$p(P|\eta) = \frac{\Gamma(\eta)}{\Gamma(\frac{\eta}{M})^M} \prod_{j=1}^{M} p_j^{\eta - 1} \tag{4.7}$$

Since the Dirichlet is a conjugate prior to the multinomial, we can marginalize P out of the equation:

$$p(Z|\eta) = \int_P p(Z|P)p(P|\eta)dP = \frac{\Gamma(\eta)}{\Gamma(\eta+N)} \prod_{j=1}^{M} \frac{(\frac{\eta}{M}+n_j)}{\Gamma(\frac{\eta}{M})} \tag{4.8}$$

which can be considered as a prior on Z. We also have:

$$p(P|Z,\eta) = \frac{p(Z|P)p(P|\eta)}{p(Z|\eta)} = \frac{\Gamma(\eta+N)}{\prod_{j=1}^{M}\Gamma(\frac{\eta}{M}+n_j)} \prod_{j=1}^{M} p_j^{n_j+\frac{\eta}{M}-1} \tag{4.9}$$

which is a Dirichlet distribution with parameters $(n_1+\frac{\eta}{M},\ldots,n_M+\frac{\eta}{M})$ from which we can show that:

$$p(Z_i = j|\eta, Z_{-i}) = \frac{n_{-i,j}+\frac{\eta}{M}}{N-1+\eta} \tag{4.10}$$

where $Z_{-i} = \{Z_1,\ldots,Z_{i-1},Z_{i+1},\ldots,Z_N\}, n_{-i,j}$ is the number of vectors, excluding X_i, in cluster j. Letting $M \to \infty$, the conditional prior gives the following limits

$$p(Z_i = j = |\eta, Z_{-i}) = \begin{cases} \frac{n_{-i,j}}{N-1+\eta} & \text{if } n_{-i,j} > 0 \quad (\text{cluster } j \in \mathscr{R}) \\ \frac{\eta}{N-1+\eta} & \text{if } n_{-i,j} > 0 \quad (\text{cluster } j \in \mathscr{U}) \end{cases} \tag{4.11}$$

where $\mathscr{R}$ and $\mathscr{U}$ are the sets of represented and unrepresented clusters, respectively. Actually, the previous equation describes a Dirichlet process of mixtures in which learning is generally based on the Gibbs sampling MCMC technique [143]. This is done by generating the vectors assignments according to the posterior distribution:

$$p(Z_i = j|Z_{-i},\mathrm{X}) \propto p(Z_i = j|Z_{-i}) \int p(X_i|Z_i = j,\Theta_j)p(\Theta_j|Z_{-i},\mathrm{X}_{-i})d\Theta_j \tag{4.12}$$

where Z_{-i} represents all the vector assignments except Z_i and X_{-i} represents all the vectors except X_i. In order to obtain the conditional posterior distributions of our infinite model's parameters given the data that we would like to cluster, we need to choose appropriate priors. Here, we considered the same priors previously proposed in [143] for the inverted Dirichlet, which is actually the multivariate case of the inverted beta in Eq. 4.4. Thus, we need to parameterize the inverted Beta as follows:

$$p_{ib}(X_{il}||\alpha_{jl}|,\mu_{jl}) = \frac{\Gamma(|\alpha_{jl}|)}{\Gamma(\frac{\mu_{jl}(|\alpha_{jl}|-1)}{1+\mu_{jl}})\Gamma(\frac{|\alpha_{jl}|+\mu_{jl}}{1+\mu_{jl}})} X_{il}^{\frac{\mu_{jl}(|\alpha_{jl}|-1)}{1+\mu_{jl}}-1}(1+X_{il})^{-|\alpha_{jl}|} \tag{4.13}$$

where $|\alpha_{jl}| = \alpha_{jl} + \beta_{jl}$,$\mu_{jl} = \frac{\alpha_{jl}}{\beta_{jl}-1}$, and for which we impose independent uniform and inverse Gamma priors, respectively:

$$p(\mu_{jl}) \sim \mathrm{U}_{[a,b]}^{jl} \tag{4.14}$$

where $a = \min\{X_{il}\}$ and $b = \max\{X_{il}\}$.

$$p(|\alpha_{jl}||\sigma,\varpi) \sim \frac{\varpi^{\sigma} exp(\frac{-\varpi}{|\alpha_{jl}|})}{\Gamma(\sigma)|\alpha_{jl}|^{\sigma+1}} \tag{4.15}$$

where σ and ϖ are hyperparameters, common to all components, representing the shape and scale of the distribution, respectively. We consider the following priors to add more flexibility to the model:

$$p(\sigma|\lambda,\delta) \sim \frac{\delta^{\lambda} exp(\frac{-\lambda}{\sigma})}{\Gamma(\lambda)\sigma^{\lambda+1}} \quad p(\varpi|\phi) \sim p(\phi)exp(-\phi\varpi) \tag{4.16}$$

Having all our priors in hand, the calculation of the parameter posteriors, given the rest of the variables, becomes straightforward:

$$p(|\alpha_{jl}||\ldots) \propto \frac{\varpi^{\sigma} exp(\frac{-\varpi}{|\alpha_{jl}|})}{\Gamma(\sigma)|\alpha_{jl}|^{\sigma+1}} \prod_{Z_i=j} p(X_i|\Theta) \tag{4.17}$$

$$p(\mu_{jl}|\ldots) \propto \prod_{Z_i=j} p(X_i|\Theta) \tag{4.18}$$

$$p(\sigma|\ldots) \propto \frac{\varpi^{M\sigma}\delta^{\lambda} exp(-\delta/\sigma)}{\Gamma(\sigma)^{M}\Gamma(\lambda)\sigma^{\lambda+1}} \prod_{j=1}^{M} \frac{exp(\frac{-\varpi}{|\alpha_{jl}|})}{|\alpha_{jl}|^{\sigma+1}} \tag{4.19}$$

$$p(\varpi|\dots) \propto \frac{\varpi^{M\sigma}\phi exp(-\phi\varpi)}{\Gamma(\sigma)^M}\prod_{j=1}^{M}\frac{exp(\frac{-\varpi}{|\alpha_{jl}|})}{|\alpha_{jl}|^{\sigma+1}} \tag{4.20}$$

With these posteriors, the learning algorithm can be summarized in algorithm 2. Note that in the initialization step, the algorithm starts by assuming that all the vectors are in the same cluster and that the initial parameters were generated as random samples from their prior distributions.

Algorithm 2 Learning of the infinite GID mixture model with simultaneous feature selection

```
{Initialization}
    Generate $Z_i$ from Eq. 4.12, $i = 1,\dots,N$ using the algorithm in [148].
    Update the number of represented components M.
    Update nj and $p_j = \frac{n_j}{N+\eta}$, $j = 1,\dots,M$
    Update the mixing parameters of unrepresented components $p_U = \frac{\eta}{\eta+N}$
    Generate $|\mu_{jl}|$ from Eq. 4.18 and $|\alpha_{jl}|$ from Eq. 4.17, $j = 1,...,M$ using Metropolis-Hastings
  [146].
    Update the hyperparameters: Generate $\sigma$ from Eq. 4.19 and from Eq. 4.20 using adaptive
  rejection sampling as proposed in [149].
```

4.2.3 Feature Selection

It is noteworthy that the model proposed in the previous section does not take into account the fact that different features may have different weights in the clustering structure. Additionally, some features may be noisy and may then compromise the generalization capabilities of the model [150]. In order to introduce feature selection in our model, it is possible to use the following formulation:

$$p(X_i|\Xi) = \sum_{j=1}^{M} p_j \prod_{l=1}^{D}[\rho_l p_{ib}(X_{il}||\alpha_{jl}|,\mu_{jl}) + (1-\rho_l)p_{ib}(X_{il}||\alpha_{jl}^{irr}|,\mu_{jl}^{irr})] \tag{4.21}$$

where $\Xi = \Theta,\rho,\Theta,,\Theta^{irr}$ is the set of all the model parameters, $\rho = (\rho_1,\dots,\rho_D)$, $\Theta^{irr} = |\alpha_{jl}^{irr}$, and $p_{ib}(X_{il}||\alpha_{jl}^{irr}|,\mu_{jl}^{irr})$ is a background distribution, common to all mixture components, that represent irrelevant features. $\rho_l = p(z_{il} = 1)$ represents the probability that the lth feature is relevant for clustering where z_{il} is a hidden variable equal to 1, if the lth feature of X_i is relevant, and is 0, otherwise. By introducing feature selection, the learning algorithm proposed in the previous section had to be slightly modified by adding simulations from the posteriors of $|\alpha_{jl}^{irr}|, \mu_{jl}^{irr}$, for which we chose the same priors that were considered for $|\alpha_{jl}|$,μ_{jl}, and ρ. We considered a beta prior with location δ_1 and scale δ_2 common to all dimensions:

$$p(\rho|\delta_1, \delta_2) = \left[\frac{\Gamma(\delta_2)}{\Gamma(\delta_1\delta_2)\Gamma(\delta_2(1-\delta_1))}\right]^D \prod_{d=1}^{D} \rho_d^{\delta_1\delta_2-1}(1-\rho_d)^{\delta_2(1-\delta_1)-1} \tag{4.22}$$

Moreover, the z_i were generated from a D-variate Bernoulli distribution with parameters $\hat{z}_{il}, \dots, \hat{z}_{iD}$, where:

$$\hat{z}_{il} = \frac{\rho_l p_{ib}(X_{il}||\alpha_{jl}|, \mu_{jl})}{\rho_l p_{ib}(X_{il}||\alpha_{jl}|, \mu_{jl}) + (1-\rho_l)p_{ib}(X_{il}||\alpha_{jl}^{irr}|, \mu_{jl}^{irr})} \tag{4.23}$$

denotes the expectation for z_{il}:

$$p(z|\rho) = \prod_{i=1}^{N}\prod_{d=1}^{D} \rho_d^{z_{id}}(1-\rho_d)^{1-z_{id}} = \prod_{d=1}^{D} \rho_d^{f_d}(1-\rho_d)(N-f_d) \tag{4.24}$$

where $f_d = \sum_{i}^{N} \mathbb{I}_{Z_{id}=1}$. Then, the posterior for ρ is:

$$p(\rho|\dots) \propto (\rho|\delta_1, \delta_2)p(z|\rho) \propto \prod_{d=1}^{D} \rho_d^{\delta_1\delta_2+f_d-1}(1-\rho_d)^{\delta_2(1-\delta_1)+N-f_d-1} \tag{4.25}$$

Note that the feature selection process starts by assuming that all features have a probability of 0.5 in order to be considered relevant. Then, this relevancy value was updated during the learning iterations.

4.3 Experimental Result

In this section, we demonstrate the utility of our model by applying it on a challenging application, namely, visual scene categorization. Moreover, we compare the proposed approach with the infinite inverted Dirichlet proposed in [9]. Comparing our results with many other generative and discriminative techniques is clearly out of the scope of this work. In this application, the values of the hyperparameters have been set experimentally to one. This choice has been found reasonable according to our simulations.

The wealth of images generated everyday has spurred a tremendous interest in developing approaches to understand the visual content of these images. In this section, we shall focus on the challenging problem of images categorization, in order to validate our GID infinite mixture model. This is a crucial step in several applications such as annotation [151, 152], retrieval [153, 154], and object recognition [155]. A common recent approach widely used for image categorization, that we follow in this application, is the consideration of the so-called bag of visual words generated via quantization of local image descriptors, such as SIFT [136].

We considered two challenging datasets in our experiments, namely, the 15 class scene recognition data set [156] and the 8 class sport events data set [157]. The 15 class scene recognition data set contains the following categories: coasts (360 images), forests (328 images), mountains (374 images), open country (410 images), highways (260 images), inside of cities (308 images), tall buildings (356 images), streets (292 images), suburb residences (241 images), bedrooms (174 images), kitchens (151 images), livingrooms (289 images), offices (216 images), stores (315 images), and industrial (311 images). Figure 4.1 displays examples of images from this data set. The 8 class sport events dataset contains the following categories: rowing (250 images), badminton (200 images), polo (182 images), bocce (137 images), snowboarding (190 images), croquet (236 images), sailing (190 images), and rock climbing (194 images). Figure 2 displays examples of images from this data set. We constructed our visual vocabulary for each data set from half of the available images in each data set. We detected interest points from these images using the difference-of-Gaussians point detector, since it has shown to have excellent performance [136].

Then, we used the SIFT descriptor [136], computed on detected keypoints of all images, generating a 128-dimensional vector for each keypoint. Moreover, extracted vectors were clustered using the K-Means algorithm using 250 visual-words. Each image in the data sets was then represented by a 250-dimensional positive vector describing the frequencies of visual words, obtained from the constructed visual vocabulary. These vectors were separated into a test set of vectors and

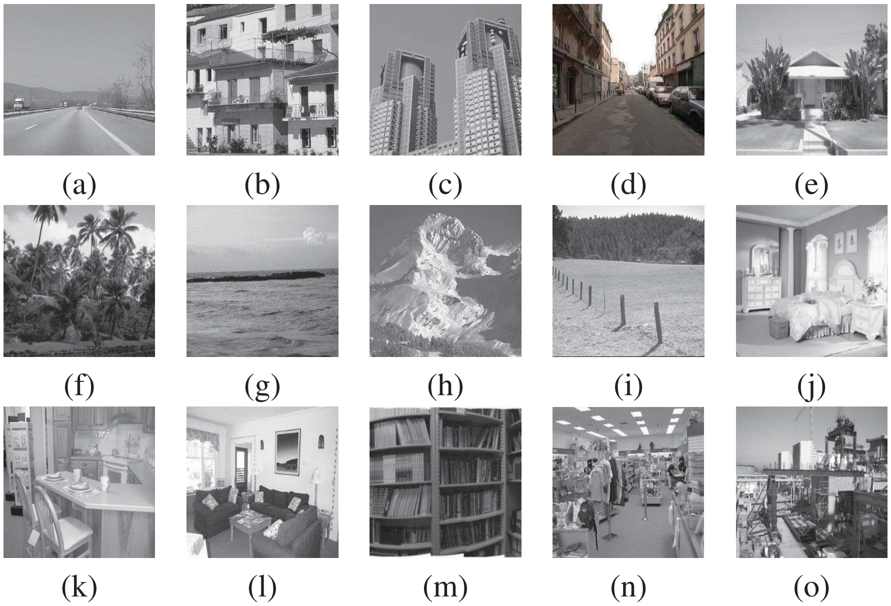


Figure 4.1: Sample images from each group in the 15 class scene recognition data set. (a) Highway, (b) Inside of cities, (c) Tall buildings, (d) Streets, (e) Suburb residence, (f) Forest, (g) Coast, (h) Mountain, (i) Open country, (j) Bedroom, (k) Kitchen, (l) Livingroom, (m) Office, (n) Store, (o) Industrial.

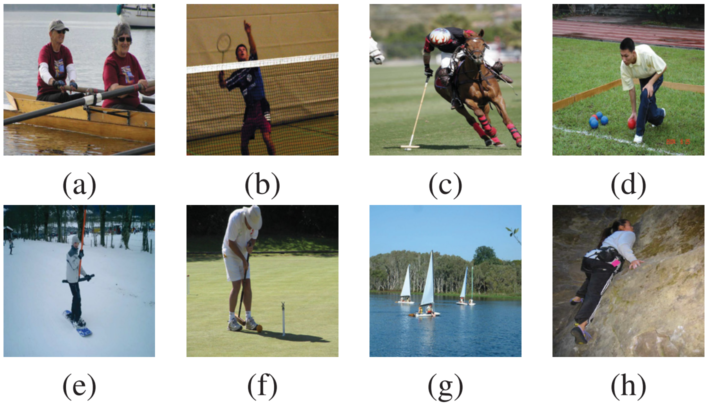


Figure 4.2: Sample images from each group in the 8 class sports event dataset. (a) rowing,(b) badminton, (c) polo, (d) bocce, (e) snowboarding, (f) croquet, (g) sailing, (h) rock climbing.

a training set of vectors. Then, we applied our learning algorithm to the training vectors for each class. After this stage, each class in the database was represented by a statistical model. Finally, in the classification stage, each unknown image was assigned to the class, thereby increasing its log-likelihood. A summary of the classification results, measured by the average values of the diagonal entries in the confusion matrices, that were obtained for the different classification tasks, is shown in Table 4.1. This table clearly shows that the GID infinite mixture, without feature selection (IGID) and with feature selection (IGIDFS), outperformed the infinite inverted Dirichlet (IID) mixture. The results can be explained by the fact that the GID is more flexible than the inverted Dirichlet. We can clearly notice that introducing feature selection further improved the results.

Table 4.1: Classification performance % obtained for the two tested data sets using three different approaches

	IGIDFS	IGID	IID
Dataset 1 (15 catgories)	75.31%	74.5%	70.11%
Dataset 2 (8 categories)	74.03%	73.25%	70.72%

4.4 Summary

In this chapter, we developed an infinite generalized inverted Dirichlet distribution. The infinite mixture models have many advantages over its finite counterpart. One advantage is that the number of components can be determined automatically, thereby avoiding the drawback of EM based approaches since the inference is based on MCMC. Using the proposed approach allowed the estimation of the model parameters, model selection, and feature selection accordingly. To confirm the merits and effectiveness of the proposed approach, experiments were conducted on a real challenging problem, namely, visual scenes. In the next chapter, we present a fully Bayesian GID mixture which is based on the Reversible Jump Markov Chain (RJMCMC).

Chapter 5

A Fully Bayesian Framework for Positive Data Clustering

The main concern with mixture modeling is to describe the data such that each observation belongs to one of a number of different groups. Mixtures of distributions provide a flexible and convenient class of models for density estimation and their statistical learning has been studied extensively. In this context, fully Bayesian approaches have been widely adopted for mixture estimation and model selection problems and have shown some effectiveness due to the incorporation of the prior knowledge about the parameters. In this chapter, we propose a fully Bayesian approach for finite generalized inverted Dirichlet (GID) mixture model learning using a reversible jump Markov chain Monte Carlo (RJMCMC) approach [21]. RJMCMC enables us to deal simultaneously with the model selection and the parameters estimation in one single algorithm. The merits of RJMCMC for GID mixture learning is investigated using synthetic data and real data extracted from an interesting application, namely, object detection.

5.1 Introduction

A finite mixture model provides a natural representation of heterogeneity when data are assumed to be generated from two or more distributions mixed in varying proportions. Finite mixture models provide a powerful, flexible, and well principled statistical approaches and have been commonly used to model complex data in many applications [142, 158–160]. Model selection and estimation of parameters are the fundamental problems in mixture modeling. To date, Gaussian mixture modeling has been the subject of much research because of its relative simplicity [142]. The Gaussian assumption is, however, not realistic in the majority of signal and image processing problems [161–163]. In general, numerous approaches have been developed for the learning of mixture parameters (i.e., for both model selection and parameter estimation). These approaches can be categorized into deterministic and Bayesian methods. Deterministic inference is an important branch of inference methodologies. It has been actively studied, especially during the last fifteen years. Using deterministic methods, data are taken as random while parameters are taken as fixed and unknown. The inference is generally based on the likelihood of the data. Among these approaches, the expectation maximization (EM) [12] algorithm has been extensively used in the case of maximum likelihood estimation. However, it has been shown that the maximum likelihood estimation suffers from singularities, convergence to local maxima [13], leads to more complex models, and then to overfitting [164]. Moreover, likelihood is a non-decreasing function based on the number of components, thus the maximum likelihood approach cannot be used as a model selection criterion. To overcome this problem, many model selection approaches based on the Bayesian approximation have been proposed such as, the Bayesian information criterion, minimum message length, and maximum entropy criterion.

Pure Bayesian techniques can be used as an alternative to learn mixture models and they generally provide good results. Using Bayesian approaches for finite mixture modeling is performed by introducing suitable priors distributions for the model parameters. Moreover, Bayesian approaches provide us with a valid inference without relying on the asymptotic normality assumption since simulation from the posterior distribution of the unknown parameters is feasible [165].

This simulation is generally based on MCMC, which is an important tool for a statistical Bayesian inference [164, 166]. In this chapter, we consider a special MCMC technique, which simultaneously performs parameter estimation and model selection for generalized inverted Dirichlet (GID) mixture, namely, the reversible jump MCMC (RJMCMC) sampling method, previously proposed in [167]. RJMCMC has been applied successfully in the past in the case of the Gaussian [21, 168] and beta [169] mixtures. It provides a general framework for Markov chain Monte Carlo (MCMC) simulation in which the dimension of the parameter space can vary between iterations of the Markov chain.

This chapter is organized as follows. In Section 5.2, we introduce the GID mixture model. Section 5.3 defines our fully Bayesian framework for learning the GID mixture using the RJMCMC technique. In Section 5.4, split/merge and birth/death moves are explained in detail. Section 5.5 presents the experimental results for generated and real data to demonstrate the merits of the proposed approach. Lastly, we provide a summary of this chapter in Section 5.6.

5.2 GID Mixture Model

Let us consider a data set $\mathscr{Y}$ composed of N D-dimensional positive vectors, $\mathscr{Y} = (\vec{Y}_1, \vec{Y}_2, \dots, \vec{Y}_N)$. We assume that $\mathscr{Y}$ is governed by a weighted sum of M generalized inverted Dirichlet (GID) component densities with parameters $\Theta_M = (\vec{\theta}_1, \vec{\theta}_2, \dots, \vec{\theta}_M, p_1, p_2, \dots, p_M)$ where $\vec{\theta}_j$ is the parameter vector of the jth component, and $\{p_j\}$ are the mixing weights which are positive and sum to one:

$$p(\vec{Y}_i|\Theta_M) = \sum_{j=1}^{M} p_j p(\vec{Y}_i|\vec{\theta}_j) \tag{5.1}$$

where $p(\vec{Y}_i|\vec{\theta}_j)$ is the GID distribution with parameters $\vec{\theta}_j = (\alpha_{j1}, \beta_{j1}, \alpha_{j2}, \beta_{j2}, \ldots, \alpha_{jD}, \beta_{jD})$. In mixture-based clustering, each data point $\vec{Y}_i$ is assigned to all classes with different posterior probabilities $p(j|\vec{Y}_i) \propto p_j p(\vec{Y}_i|\vec{\theta}_j)$. The GID distribution allows the factorization of the posterior probability as shown in [44]

$$p(j|\vec{Y}_i) \propto p_j \prod_{l=1}^{D} p_{ib}(X_{il}|\theta_{jl}) \tag{5.2}$$

where we have set $X_{i1} = Y_{i1}$ and $X_{il} = \frac{Y_{il}}{1+\sum_{k=1}^{l-1} Y_{ik}}$ for $l > 1$. $p_{ib}(X_{il}|\theta_{jl})$ is an inverted beta distribution with parameters $\theta_{jl} = (\alpha_{jl}, \beta_{jl})$, $\beta_{jl} > 2$, $l = 1, \ldots, D$. Thus, the clustering structure underlying $\mathscr{Y}$ is the same as the one underlying $\mathscr{X} = (\vec{X}_1, \ldots, \vec{X}_N)$, where $\vec{X}_i = (X_{i1}, \ldots, X_{iD})$, $i = 1, \ldots, N$, and is governed by the following mixture model with conditionally independent features:

$$p(\vec{X}_i|\Theta_M) = \sum_{j=1}^{M} p_j \prod_{l=1}^{D} p_{ib}(X_{il}|\theta_{jl}) \tag{5.3}$$

where

$$p_{ib}(X_{il}|\theta_{jl}) = \frac{\Gamma(\alpha_{jl}+\beta_{jl})}{\Gamma(\alpha_{jl})\Gamma(\beta_{jl})} X_{il}^{\alpha_{jl}-1}(1+X_{il})^{-\alpha_{jl}-\beta_{jl}} \tag{5.4}$$

The mean and the variance of the inverted Beta distribution are as follows:

$$\mu_{jl} = \frac{\alpha_{jl}}{\beta_{jl}-1} \tag{5.5}$$

$$\sigma_{jl}^2 = \frac{\alpha_{jl}(\alpha_{jl}+\beta_{jl}-1)}{(\beta_{jl}-2)(\beta_{jl}-1)^2} \tag{5.6}$$

Using Eqs. 5.5 and 5.6, the parameters α_{jl} and β_{jl} of the inverted beta distribution can be written with respect to the mean and the variance as follows:

$$\alpha_{jl} = \frac{\mu_{jl}{}^2(1+\mu_{jl}) + \mu_{jl}\sigma_{jl}^2}{\sigma_{jl}^2} \tag{5.7}$$

$$\beta_{jl} = \frac{\mu_{jl}(1+\mu_{jl}) + 2\sigma_{jl}^2}{\sigma_{jl}^2} \tag{5.8}$$

then, the probability density function of the inverted beta, as a function of its mean and variance, can be written as follows:

$$\begin{aligned} p_{ib}(X_{il}|\mu_{jl},\sigma_{jl}^2) &= \frac{1}{\mathscr{B}\left(\frac{{\mu_{jl}}^2(1+\mu_{jl})+\mu_{jl}\sigma_{jl}^2}{\sigma_{jl}^2}, \frac{\mu_{jl}(1+\mu_{jl})+2\sigma_{jl}^2}{\sigma_{jl}^2}\right)} \\ &\times \quad X_{il}^{\left(\frac{{\mu_{jl}}^2(1+\mu_{jl})+\mu_{jl}\sigma_{jl}^2}{\sigma_{jl}^2}-1\right)} \\ &\times \quad \left(1+X_{il}\right)^{-\left(\frac{{\mu_{jl}}^2(1+\mu_{jl})+\mu_{jl}\sigma_{jl}^2+\mu_{jl}(1+\mu_{jl})+2\sigma_{jl}^2}{\sigma_{jl}^2}\right)} \end{aligned} \tag{5.9}$$

where $\mathscr{B}$ is the beta function which is defined as $\mathscr{B}(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$.

The main problem, when dealing with mixture models, is to estimate the parameters. A huge number of methods in the literature have been developed in the past [142]. Among these methods, the maximum likelihood estimation, which maximizes the likelihood through the expectation maximization (EM) algorithm [12, 13], has received a lot of attention. However, the EM algorithm suffers from some drawbacks. First, it is highly dependent on the initialization values, which is the main reason why it may converge to a local maxima. Second, it suffers from the overfitting problem. In EM-based formulation, a latent allocation vector is introduced $\vec{Z}_i = (Z_{i1},\dots,Z_{iM})$ and indicates to which mixture component each vector $\vec{X}_i$ belongs to, such that $Z_{ij} \in \{0,1\}$, $\sum_{j=1}^{M} Z_{ij} = 1$ and $Z_{ij} = 1$ if $\vec{X}_i$ belongs to component j, and to 0, otherwise. $\mathscr{Z} = \{\vec{Z}_1,\dots,\vec{Z}_N\}$ is known as the set of "membership vectors" of the mixture model and its unique elements, Z_i, are supposed to be drawn independently from the following distribution:

$$p(Z_i = j) = p_j \quad j = 1,\dots,M. \tag{5.10}$$

Thus, the distribution of $\vec{X}_i$, given the class label $\vec{Z}_i$, is:

$$p(\vec{X}_i|\Theta_M,\vec{Z}_i) = \prod_{j=1}^{M}\left(\prod_{l=1}^{D} p_{ib}(X_{il}|\theta_{jl})\right)^{Z_{ij}} \tag{5.11}$$

5.3 GID Bayesian Learning Using RJMCMC

One of the main concerns in mixture modeling is model selection (i.e. determining the number of components). Within Bayesian modeling, many approaches have been proposed to infer the optimal number of components. Examples of Bayesian model selection approaches include Bayes factors, Bayesian information criterion (BIC), deviance information criterion (DIC), RJMCMC, and birth and death processes [170–172]. In this work, we develop a RJMCMC-based method. It allows us to successfully perform both model selection and parameter estimation in one single algorithm. In our proposed Bayesian framework, the number of component M, the parameters which govern the mixture $\vec{\theta}$ components, and the mixing weight $\vec{P} = (p_1, \dots, p_M)$, are considered as drawn from an appropriate distribution. The joint distribution of all variables can be written as:

$$p(M,\vec{P},Z,\vec{\theta},\mathscr{X}) = p(M)p(\vec{P}|M)p(Z|\vec{P},M)p(\vec{\theta}|Z,\vec{P},M)p(\mathscr{X}|\vec{\theta},Z,\vec{P},M) \tag{5.12}$$

where the conditional independencies $p(\vec{\theta}|Z,\vec{P},M) = p(\vec{\theta}|M)$ and $p(\mathscr{X}|\vec{\theta},Z,\vec{P},M) = p(\mathscr{X}|\vec{\theta},Z)$ are imposed. The joint distribution can be written as follows:

$$p(M,\vec{P},Z,\vec{\theta},\mathscr{X}) = p(M)p(\vec{P}|M)p(Z|\vec{P},M)p(\vec{\theta}|M)p(\mathscr{X}|\vec{\theta},Z,\vec{P},M) \tag{5.13}$$

Now, the main goal of the Bayesian inference is to generate realizations from the conditional joint density $p(M,\vec{P},Z,\vec{\theta}|\mathscr{X})$.

5.3.1 Priors and Posteriors

In this section, we will define the priors of the different parameters in our hierarchical Bayesian model. These parameters are supposed to be drawn independently. For our model, we have chosen an inverted beta and inverse gamma distributions as priors for the mean μ_{jl} and the variance σ_{jl}^2, respectively.

$$\begin{aligned} p(\vec{\mu}_j|\varepsilon,\zeta) &= \prod_{l=1}^{D} \frac{1}{\mathscr{B}\left(\frac{\varepsilon^2(1+\varepsilon)+\varepsilon\zeta}{\zeta}, \frac{\varepsilon(1+\varepsilon)+2\zeta}{\zeta}\right)} \mu_{jl}^{\left(\frac{\varepsilon^2(1+\varepsilon)+\varepsilon\zeta}{\zeta}-1\right)} \\ &\times \left(1+\mu_{jl}\right)^{-\left(\frac{\varepsilon^2(1+\varepsilon)+\varepsilon\zeta+\varepsilon(1+\varepsilon)+2\zeta}{\zeta}\right)} \end{aligned} \tag{5.14}$$

where ε_{jl} is the location and ζ_{jl} is the shape parameter for the inverted beta distribution. A common choice for a prior, for the variance $\vec{\sigma}_j^2 = (\sigma_{j1}^2, \ldots, \sigma_{jD}^2)$, is the inverse gamma distribution. Then:

$$p(\vec{\sigma}_j^2|\vartheta,\varpi) \sim \prod_{l=1}^{D} \frac{\vartheta^{\varpi}}{\Gamma(\varpi)} \sigma^{2}{}_{jl}^{-\varpi-1} \exp\left(\frac{\vartheta}{\sigma^2}\right) \tag{5.15}$$

where ϑ and ϖ represent the shape and scale parameters of the inverse gamma distribution, respectively. Using Eqs. 5.14 and 5.15, we have:

$$p(\vec{\Theta}|M,\tau) = \prod_{j=1}^{M} p(\vec{m}_j|\varepsilon,\zeta)p(\vec{v}_j|\vartheta,\varpi) \tag{5.16}$$

where $\tau = (\varepsilon,\zeta,\vartheta,\varpi)$ are the hyperparameters of $\vec{\Theta}$. Therefore, the full conditional posterior distribution for mean $\vec{m}_j$ and the variance $\vec{\sigma_{jl}^2}$ can be written as follows:

$$\begin{aligned} p(\vec{\mu}_{jl}|\ldots) &\propto \prod_{j=1}^{M} p(\vec{\mu}_{jl}|\varepsilon_{jl},\zeta_{jl})p(\vec{\sigma}_{jl}^2|\vartheta_{jl},\varpi_{jl}) \prod_{i=1}^{N} p(\vec{X}_i|\vec{\Theta}_{Z_i}) \\ &\propto p(\vec{\mu}_{jl}|\varepsilon_{jl},\zeta_{jl}) \prod_{i=1}^{N} p(\vec{X}_i|\vec{\Theta}_{Z_i}) \end{aligned} \tag{5.17}$$

$$\begin{aligned} p(\vec{\sigma}_{jl}^2|\ldots) &\propto \prod_{j=1}^{M} p(\vec{\mu}_{jl}|\varepsilon_{jl},\zeta_{jl})p(\vec{\sigma}_{jl}^2|\vartheta_{jl},\varpi_{jl}) \prod_{i=1}^{N} p(\vec{X}_i|\vec{\theta}_{Z_i}) \\ &\propto p(\vec{\sigma}_{jl}^2|\vartheta_{jl},\varpi_{jl}) \prod_{i=1}^{N} p(\vec{X}_i|\vec{\Theta}_{Z_i}) \end{aligned} \tag{5.18}$$

The $|\ldots$ is used to denote conditioning on all other variables. In addition, the typical prior choice for the mixing weight $\vec{P}$ is the Dirichlet distribution since it is defined under the constraint $p_1,\ldots,p_M : \sum_{j=1}^{M} p_j < 1$. Then, the prior can be written as follows:

$$p(\vec{P}|M,\delta) = \frac{\Gamma(\sum_{j=1}^{M}\delta_j)}{\prod j = 1^M \Gamma\delta_j} \prod_{j=1}^{M} p_j^{\delta_j - 1} \tag{5.19}$$

Also, the prior of the membership variable Z is:

$$p(Z|P,M) = \prod_{j=1}^{M} p_j^{n_j} \tag{5.20}$$

where n_j represents the number of vectors belonging to the j^{th} cluster. Using Eqs. 5.19 and 5.20 we get:

$$\begin{aligned} p(\vec{P}|\ldots) &\propto p(Z|\vec{P},M)p(\vec{P}|M,\delta) \\ &\propto \prod_{j=1}^{M} p_j^{n_j} \frac{\Gamma(\sum_{j=1}^{M}\delta_j)}{\prod_{j=1}^{M}\Gamma(\delta_j)} \prod_{j=1}^{M} p_j^{\delta_j - 1} \propto p_j^{n_j+\delta_j-1} \end{aligned} \tag{5.21}$$

which is simply proportional to a Dirichlet distribution with parameters $(\delta_1 + n_1, \ldots, \delta_M + n_M)$. Besides, using Eq. 5.10 the membership variable posterior can be obtained as follows:

$$p(Z_i = j|\ldots) \propto p_j \prod_{l=1}^{D} p_{ib}(X_{il}|\Theta_{jl}) \tag{5.22}$$

Another hierarchical level can be introduced to represent the priors of the hyperparameters in the model. First, the hyperparameters, ε and ζ, which are associated with $\vec{\mu}_j$, are given uniform and inverse gamma priors, respectively:

$$p(\varepsilon) \sim \mathscr{U}_{[a,b]} \tag{5.23}$$

$$p(\zeta|\varphi,\rho) \sim \frac{\rho^{\varphi}\exp(-\rho/\zeta)}{\Gamma(\varphi)\zeta^{\varphi+1}} \tag{5.24}$$

where $a = \min\{X_{il}, i = 1, \ldots, N; l = 1, \ldots, D\}$ and $b = \max\{X_{il}, i = 1, \ldots, N; l = 1, \ldots, D\}$. According to the two previous equations, the conditional posterior for ε and ζ can be written as:

$$p(\varepsilon|\ldots) = p(\varepsilon)\prod_{j=1}^{M} p(\vec{\mu}_j|\varepsilon,\zeta) \tag{5.25}$$

$$p(\zeta|\ldots) = p(\zeta|\varphi,\rho)\prod_{j=1}^{M} p(\vec{\mu}_j|\varepsilon,\zeta) \tag{5.26}$$

Also, the hyperparameters for ϑ and ϖ, which are associated with the variance $\vec{\sigma}_j^2$, are given inverse gamma and exponential priors, respectively:

$$p(\vartheta|\lambda,\nu) \sim \frac{\nu^{\lambda}\exp(-\nu/\vartheta)}{\Gamma(\lambda)\vartheta^{\lambda+1}} \tag{5.27}$$

$$p(\varpi|\phi) \sim \phi\exp(-\phi\varpi) \tag{5.28}$$

From these two pervious equations, the conditional posteriors for ϑ and ϖ are written as:

$$p(\vartheta|\ldots) \propto p(\vartheta|\lambda,\nu)\prod_{j=1}^{M} p(\vec{\sigma}_j^2|\vartheta,\varpi) \tag{5.29}$$

$$p(\varpi|\ldots) \propto p(\varpi|\phi)\prod_{j=1}^{M} p(\vec{\sigma}_j^2|\vartheta,\varpi) \tag{5.30}$$

Finally, for the number of components, M, the common choice is a uniform distribution between 1 and a predefined integer M_{max}.

5.4 RJMCMC Moves

Practically, RJMCMC allows moves between parameter subspaces by allowing the following six types of moves:

1. Update the mixing parameters $\vec{P}$

2. Update the parameters $\vec{\mu}_{jl}$ and $\vec{\sigma}^2_{jl}$

3. Update the membership variable Z

4. Update the hyperparmeters $\varepsilon, \zeta, \vartheta$,and ϖ

5. Split one component into two, or merge two into one

6. The birth or death of an empty component

Each step is called a move, $t = 1, \ldots, 6$, and a sweep is defined as a complete pass over the six moves. Since the first four moves do not change the number of clusters, they can be considered as classic Gibbs sampling moves. On the other hand, moves 5 and 6 involve changing the number of components, M, by 1.

Assume that we are in state Δ_M, where $\Delta_M = (Z, P, M)$. Then the MCMC step representing move (5), takes the form of a Metropolis-Hastings step. This is accomplished by proposing a move from a state Δ_M to a state $\hat{\Delta}_M$, with a target probability distribution (posterior distribution) of $p(\Delta_M|\chi)$ and a proposal distribution $q_t(\Delta_M, \hat{\Delta}_M)$ for the move t. When we are in the current state Δ_M, a given move t to a destination state $\hat{\Delta}_M$ is accepted with probability:

$$p_t(\Delta_M, \hat{\Delta}_M) = \left(1, \frac{p(\hat{\Delta}_M|\chi) q_t(\hat{\Delta}_M, \Delta_M)}{p(\Delta_M|\chi) q_t(\Delta_M, \hat{\Delta}_M)}\right) \tag{5.31}$$

In the case of a move type where the dimension of the parameter does not change, we use an ordinary ratio of densities. A move from a point Δ_M to $\hat{\Delta}_M$, in a higher dimensional space, is done by drawing a vector of continuous random variables u, independent of Δ_M. The new state $\hat{\Delta}_M$ is determined by using an invertible deterministic function of Δ_M and u, f(Δ_M, u) [21]. On the other hand, the move from $\hat{\Delta}_M$ to Δ_M can be carried out by using the inverse transformation. Hence, the move acceptance probability is given by:

$$p_t(\Delta_M, \hat{\Delta}_M) = \min\left(1, \frac{p(\hat{\Delta}_M|\chi) r_m(\hat{\Delta}_M)}{p(\Delta_M|\chi) r_m(\Delta_M) q(u)} \left| \frac{\partial(\hat{\Delta}_M)}{\partial(\Delta_M, u)} \right.\right) \tag{5.32}$$

where $r_m(\Delta_M)$ is the probability of choosing move type m when we are in state Δ_M, and $q(u)$ is the density function of u. The last term, $\frac{\partial(\hat{\Delta}_M)}{\partial(\Delta_M,u)}$, is the Jacobian function arising from the variable change from state (Δ_M, u) to state $\hat{\Delta}_M$. All RJMCMC moves are discussed in the following subsections.

5.4.1 Gibbs Sampling Move

The first four steps of RJMCMC are based on simple Gibbs sampling where the parameters are drawn from their known full conditional distributions. The first move is to draw the mixing weight from a Dirichlet distribution as shown in Eq. 5.21. The second move is based on drawing the mixture parameters using Eqs. 5.14 and 5.15. According to these equations, it is clear that the full conditional distributions are complex and are not in well-known forms. So, the use of the Gibbs sampling is not an appropriate choice in this case. The Metropolis-Hastings (M-H) algorithm [173, 174] could be used. At sweep t, the mean μ_{jl} can be generated using the M-H algorithm as follows:

1. Generate $\hat{\mu}_j \sim q\left(\mu_j|\mu_j^{(t-1)}\right)$ and $u \sim \mathscr{U}_{[0,1]}$
2. Calculate $r = \frac{p(\hat{\mu}_j|\ldots)q\left(\mu_j^{(t-1)}|\hat{\mu}_j\right)}{p(\mu_j^{(t-1)}|\ldots)q\left(\hat{\mu}_j|\mu_j^{(t-1)}\right)}$
3. If $r < u$ then μ_j^t else $\mu_j^t = \mu_j^{(t-1)}$

The most important issue regarding the M-H algorithm, was choosing the candidate generating density q (proposal distribution), in order to keep the mean within the range of $\mu \in [a,b]$. A popular choice for q is a random walk where the previously simulated parameter (μ_{jl}) value was used to generate the value $\hat{\mu}_{jl}$. We propose to generate the new mean, $\mu_j^{(t)}$, from the inverted beta $\mathscr{I}\mathscr{B}$ distribution (w.r.t the mean and variance), where its mean is the previously computed mean value, $\mu_j^{(t-1)}$, and its variance is a constant value C (we take $C = 2.5$). The new generated value of the mean, using the proposal distribution, is:

$$\hat{\mu}_j \sim \mathscr{I}\mathscr{B}\left(\mu_j^{(t-1)}, C\right) \tag{5.33}$$

For the variance, ${\sigma^2}^{(t)}$, we have:

1. Generate $\hat{\sigma^2}_j \sim q\left({\sigma^2}_j|{\sigma^2}_j^{(t-1)}\right)$ and $u \sim \mathscr{U}_{[0,1]}$
2. Calculate $r = \frac{p(\hat{\sigma^2}_j|\ldots)q\left({\sigma^2}_j^{(t-1)}|\hat{\sigma^2}_j\right)}{p({\sigma^2}_j^{(t-1)}|\ldots)q\left(\hat{\sigma^2}_j|{\sigma^2}_j^{(t-1)}\right)}$
3. If $r < u$ then ${\sigma^2}^t_j$ else ${\sigma^2}^t_j = {\sigma^2}_j^{(t-1)}$

where the proposal distribution, q, is given by:

$$\hat{\sigma^2}_j \sim \mathscr{LN}\left(\sigma_j^{2^{(t-1)}}, e^2\right) \tag{5.34}$$

where $\mathscr{LN}$ refers to the lognormal distribution with mean $\log\left({\sigma^2}_j^{(t-1)}\right)$ and variance e^2. The third move is to generate the missing data $Z_i (1 <= i <= N)$ from simulated standard uniform random variables, u_i. $Z_i = j$ if $p(Z_i = 1|\ldots) + \cdots + p(Z_i = j-1|\ldots) < u_i \leq p(Z_i = 1|\ldots) + \ldots p(Z_i = j|\ldots)$. Finally, the Gibbs sampling was used to update the hyperparameters $\varepsilon, \zeta, \vartheta$, and ϖ given by Eqs. 5.25, 5.26, 5.29, and 5.30, respectively.

5.4.2 Split and Combine Moves

In move (5), we made a random choice between attempting to split or combine, with probabilities a_M and b_M, where $b_M = 1 - a_M$, respectively. It is clear that $a_{M_{max}} = 0$ and $b_1 = 0$, otherwise we choose $a_M = b_M = 0.5$ for $M = 1, \ldots, M_{max}$, where M_{max} is the maximum value allowed for M. The combined move was constructed by randomly choosing a pair of components (j_1, j_2), which must be adjacent. In other words, they must meet the following constraint: $\mu_{j1} < \mu_{j2}$, when there is no other μ_j in the interval $[\mu_{j1}, \mu_{j2}]$. Then, these two components can be merged and M is reduced by 1. We denote the newly formed component as j^*, which contains all the observations that were allocated to j_1 and j_2. Finally, we generated the parameter values for the new components $p_{j*}, \mu_{j*}, \sigma^2_{j*}$ by preserving the zeroth, first, and second moments, which are calculated as follows:

$$p_{j*} = p_{j1} + p_{j2} \tag{5.35}$$

$$p_{j*}\mu_{j*} = p_{j1}\mu_{j1} + p_{j2}\mu_{j2} \tag{5.36}$$

$$p_{j*}(\mu_{j*} + \sigma_{j*}^2) = p_{j1}(\mu_{j1} + \sigma_{j1}^2) + p_{j2}(\mu_{j2} + \sigma_{j2}^2) \tag{5.37}$$

For the split type move, a component $j*$ was chosen randomly and split into the two components $j1$ and $j2$ with new parameters $p_{j1}, \mu_{j1}, \sigma_{j1}^2$ and parameters $p_{j2}, \mu_{j2}, \sigma_{j2}^2$, respectively. This confirms Eqs. 5.35, 5.36, and 5.37. Since there are 3 degrees of freedom in achieving this, we need to generate, from a beta distribution, a three-dimensional random vector $u = [u_1, u_2, u_3]$ in order to define the new parameters [21]. We set:

$$p_{j1} = w_{j*}u_1 \quad p_{j1} = w_{j*}(1 - u_1) \tag{5.38}$$

$$\begin{aligned} \mu_{j1} &= \mu_{j*} - u2\sqrt{\sigma_{j*}^2 \frac{p_{j2}}{p_{j1}}} \\ \mu_{j2} &= \mu_{j*} + u2\sqrt{\sigma_{j*}^2 \frac{p_{j1}}{p_{j2}}} \end{aligned} \tag{5.39}$$

$$\begin{aligned} \sigma_{j1}^2 &= u_3(1 - u_2^2)\sigma_{j*}^2 \frac{p_{j*}}{p_{j1}} \\ \sigma_{j2}^2 &= (1 - u_3)(1 - u_2^2)\sigma_{j*}^2 \frac{p_{j*}}{p_{j2}} \end{aligned} \tag{5.40}$$

For the newly generated components, the adjacency condition defined in the combine move must be checked in order to make sure whether the split/combine is reversible or not. If this condition is rejected, then the split/combine move is not reversible and is rejected. Otherwise, the split move is accepted and we reallocate the $j*$ into the new components j_1 and j_2 using Eq. (5.10). According to Eq. 5.32, the acceptance probability R for the split and combine move types can be calculated using the following equation:

$$R = \frac{P(Z, P, M+1, \varepsilon, \zeta, \varpi, \vartheta|\mathrm{X}) b_{M+1}}{p(Z, P, M, \varepsilon, \zeta, \varpi, \vartheta|\mathrm{X}) a_M P_{alloc} q(u)} \left| \frac{\partial \hat{\Delta}_M}{\partial(\Delta_M, u)} \right| \tag{5.41}$$

where the acceptance probability for the split is $min(1,R)$, and for the combine move, is $min(1,R^{-1})$. P_{alloc} is the probability of making this particular allocation to components j_1 and j_2:

$$\begin{aligned} P_{alloc} = & \prod_{Z_i=j_1} \frac{p_{j1}p(x_i|\mu_{j1},\sigma^2_{j1})}{p_{j1}p(x_i|\mu_{j1},\sigma^2_{j1})+p_{j2}p(x_i|\mu_{j2},\sigma^2_{j2})} \\ & \times \prod_{Z_i=j_2} \frac{p_{j2}p(x_i|\mu_{j2},\sigma^2_{j2})}{p_{j1}p(x_i|\mu_{j1},\sigma^2_{j1})+p_{j2}p(x_i|\mu_{j2},\sigma^2_{j2})} \end{aligned} \tag{5.42}$$

Also, $\left|\frac{\partial\hat{\Delta}_M}{\partial(\Delta_M,u)}\right|$ is the Jacobian of the transformation from the state $(w_{j*},\mu_{j*},\sigma^2_{j*}$ $,u_1,u_2,u_3)$ to state $(w_{j1},\mu_{j1},\sigma^2_{j1},w_{j2},\mu_{j2},\sigma^2_{j2})$:

$$\left|\frac{\partial\hat{\Delta}_M}{\partial(\Delta_M,u)}\right| = \frac{|\mu_{j1}-\mu_{j2}|p_{j*}\sigma^2_{j1}\sigma^2_{j2}}{u_2(1-u_2^2)u_3(1-u_3)\sigma^2_{j*}} \tag{5.43}$$

5.4.3 Birth and Death Moves

In Death/Birth moves, we first made a random choice between birth and death with the same a_m and b_M as above. If the birth move was chosen, the values of the parameters of the new components (μ_{j*},σ^2_{j*}) were derived from the associated prior distributions, given by Eqs. 5.17 and 5.18, respectively. Also, the mixing weight of the new component was drawn from:

$$p_{j*} \sim \mathscr{B}e(1,M) \tag{5.44}$$

In order to keep the constraint $\sum_{j=1}^{M} p_j + p_{j*} = 1$ true, we re-scale the previous value of $p_j, j = 1:M$, by multiplying them with $1-p_{j*}$. The acceptance probabilities for the birth and death are $min1,R$ and $min1,R^-1$, respectively, where:

$$R = \frac{p(M+1)}{p(M)}\frac{1}{\mathscr{B}(\delta,M\delta)}p_{j*}^{\delta-1}(1-p_{j*})^{N+M\delta-M}(M+1)\frac{a_{M+1}}{M_0 b_M}\frac{1}{p(p_{j*})}(1-p_{j*}^M) \tag{5.45}$$

where $\mathscr{B}$ is a beta function and M_0 is the number of empty components before the birth.

5.5 Experimental Results

In this section, experiments were carried out in order to evaluate the benefits of using the proposed model. The simulations were conducted on both synthetic and real data extracted from a challenging application, namely, object detection.

5.5.1 Synthetic Data

This section is dedicated to the generation of datasets. Its main aim is to investigate the ability of our algorithm to estimate the mixture parameters and to select the number of clusters correctly. We generated three different multidimensional datasets (3-dimensional) from the GID mixture model. The first dataset was generated from a 2-component model. The second one was generated from a 3-component mixture. The third dataset was generated from a 4-component model. Table 5.1 shows the real and estimated parameters obtained for these datasets. On the other hand, Table 5.2 shows the estimated posterior probabilities for the considered number of components for the three datasets, as well as the percentage of accepted split-combine and birth-death moves. According to this table, it is clear that our algorithm predicts the correct number of components each time. Figure 5.1 shows how the algorithm moves between the components, which is shown by plotting

Table 5.1: Real parameters used to generate the synthetic data sets (n_j represents the number of elements in cluster j) and the estimated ones using our RJMCMC algorithm.

		Real Parameters			Estimated Parameters		
Dataset	j	μ_j	σ_j^2	p_j	$\hat{\mu}_j$	$\hat{\sigma}_j^2$	$\hat{p}_j$
Dataset1	1	1.00	2.00	0.50	0.98	2.10	0.45
	2	7.00	4.00	0.50	7.50	6.00	0.55
Dataset2	1	1.00	1.25	0.33	0.98	1.56	0.26
	2	4.50	2.00	0.33	4.57	3.62	0.34
	3	9.00	3.30	0.33	9.37	5.17	0.40
Dataset3	1	1.00	1.25	0.25	0.86	1.21	0.21
	2	3.50	2.50	0.25	2.21	6.7	0.24
	3	6.50	1.67	0.25	2.24	6.80	0.25
	4	12.00	11.11	0.25	12.07	14.06	0.30

the number of components as a function of the number of sweeps. According to the obtained results, we can conclude that our algorithm has an excellent learning ability.

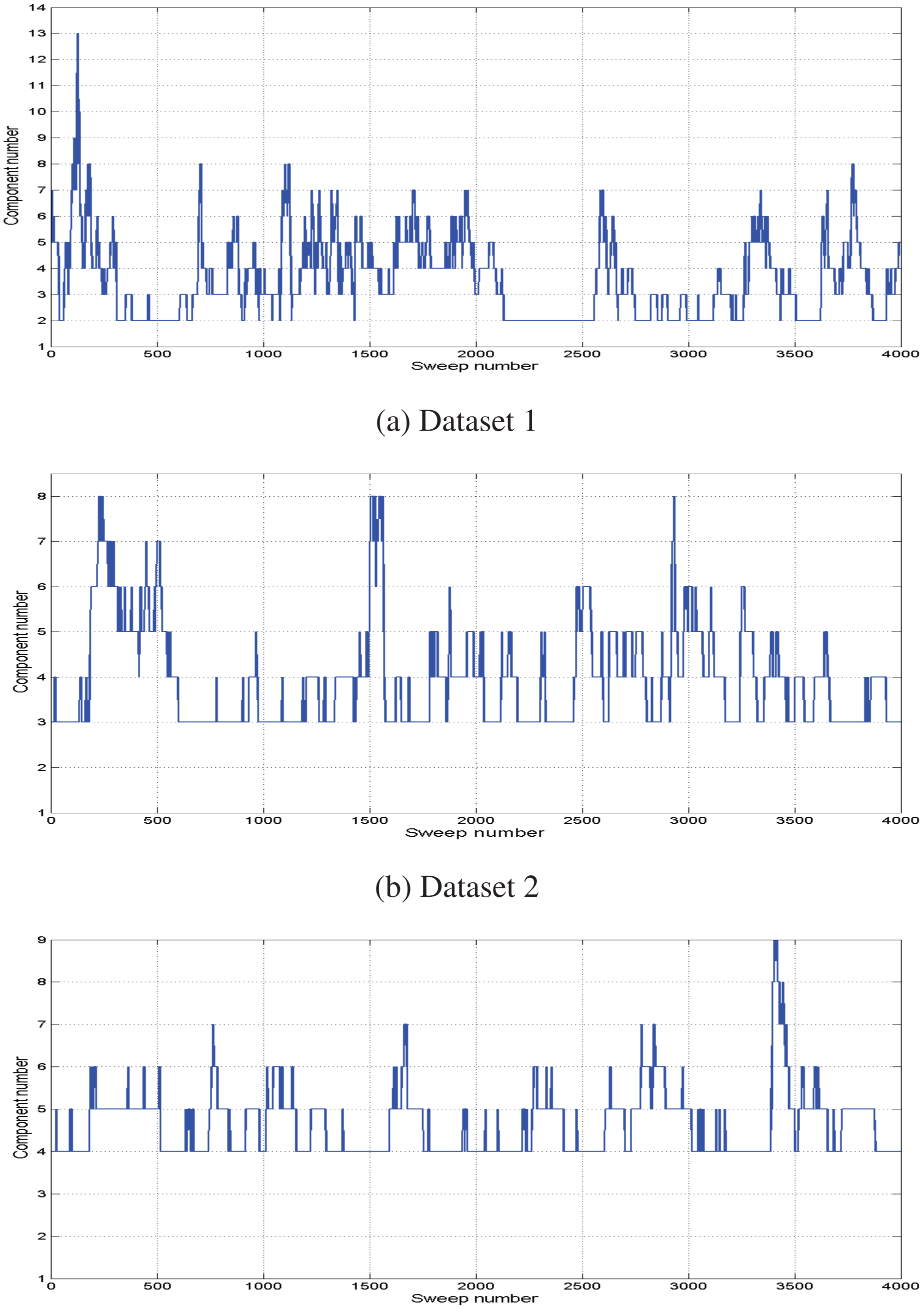
Table 5.2: The estimated posterior probabilities of the number of components given the data for the three datasets

Datasets	N	$p(k\|y)$		
Dataset 1	200	$p(1\|y) = 0.0017$ $p(4\|y) = 0.0113$	**p(2|y)= 0.9123** $p(5\|y) = 0.0050$	$p(3\|y) = 0.0697$ $p(> 5\|y) = 0.0$
Dataset 2	300	$p(1\|y) = 0.0003$ $p(4\|y) = 0.2948$	$p(2\|y) = 0.1903$ $p(5\|y) = 0.1420$	**p(3|y)=0.4703** $p(> 5\|y) = 0.0929$
Dataset 3	400	$p(1\|y) = 0.0$ **p(4|y)=0.4180**	$p(2\|y) = 0.0$ $p(5\|y) = 0.2190$	$p(3\|y) = 0.2657$ $p(> 5\|y) = 0.0973$

5.5.2 Object Detection

Advances in multimedia technology has caused an exponential increase in the number of images generated daily. This huge amount of visual data needs to be efficiently organized and indexed. To solve this problem, a lot of different approaches have been developed [175, 176]. Object detection is an important and challenging problem related to content-based image indexing and retrieval. It has several applications such as video surveillance and object recognition. In this section, we shall focus on the application of our model to pedestrian and car detection problems.

An important step in object detection is the extraction of low level features to describe the images. Many visual descriptors have been proposed in the past (see, for instance, [177]). Here, we use the local Histogram of Oriented Gradient (HOG) descriptor, which generates positive features and has been shown to be efficient and convenient for different object detection tasks [83]. Experiments were conducted by considering three windows for the HOG descriptor. This allowed each image to be represented by an 81-dimensional vector of features. The experimental results were conducted by considering our proposed GID mixture model using the RJMCMC-based learning model, (GID-RJMCMC). The performance obtained by the proposed model (GIDM-RJMCMC) was compared with the GID and GMM mixture models using the EM-based learning [178] (GIDFS/GMMFS).



(a) Dataset 1

(b) Dataset 2

(c) Dataset 3

Figure 5.1: The number of components vs. the number of sweeps for (a) Dataset 1, (b) Dataset 2, and (c) Dataset 3.

This was performed both without and with feature selection (GIDnoFS/GMMnoFS), as developed in [44].

Car Detection

The dataset that we consider here contains images of car side views which were collected at UIUC [1]. The dataset consists of 1050 images (550 car and 500 non-car images). Figures 5.2 and 5.3 show examples of images from this dataset. The first 100 images from both car and non-car sets were used for training and the rest for testing.

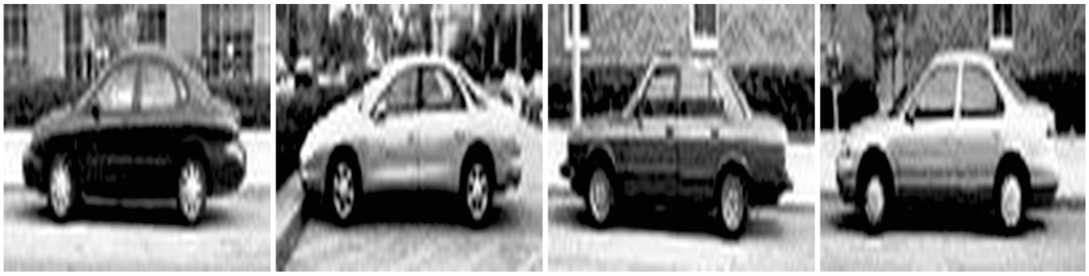


Figure 5.2: Examples of car images.

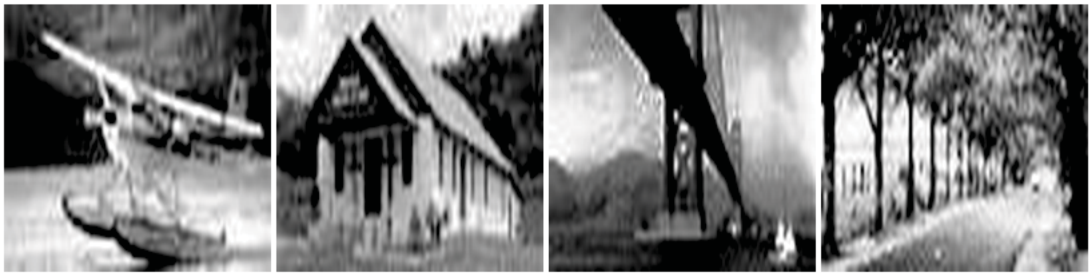


Figure 5.3: Examples of non-car images.

Table 5.3 shows the detection accuracies using GIDM-RJMCMC, Gaussian, and GID mixtures learned via EM with and without feature selection. According to this table, it is clear that the GIDM-RJMCMC outperforms the other tested approaches.

[1] http://cogcomp.cs.illinois.edu/Data/Car/

Table 5.3: Car detection accuracy when different approaches were considered.

Model	Accuracy (%)
GIDM-RJMCMC	**85.88%**
GIDFS	**84.59%**
GIDnoFS	**80.76%**
GMMFS	**74.00%**
GMMnoFS	**72.77%**

Human Detection

Another challenging task, that we considered here, is human detection. We considered the INRIA Static Person dataset [2] to evaluate the proposed model. The data consists of both positive (containing humans) and negative examples (images that do not contain humans). 400 images were used for training (200 positive examples and 200 negative ones). On the other hand, the testing set consisted of 741 images, 288 of them were positive examples and the remaining 453 were negative examples. Figures 5.4 and 5.5 show samples of positive and negative examples, respectively.

Table 5.4 shows the classification accuracy for the INRIA dataset. According to this table, it is clear that the proposed model GIDM-RJMCMC outperforms GIDFS, GIDnoFS, GMMFS, and GMMnoFS. On the other hand, for model selection, and for both datasets (car and human), our algorithm successfully determined the correct number of components as shown in Table 5.5.

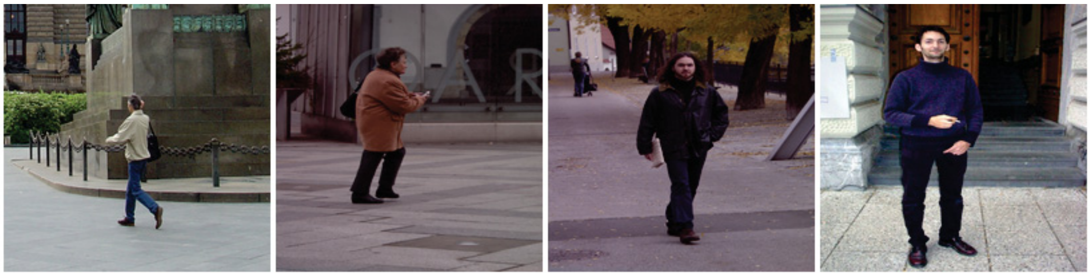


Figure 5.4: Examples of images containing humans.

[2] http://pascal.inrialpes.fr/data/human/

Figure 5.5: Examples of negative images used for human detection task.

Model	Accuracy (%)
GIDM-RJMCMC	**72.60%**
GIDFS	**68.55%**
GIDnoFS	**65.56%**
GMMFS	**57.35%**
GMMnoFS	**53.00%**

Table 5.4: Human detection accuracies when different approaches were considered.

Datasets		$p(k\|y)$	
Car Detection	$p(1\|y) = 0.1361$	**p(2|y)= 0.7810**	$p(3\|y) = 0.0571$
Dataset	$p(4\|y) = 0.0223$	$p(5\|y) = 0.0027$	$p(> 5\|y) = 0.0007$
Human Detection	$p(1\|y) = 0.4049$	**p(2|y)=0.5660**	$p(3\|y) = 0.0291$
Dataset	$p(4\|y) = 0.0000$	$p(5\|y) = 0.0000$	$p(> 5\|y) = 0.0000$

Table 5.5: The estimated posterior probabilities of the number of components for the car and human datasets.

5.6 Summary

In this chapter, a fully Bayesian mixture of GID distributions were developed using RJMCMC methods. The GID distribution has a very nice property which allows for the representation of GID samples in a transformed space in which features are independent and follow the inverted beta distributions. In this chapter, the inverted beta distribution was represented by its mean and variance. A fully Bayesian estimation was used to estimate the mean and variance and the related hyperparameters. In general, the framework proposed was based on RJMCMC, which is capable of jumping between the parameter subspace corresponding to different numbers of mixture components. Moreover, we were able to simultaneously estimate the model parameters and the number of components. In order to demonstrate the merits of the proposed model, generated datasets composed of two, three, and four components, as well as the real life problem of object detection, were used.

Chapter 6

Conclusion and Future Work

Clustering plays a crucial role in various data mining and knowledge discovery applications. The majority of existing clustering algorithms assume that clusters follow Gaussian distributions. This assumption is not practical in the majority of signal and image processing problems. This is the case of positive data that are naturally appear in several real life applications. The inverted Dirichlet distribution has been proposed for modeling such data. However, it suffers from major drawbacks such as its very restrictive, strictly positive covariance structure. In this thesis, we considered applying the generalized inverted Dirichlet in order to overcome this limitation. Our approach achieved the clustering by representing the data using a mixture model of GID distributions. In fact, based on this distribution, throughout this thesis, we have developed different estimation approaches.

Feature selection algorithms, which are based on mixture models, assume that the data in each component follow Gaussian distribution. Unlike these approaches, in Chapter 2, we proposed an unsupervised feature selection robust to outliers. The developed statistical framework in Chapter 2 was based on the finite GID mixture model and learned via the minimization of a message length objective by deploying an EM framework. We have presented an algorithm in which the problems of model selection, determining the relevant features, and outlier rejection were tackled simultaneously. The great advantage of our learning approach was that it allowed a compromise between goodness of fit and a model‘s complexity. Empirical results which involve generated data, as well as real applications concerning visual scenes and objects classification, show that the proposed approach was promising. In terms of feature selection, our model not only maintained the accuracy, but improved it. Also, it showed the influence of outlier observations on the performance of the developed framework.

Finite mixture modeling addressed two problems: parameter estimation and selection of the number of clusters. Computational approaches like EM [80] suffer from several drawbacks [91]. Most of them are optimization drawbacks such as the appearance of local maxima and singularities. Moreover, in the case of high dimensional data, it is hard to obtain a reliable estimation, which is the capability to predict the densities at new data points [68]. Given a proper prior, a Bayesian approach to the mixture estimation problem always provides estimations which can be written explicitly for conjugate priors [122]. In order to overcome the problems related to EM-based estimation, we developed and evaluated a comprehensive framework for the Bayesian learning of finite GID mixture models in Chapter 3. The Bayesian learning was performed by considering the Gibbs sampling, with the Metropolis-Hastings algorithm, which allows obtaining Monte Carlo estimates from the model's posterior quantities of interest. We addressed various aspects of the inference problem, including the choice of the priors and the selection of optimal models from the simulation outputs using the BIC criterion. The proposed approach has been validated using two problems that have received widespread research interest, namely, object classification and forgery detection. According to the obtained results, we can first point out that it is important to pay attention to both modeling assumptions and flexibility. Second, we can conclude that choosing a single model, as done in the case of the ML approach, may be not satisfactory. A better approach is to consider a set of plausible models built from the observed data and from prior knowledge that we need to select.

In Chapter 4, a statistical approach representing the data using an infinite mixture model of GID distributions which used a feature weighting component, was introduced. Feature selection was introduced in order to remove irrelevant features that may compromise the clustering process. Our simulations, based on the challenging problem of image categorization, have shown the efficiency of the proposed model.

In Chapter 5 we were able to establish a reversible jump MCMC algorithm for a full Bayesian analysis of Generalized Inverted Dirichlet mixtures. The proposed learning approach allowed for the simultaneous model selection and parameter estimations in one single algorithm. We presented experimental results using synthetic data and a challenging real-life application, namely, object detection. According to the obtained results, it is clear that the proposed approach is promising.

Although good results have been obtained by using the Bayesian inference, its computational (proposed in Chapter 3) cost is high. As a deterministic alternative to fully Bayesian learning, some researchers have turned their attention to variational learning. Thus, a potential future work could be devoted to the development of a variational inference framework for this model as described in [179]. Future work could also be devoted to the extension of the proposed framework in order to handle semi-supervised learning, which has been shown to offer significant advantages in several applications [180]. Several other directions present themselves for future efforts. Indeed, future work could be devoted to the application of the proposed work to problems where the rare events (i.e. outliers) are more interesting than inliers. Two such problems are the detection of credit card fraud and the monitoring of criminal activities. Moreover, the introduction of feature selection, as shown in [44], to a GID based on RJMCMC methodology, may improve the learning results.

Bibliography

[1] U.Von. Luxburg and S. Bubeck and S. Jegelka and M. Kaufmann. Consistent Minimization of Clustering Objective Functions. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[2] O. Shamir and N. Tishby. Cluster Stability for Finite Samples. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[3] I. Gitman and M.D. Levine. An Algorithm for Detecting Unimodal Fuzzy Sets and Its Application as a Clustering Technique. *IEEE Transactions on Computers*, 19(7):583–593, 1970.

[4] W.L.G. Koontz and K. Fukunaga. A Nonparametric Valley-Seeking Technique for Cluster Analysis. *IEEE Transactions on Computers*, 21(2):171–178, 1972.

[5] J.A. Hartigan. Direct Clustering of a Data Matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

[6] M. Ankerst and M.M. Breunig and H.P. Kriegel and J. Sander. OPTICS: Ordering Points to Identify the Clustering Structure. In *ACM Sigmod Record*, volume 28, pages 49–60. ACM, 1999.

[7] B.S. Everitt and S. Landau and M. Leese. *Cluster Analysis*. Wiley, 4th edition, January 2009.

[8] T. Bdiri and N. Bouguila. Positive Vectors Clustering Using Inverted Dirichlet Finite Mixture Models. *Expert Systems With Applications*, 39(2):1869–1882, 2012.

[9] T. Bdiri and N. Bouguila. Learning Inverted Dirichlet Mixtures for Positive Data Clustering. In *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, pages 265–272, 2011.

[10] A.K. Gupta and S. Nadarajah. Exact and Approximate Distributions for the Linear Combination of Inverted Dirichlet Components. *Journal of The Japan Statistical Society*, 36(2):225–236, 2006.

[11] G.G. Tiao and I. Cuttman. The Inverted Dirichlet Distribution with Applications. *Journal of the American Statistical Association*, 60(311):793–805, 1965.

[12] A.P. Dempster and N.M. Laird and D.B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the royal statistical society. Series B*, 39(1):1–38, 1977.

[13] G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*, volume 382. John Wiley & Sons, 2007.

[14] G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978.

[15] A. Barron and J. Rissanen and B. Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, 1998.

[16] H. Akaike. A New Look at the Statistical Model Identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

[17] G. Schwarz. Estimating the Dimension of a Model. *Annals of Statistics*, 6:461–464, 1978.

[18] D.J. Spiegelhalter, N.G. Best, B.P. Carlin, and A.V.D. Linde. Bayesian Measures of Model Complexity and Fit. *Journal of the Royal Statistical Society: Series B*, 64(4):583–639, 2002.

[19] M.H.C. Law and M.A.T. Figueiredo and A.K. Jain. Simultaneous Feature Selection and Clustering Using Mixture Models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1154 –1166, sept. 2004.

[20] M. Stephens. Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods. *Annals of Statistics*, 28(1):40–74, 2000.

[21] S. Richardson and J.P. Green. On Bayesian Analysis of Mixtures with an Unknown Number of Components. *Journal of the Royal Statistical Society. Series B*, 59(4):731–792, 1997.

[22] H. Zheng and Y. Zhang. Feature Selection for High Dimensional Data in Astronomy. *Advances in Space Research*, 41:1960–1964, 2008.

[23] P. Indyk and R. Motwani. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998.

[24] S. Berchtold, C. Böhm, D.A. Keim, and H.P. Kriegel. A Cost Model for Nearest Neighbor Search in High-Dimensional Data Space. In *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 78–86. ACM, 1997.

[25] J. Bayardo J. Roberto. Efficiently Mining Long Patterns from Databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 85–93, 1998.

[26] R. Weber, H.J. Schek and S. Blott. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In *Proceedings of 24rd International Conference on Very Large Data Bases (VLDB)*, pages 194–205, 1998.

[27] A. Gionis, P. Indyk and R. Motwani. Similarity Search in High Dimensions via Hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB)*, pages 518–529, 1999.

[28] P.E. Jouve and N. Nicoloyannis. A Filter Feature Selection Method for Clustering. In *Proceedings of the 15th International Symposium on Foundations of Intelligent Systems (ISMIS)*, pages 583–593, 2005.

[29] H.P. Kriegeland P. Kröger and A. Zimek. Clustering High-Dimensional Data: A Survey on Subspace Clustering, Pattern-Based Clustering, and Correlation Clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):1, 2009.

[30] E.P. Xing, M.I. Jordan and R.M. Karp. Feature Selection for High-Dimensional Genomic Microarray Data. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 601–608, 2001.

[31] L. Lazzeroni and A. Owen. Plaid Models for Gene Expression Data. *Statistica Sinica*, 12:61–86, 2002.

[32] P. Marttinen, J. Corander, P. Törönen, and L. Holm. Bayesian Approach of Functionally Divergent Protein Subgroups and their Function Specific Residues. *Bioinformatics*, 22(20):2466–2474, 2006.

[33] T. Fawcett and F. Provost. Adaptive Fraud Detection. *Data Mining and Knowledge Discovery*, 1:291–316, 1997.

[34] E. Gabrilovich and S. Markovitch. Text Categorization With Many Redundant Features: Using Aggressive Feature Selection to Make SVMs Competitive With C4.5. In *Proceedings of the twenty-first international conference on Machine learning (ICML)*, pages 321–328, 2004.

[35] T. Fawcett and F. Provost. Combining Data Mining and Machine Learning for Effective User Profiling. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 8–13, 1996.

[36] M.H.C. Law, M.A.T. Figueiredo and A.K. Jain. Simultaneous Feature Selection and Clustering Using Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1154–1166, 2004.

[37] N. Bouguila, K. Almakadmeh and S. Boutemedjet. A Finite Mixture Model for Simultaneous High-Dimensional clustering, localized feature selection and outlier rejection. *Expert Systems with Applications*, 39(7):6641–6656, 2012.

[38] E.I. George. The Variable Selection Problem. *Journal of the American Statistical Association*, 95(452):1304–1308, 2000.

[39] E.I. George and D. P. Foster. Calibration and Empirical Bayes Variable Selection. *Biometrika*, 87(4):731–747, 2000.

[40] H. Frigui and O. Nasraoui. Unsupervised Learning of Prototypes and Attribute Weights. *Pattern Recognition*, 37(3):567–581, 2004.

[41] M.D. Hawkins. *Identification of Outliers*. Chapman and Hall, London ; New York :, 1980.

[42] V. Barnett and T. Lewis. *Outliers in Statistical Data*. Wiley Series in Probability & Statistics. Wiley, April 1994.

[43] S. Hawkins, H. He, G. Williams, and R. Baxter. Outlier Detection Using Replicator Neural Networks. In *Proceedings of the Fifth International Conference and Data Warehousing and Knowledge Discovery*, pages 170–180, 2002.

[44] M. Al Mashrgy and T. Bdiriand and N. Bouguila. Robust Simultaneous Positive Data Clustering and Unsupervised Feature Selection Using Generalized Inverted Dirichlet Mixture Models. *Knowledge-Based Systems*, 59:182–195, 2014.

[45] S. Bourouis and M. Al Mashrgy and N. Bouguila. Bayesian Learning of Finite Generalized Inverted Dirichlet Mixtures: Application to Object Classification and Forgery Detection. *Expert Systems with Applications*, 41(5):2329–2336, 2014.

[46] N. Bouguila and M. Al Mashrgy. An Infinite Mixture Model of Generalized Inverted Dirichlet Distributions for High-Dimensional Positive Data Modeling. In *Information and Communication Technology-International Conference, ICT-EurAsia*, pages 296–305, 2014.

[47] A.D. Gordon. Constructing Dissimilarity Measures. *Journal of Classification*, 7(2):257–269, 1990.

[48] R. Gnanadesikan, J.R. Kettenring and S.L. Tsao. Weighting and Selection of Variables for Cluster Analysis. *Journal of Classification*, 12(1):113–136, 1995.

[49] H. Liu and R. Setiono. Dimensionality Reduction via Discretization. *Knowledge-Based Systems*, 9:67–72, 1996.

[50] J.B. Copas. Regression, Prediction and Shrinkage. *Journal of the Royal Statistical Society. Series B*, 45(3):311–354, 1983.

[51] M. Charikar, V. Guruswami, R. Kumar, S. Rajagopalan and A. Sahai. Combinatorial Feature Selection Problems. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 631–640, 2000.

[52] X. Wang and E.I. George. Adaptive Bayesian Criteria in Variable Selection for Generalized Linear Models. *Statistica Sinica*, 17:667–690, 2007.

[53] S. Boutemedjet and N. Bouguila and D. Ziou. A Hybrid Feature Extraction Selection Approach for High-Dimensional Non-Gaussian Data Clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(8):1429–1443, 2009.

[54] C. Keribin. Consistent Estimation of the Order of Mixture Models. *The Indian Journal of Statistics*, 62, Series A(1):49–66, 2000.

[55] J. Rissanen. Hypothesis Selection and Testing by the MDL Principle. *The Computer Journal*, 42(4):260–269, 1999.

[56] C.S. Wallace. *Statistical and Inductive Inference by Minimum Message Length*. Springer-Verlag, 2005.

[57] A. Chaturvedi, J. D. Carroll, P. E. Green and J.A. Rotondo. A Feature-Based Approach to Market Segmentation Via Overlapping K-Centroids Clustering. *Journal of Marketing Research*, 34(3):370–377, 1997.

[58] A. Hinneburg and D.A. Keim. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 58–65, 1998.

[59] G. Sheikholeslami, S. Chatterjee and A. Zhang. WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases. In *Proceedings of 24rd International Conference on Very Large Data Bases (VLDB)*, pages 428–439, 1998.

[60] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 157–166, 2005.

[61] M.S. Yang and K.L. Wu. Unsupervised Possibilistic Clustering. *Pattern Recognition*, 39:5–21, 2006.

[62] R. Gottardo and A. Raftery. Bayesian Robust Transformation and Variable Selection: A Unified Approach. *The Canadian Journal of Statistics*, 37(3):361–380, 2009.

[63] G.S. Lingappaiah. On the Generalised Inverted Dirichlet Distribution. *Demonstratio Mathematica*, 9:423–433, 1976.

[64] N. Bouguila. A Model-Based Approach for Discrete Data Clustering and Feature Weighting Using MAP and Stochastic Complexity. *IEEE Transactions on Knowledge and Data Engineering*, 21(12):1649–1664, 2009.

[65] Y. Ohsawa, and P. McBurney. *Chance Discovery*. Springer, 2003.

[66] M. Markou and S. Singh. Novelty Detection: A Review- Part I: Statistical Approaches. *Signal Processing*, 83(12):2481–2497, 2003.

[67] E. Suzuki and J. M. Zytkow. Unified Algorithm for Undirected Discovery of Exception Rules. In *Principles of Data Mining and Knowledge Discovery*, pages 169–180, 2000.

[68] M.V. Joshi and V. Kumar. CREDOS: Classification Using Ripple Down Structure (A Case for Rare Classes). In *SDM*, 2004.

[69] P. Meer, D. Mintz and A. Rosenfeld, and D.Y. Kim . Robust Regression Methods for Computer Vision: A Review. *International Journal of Computer Vision*, 6:59–70, 1991.

[70] K.L. Lange, R.J.A. Little and J.M.G. Taylor. Robust Statistical Modeling Using the t Distribution. *Journal of the American Statistical Association*, 84(408):881–896, 1989.

[71] I. Ruts and P.J. Rousseeuw. Computing Depth Contours of Bivariate Point Clouds. *Computational Statistics and Data Analysis*, 23:153–168, 1996.

[72] E.M. Knorr and R.T. Ng. Finding Intensional Knowledge of Distance-Based Outliers. In *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB)*, pages 211–222, 1999.

[73] E.M. Knorr and R.T. Ng and R.H. Zamar. Robust Space Transformations for Distance-Based Operations. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 126–135, 2001.

[74] M. Hubert and S. Engelen. Robust PCA and Classification in Biosciences. *Bioinformatics*, 20(11):1728–1736, 2004.

[75] X. Zhuang, T. Wang and P. Zhang. A Highly Robust Estimator Through Partially Likelihood Function Modeling and Its Application in Computer Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):19–35, 1992.

[76] M.K. Titsias and C.K.I. Williams. Sequentially Fitting Mixture Models using an Outlier Component. In *Proceedings of the 6th International Workshop on Advances in Scattering and Biomedical Engineering*, pages 386–393, 2003.

[77] N. Bouguila. Count Data Clustering Using Unsupervised Localized Feature Selection and Outliers Rejection. In *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*, pages 1020–1027, Nov 2011.

[78] D.F Andrews. A Robust Method for Multiple Linear Regression. *Technometrics*, 16(4):523–531, 1974.

[79] B. Zhang. Generalized K-Harmonic Means-Dynamic Weighting of Data in Unsupervised Learning. In *Proceedings of the First SIAM International Conference on Data Mining (SDM)*, pages 1–13, 2001.

[80] C. Robert. *The Bayesian choice*. Springer-Verlag, 2001.

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

[82] A. Oliva and A. Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

[83] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[84] K. Levi and Y. Weiss. Learning Object Detection from a Small Number of Examples: The Importance of Good Features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages II–53–II–60, 2004.

[85] J. Marroquin, S. Mitter, and T. Poggio. Probabilistic Solution of Ill-Posed Problems in Computational Vision. *Journal of the American Statistical Association*, 82(397):76–89, 1987.

[86] J.A. Achcar, and G.D.A. Pereira. Use of Exponential Power Distributions for Mixture Models in the Presence of Covariates. *Journal of Applied Statistics*, 26(6):669–679, 1999.

[87] J. Besag and P.J. Green. Spatial Statistics and Bayesian Computation. *Journal of the Royal Statistical Society, Series B*, 55(1):25–37, 1993.

[88] C.J. Geyer. On the Convergence of Monte Carlo Maximum Likelihood Calculations. *Journal of the Royal Statistical Society, Series B*, 56(1):261–274, 1994.

[89] P. Mykland, L. Tierney, and B. Yu. Regeneration in Markov Chain Samplers. *Journal of the American Statistical Association*, 90(429):233–241, 1995.

[90] P. Fearnhead. Direct Simulation for Discrete Mixture Distributions. *Statistics and Computing*, 15(2):125–133, 2005.

[91] C. Robert, and G. Casella. *Monte Carlo Statistical Methods*. Springer, second edition, 2004.

[92] S.J. Godsill. On the Relationship Between MCMC Model Uncertainty Methods. *Journal of Computational and Graphical Statistics*, 10:230–248, 2001.

[93] D.J. Spiegelhalter, N.G. Best,B.P. Carlin, and A.V.D. Linde. Bayesian Deviance, Deviance, the Effective Number of Parameters, and the Comparison of Arbitrarily Complex Models. Technical Report 98-009, Cambridge University, 1998.

[94] R.E. Kass and A.E. Raftery. Bayes Factors. *Journal of the American Statistical Association*, 90:773–795, 1995.

[95] M. Stephens. Bayesian Analysis of Mixture Models with an Unknown Number of Components- An Alternative to Reversible Jump Methods. *The Annals of Statistics*, 28(1):40–74, 2000.

[96] A.E. Gelfand and B.P. Carlin. Maximum-Likelihood Estimation for Constrained- or Missing-Data Models. *The Canadian Journal of Statistics*, 21(3):303–311, 1993.

[97] J. Besag, P. Green, D. Higdon, and K. Mengersen. Bayesian Computation and Stochastic Systems. *Statistical Science*, 10(1):3–66, 1995.

[98] A.E. Gelfand, S.E. Hills, A. Racine-Poon and A.F.M. Smith. Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling. *Journal of the American Statistical Association*, 85(412):972–985, 1990.

[99] D. Geiger and D. Heckerman. Parameters Priors for Directed Acyclic Graphical Models and the Characterization of Several Probability Distributions. *The Annals of Statistics*, 30(5):1412–1440, 2002.

[100] P.M. Lee. *Bayesian Statistics: An Introduction*. John Wiley & Sons, 2012.

[101] T. Bengtsson and J.E. Cavanaugh. An Improved Akaike Information Criterion for State-Space Model Selection. *Computational Statistics & Data Analysis*, 50(10):2635–2654, 2006.

[102] L. Tierney and J.B. Kadane. Accurate Approximations for Posterior Moments and Marginal Densities. *Journal of the American Statistical Association*, 81(393):82–86, 1986.

[103] L. Tierney, R.E. Kass, and J.B. Kadane. Fully Exponential Laplace Approximations to Expectations and Variances of Nonpositive Functions. *Journal of the American Statistical Association*, 84(407):710–716, 1989.

[104] S. Chib and E. Greenberg. Analysis of Multivariate Probit Models. *Biometrika*, 85(2):347–361, 1998.

[105] A. Frigessi, P. Di Stefano, C.R. Hwang, and S.J. Sheu. Convergence Rates of the Gibbs Sampler, the Metropolis Algorithm and Other Single-site Updating Dynamics. *Journal of the Royal Statistical Society. Series B*, 55(1):205–219, 1993.

[106] A.E. Raftery and S.M. Lewis. One Long Run with Diagnostics: Implementation Startegies for Markov Chain Monte Carlo. *Statistical Science*, 7(4):493–497, 1992.

[107] A.E. Raftery, and S. M. Lewis. Implementing MCMC. In *Markov Chain Monte Carlo in Practice*, pages 115–130. London: Chapman and Hall, 1996.

[108] G.D. Kleiter. Bayesian Diagnosis in Expert Systems. *Artificial Intelligence*, 54(1-2):1–32, 1992.

[109] E. Castillo and A.S. Hadi and C. Solares. Learning and Updating of Uncertainty in Dirichlet Models. *Machine Learning*, 26(1):43–63, 1997.

[110] K.C. Yow and R. Cipolla. Feature-Based Human Face Detection. *Image and Vision Computing*, 15(9):713–735, 1997.

[111] P. Aigrain and H.J. Zhang and D. Petkovic. Content-Based Representation and Retrieval of Visual Media: A State-of-the-Art Review. *Multimedia Tools and Applications*, 3(3):179–202, 1996.

[112] E. Rivlin, S.J. Dickinson, and A. Rosenfeld. Recognition by Functional Parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 267–274. IEEE, 1994.

[113] Z. Biuk and S. Loncaric. Face Recognition From Multi-Pose Image Sequence. In *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 319–324. IEEE, 2001.

[114] B. Moghaddam and A. Pentland. Probabilistic Visual Learning for Object Detection. In *Proceedings of the Fifth IEEE International Conference on Computer Vision (ICCV)*, pages 786–793. IEEE, 1995.

[115] H. Wang, P. Li and, T. Zhang. Histogram Feature-Based Fisher Linear Discriminant for Face Detection. *Neural Computing and Applications*, 17(1):49–58, 2008.

[116] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of Adjacent Contour Segments for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):36–51, 2008.

[117] Y. Adini, Y. Moses and S. Ullman. Face Recognition: The Problem of Compensating for Changes in Illumination Direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):721–732, 1997.

[118] A. Torralba. Modeling Global Scene Factors in Attention. *Journal of the Optical Society of America A*, 20(7):1407–1418, 2003.

[119] T. Serre, L. Wolf and T. Poggio. Object Recognition with Features Inspired by Visual Cortex. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages II.994–II.1000. IEEE, 2005.

[120] K.M. Chen and S.Y. Chen. Color Texture Segmentation Using Feature Distributions. *Pattern Recognition Letters*, 23(7):755–771, 2002.

[121] W.S. Zheng and S. Gong and T. Xiang. Quantifying and Transferring Contextual Information in Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):762–777, 2012.

[122] F. Jurie and C. Schmid. Scale-Invariant Shape Features for Recognition of Object Categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages II.90–II.96. IEEE, 2004.

[123] V. Ferrari,T. Tuytelaars, and L.J.V. Gool. Object Detection by Contour Segment Networks. In *Proceedings of the 9th European Conference on Computer Vision (ECCV)*, pages 14–28. Springer, 2006.

[124] H. Farid. Image Forgery Detection. *IEEE Signal Processing Magazine*, 26(2):16–25, 2009.

[125] A. Swaminathan and M. Wu and K.J.R. Liu. Digital Image Forensics via Intrinsic Fingerprints. *IEEE Transactions on Information Forensics and Security*, 3(1):101–117, 2008.

[126] A.C. Popescu, and H. Farid. Exposing Digital Forgeries in Color Filter Array Interpolated Images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959, 2005.

[127] A.C. Popescu and H. Farid. Exposing Digital Forgeries by Detecting Traces of Resampling. *IEEE Transactions on Signal Processing*, 53(2):758–767, 2005.

[128] I. Avcibas, S. Bayram, N.Memon, M. Ramkumar, and B. Sankur. A classifier Design for Detecting Image Manipulations. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 2645–2648. IEEE, 2004.

[129] Farid, H. Exposing Digital Forgeries From JPEG Ghosts. *IEEE Transactions on Information Forensics and Security*, 4(1):154–160, 2009.

[130] G. Li and Q. Wu and D. Tu and S. Sun. A Sorted Neighborhood Approach for Detecting Duplicated Regions in Image Forgeries Based on DWT and SVD. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1750–1753. IEEE, 2007.

[131] W. Luo and J. Huang and G. Qiu. Robust Detection of Region-Duplication Forgery in Digital Image. In *Proceedings of the18th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 746–749. IEEE, 2006.

[132] E. Ardizzone and A. Bruno and G. Mazzola. Detecting Multiple Copies in Tampered Images. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 2117–2120. IEEE, 2010.

[133] H. Huang and W. Guo and Y. Zhang. Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm. In *Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA)*, volume 2, pages 272–276. IEEE, 2008.

[134] I. Amerini and L. Ballan and R. Caldelli and A.D. Bimbo and G. Serra. A SIFT-Based Forensic Method for Copy-Move Attack Detection and Transformation Recovery. *IEEE Transactions on Information Forensics and Security*, 6(3-2):1099–1110, 2011.

[135] V. Christlein and C. Riess and J. Jordan and C. Riess and E. Angelopoulou. An Evaluation of Popular Copy-Move Forgery Detection Approaches. *IEEE Transactions on Information Forensics and Security*, 7(6):1841–1854, 2012.

[136] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[137] I. Amerini and L. Ballan and R. Caldelli and A.D Bimbo and G. Serra. Geometric Tampering Estimation by Means of a SIFT-Based Forensic Analysis. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 1702–1705. IEEE, 2010.

[138] A.P. Topchy and M.H.C. Law and A.K. Jain and A.L. Fred. Analysis of consensus partition in cluster ensemble. In *Data Mining, 2004. ICDM '04. Fourth IEEE International Conference on*, pages 225–232, Nov 2004.

[139] J.C. Bezdek, and R.J. Hathaway, . and J.M. Huband and C. Leckie, and R. Kotagiri. Approximate Clustering in Very Large Relational Data. *International Journal of Intelligent Systems*, 21(8):817–841, 2006.

[140] W. Chen and G. Feng. Spectral Clustering with Discriminant Cuts . *Knowledge-Based Systems*, 28(0):27 – 37, 2012.

[141] K.Y. Huang. A Hybrid Particle Swarm Optimization Approach for Clustering and Classification of Datasets . *Knowledge-Based Systems*, 24(3):420 – 426, 2011.

[142] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley series in probability and statistics. J. Wiley & Sons, 2000.

[143] T. Bdiri and N. Bouguila. An Infinite Mixture of Inverted Dirichlet Distributions. In *ICONIP (2)*, pages 71–78, 2011.

[144] N. Bouguila and D. Ziou. A Nonparametric Bayesian Learning Model: Application to Text and Image Categorization. In *Advances in Knowledge Discovery and Data Mining*, pages 463–474. Springer, 2009.

[145] N. Bouguila and D. Ziou. A Dirichlet Process Mixture of Generalized Dirichlet Distributions for Proportional Data Modeling. *Neural Networks, IEEE Transactions on*, 21(1):107–122, Jan 2010.

[146] J.M. Marin and C. Robert. *Bayesian Core: A Practical Approach to Computational Bayesian Statistics*. Springer Texts in Statistics. Springer, 2007.

[147] N. Bouguila and W. ElGuebaly. On Discrete Data Clustering. In *Advances in Knowledge Discovery and Data Mining*, pages 503–510. Springer, 2008.

[148] R.M. Neal. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 9(2):pp. 249–265, 2000.

[149] W.R. Gilks, and P. Wild. Algorithm AS 287: Adaptive Rejection Sampling from Log-Concave Density Functions. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 42(4):pp. 701–709, 1993.

[150] W. Fan and N. Bouguila and D. Ziou. Unsupervised Hybrid Feature Extraction Selection for High-Dimensional Non-Gaussian Data Clustering with Variational Inference. *Knowledge and Data Engineering, IEEE Transactions on*, 25(7):1670–1685, July 2013.

[151] A.B. Benitez, and S.F. Chang. Semantic Knowledge Construction from Annotated Image Collections. In *Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on*, volume 2, pages 205–208 vol.2, 2002.

[152] E. Chang and G. Kingshy and G. Sychay and G. Wu. CBSA: Content-Based Soft Annotation for Multimodal Image Retrieval Using Bayes Point Machines. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(1):26–38, Jan 2003.

[153] J. He and M. Li and H.J. Zhang and H. Tong and C. Zhang. Manifold-Ranking Based Image Retrieval. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, MULTIMEDIA '04, pages 9–16, New York, NY, USA, 2004. ACM.

[154] X.J. Wang and W.Y. Maand G.R. Xue and X. Li. Multi-model Similarity Propagation and Its Application for Web Image Retrieval. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, MULTIMEDIA '04, pages 944–951, New York, NY, USA, 2004. ACM.

[155] L. Spirkovska and M.B. Reid. Higher-Order Neural Networks Applied to 2D and 3D Object Recognition. *Machine Learning*, 1994.

[156] S. Lazebnik and C. Schmid and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178, 2006.

[157] L.J. Li and L. Fei-Fei. What, where and who? Classifying Events by Scene and Object Rrecognition. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007.

[158] N. Bouguila. Spatial Color Image Databases Summarization. In *the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 953–956, Honolulu, HI, Apr. 2007.

[159] N. Bouguila and D. Ziou. Online Clustering Via Finite Mixtures of Dirichlet and Minimum Message Length. *Engineering Applications of Artificial Intelligence*, 19(4):371–379, 2006.

[160] N. Bouguila and D. Ziou. Improving Content Based Image Retrieval Systems Using Finite Multinomial Dirichlet Mixture. In *The IEEE Workshop on Machine Learning for Signal Processing (MLSP)*, pages 23–32, Sao Luis, Brazil, Oct. 2004.

[161] N. Bouguila and D. Ziou. A Powerful Finite Mixture Model Based on the Generalized Dirichlet Distribution: Unsupervised Learning and Applications. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, pages 280–283, 2004.

[162] N. Bouguila and D. Ziou. Dirichlet-Based Probability Model Applied to Human Skin Detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 521–524, 2004.

[163] N. Bouguila and W. ElGuebaly. Discrete Data Clustering Using Finite Mixture Models. *Pattern Recognition*, 42(1):33–42, 2009.

[164] C. Robert. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer, 2nd edition, 2007.

[165] F.S. Schnatter. *Finite Mixture and Markov Switching Models*. Springer Verlag, 2006.

[166] T. Elguebaly and N. Bouguila. Bayesian Learning of Finite Generalized Gaussian Mixture Models on Images. *Signal Processing*, 91(4):801–820, 2011.

[167] P.J. Green. Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination. *Biometrika*, 82:711–732, 1995.

[168] Z. Zhang and K. Chan and Y. Wu and C. Chen. Learning a Multivariate Gaussian Mixture model with the Reversible Jump MCMC Algorithm. *Statistics and Computing*, 14(4):343–355, 2004.

[169] N. Bouguila and T. Elguebaly. A fully Bayesian Model Based on Reversible Jump MCMC and Finite Beta Mixtures for Clustering. *Expert Systems with Applications*, 39(5):5946–5959, 2012.

[170] N. Bouguila and D. Ziou and R.I. Hammoud. On Bayesian Analysis of a Finite Generalized Dirichlet Mixture Via a Metropolis-within-Gibbs Sampling. *Pattern Analysis and Applications*, 12(2):151–166, 2009.

[171] T. Elguebaly and N. Bouguila. A Bayesian Approach for the Classification of Mammographic Masses. In *Developments in eSystems Engineering (DeSE), 2013 Sixth International Conference on*, pages 99–104, Dec 2013.

[172] N. Bouguila and J.H. Wang and A.B. Hamza. Software Modules Categorization through Likelihood and Bayesian Analysis of Finite Dirichlet Mixtures. *Journal of Applied Statistics*, 37(2):235–252, 2010.

[173] G. Casella and E.I. George. Explaining the Gibbs Sampler. *The American Statistician*, 46(3):167–174, 1992.

[174] S. Chib and E. Greenberg. Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4):327–335, 1995.

[175] N. Bouguila and W. ElGuebaly. A Statistical Model for Histogram Refinement. In *Artificial Neural Networks-ICANN 2008*, pages 837–846. Springer, 2008.

[176] N. Bouguila and K. Daoudi. Learning Concepts from Visual Scenes Using a Binary Probabilistic Model. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–5, 2009.

[177] A. Rocha and S. Goldenstein. PR: More Than Meets the Eye. In *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[178] M. Al Mashrgy and N. Bouguila and K. Daoudi. A Statistical Framework for positive Data Clustering With Feature Selection: Application to Object Detection. In *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, pages 1–5. IEEE, 2013.

[179] W. Fan and N. Bouguila and D. Ziou. Variational Learning for Finite Dirichlet Mixture Models and Applications. *IEEE Transactions on Neural Networks and Learning Systems*, 23(5):762–774, 2012.

[180] Q. Tian and J. Yu and Q. Xue and N. Sebe. A New Analysis of the Value of Unlabeled Data in Semi-Supervised Learning for Image Retrieval. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1019–1022, 2004.

Appendix A

We know that the posterior probability is $p(j|\vec{Y}_i) \propto p_j p(\vec{Y}_i|\vec{\theta}_j)$, so every vector $\vec{Y}_i$ is assigned to its cluster j, such that $j = \arg\max_j p(j|\vec{Y}_i) = \arg\max_j p_j p(\vec{Y}_i|\vec{\theta}_j)$. For GID, it is possible to compute the posterior probability by examining the form of the product in Equation 2.7 by considering each feature separately. So if we want to consider the feature D, in Equation 2.7, it becomes, for a specific vector $\vec{Y}_i = (Y_1, Y_2, ..., Y_D)$:

$$\frac{1}{B(\theta_{jD})} Y_D^{\alpha_{jD}-1}(1+\sum_{l=1}^{D} Y_l)^{-\beta_{jD}-\alpha_{jD}+\beta_{j(D+1)}} \prod_{l=1}^{D-1} \frac{1}{B(\theta_{jl})} Y_D^{\alpha_{jl}-1}(1+\sum_{k=1}^{l} Y_k)^{-\beta_{jl}-\alpha_{jl}+\beta_{j(l+1)}} \tag{A.1}$$

where $\frac{1}{B(\theta_{jl})} = \frac{\Gamma(\alpha_{jd}+\beta_{jd})}{\Gamma(\alpha_{jd})\Gamma(\beta_{jd})}$. As $\beta_{j(D+1)} = 0$, Equ. A.1 becomes :

$$\frac{1}{B(\theta_{jD})} Y_D^{\alpha_{jD}-1}(1+\sum_{l=1}^{D} Y_l)^{-\beta_{jD}-\alpha_{jD}} \prod_{l=1}^{D-1} \frac{1}{B(\theta_{jl})} Y_D^{\alpha_{jl}-1}(1+\sum_{k=1}^{l} Y_k)^{-\beta_{jl}-\alpha_{jl}+\beta_{j(l+1)}} \tag{A.2}$$

by multiplying equation A.2 by the following constant:$(1+\sum_{l=1}^{D-1} Y_l)^{\beta_{jD}+\alpha_{jD}-\alpha_{jD}+1} = (1+\sum_{l=1}^{D-1} Y_l)^{\beta_{jD}+1}$. Equation A.2 becomes proportional to:

$$\begin{aligned} &\frac{1}{B(\theta_{jD})}(\frac{Y_D}{1+\sum_{l=1}^{D-1} Y_l})^{\alpha_{jD}-1}(1+\frac{Y_D}{1+\sum_{l=1}^{D-1} Y_l})^{-\beta_{jD}-\alpha_{jD}} \\ &\quad \times \prod_{l=1}^{D-1} \frac{1}{B(\theta_{jl})} Y_D^{\alpha_{jl}-1}(1+\sum_{k=1}^{l} Y_k)^{-\beta_{jl}-\alpha_{jl}+\beta_{j(l+1)}} \end{aligned} \tag{A.3}$$

We know that:

$$\frac{1}{B(\theta_{jD})}(\frac{Y_D}{1+\sum_{l=1}^{D-1}Y_l})^{\alpha_{jD}-1}(1+\frac{Y_D}{1+\sum_{l=1}^{D-1}Y_l})^{-\beta_{jD}-\alpha_{jD}} = p_{ib}(\frac{Y_D}{1+\sum_{l=1}^{D-1}Y_l}|\theta_{jD}) \tag{A.4}$$

so equation A.2 becomes:

$$p_{ib}(\frac{Y_D}{1+\sum_{l=1}^{D-1}Y_l}|\theta_{jD})\prod_{l=1}^{D-1}\frac{1}{B(\theta_{jl})}Y_D^{\alpha_{jl}-1}(1+\sum_{k=1}^{l}Y_k)^{-\beta_{jl}-\alpha_{jl}+\beta_{j(l+1)}} \tag{A.5}$$

For every remaining feature l in the product from 1 to $D-1$, we multiply Equation A.5 by the constant $(1+\sum_{k=1}^{l-1}Y_k)^{\beta_{jl}+\alpha_{jl}-\alpha_{jl}+1}(1+\sum_{k=1}^{l}Y_k)^{-\beta_{j(l+1)}} = (1+\sum_{k=1}^{l-1}Y_k)^{\beta_{jl}+1}(1+\sum_{k=1}^{l}Y_k)^{-\beta_{j(l+1)}}$. So equation A.5 will be proportional to: $\prod_{l=1}^{D} p_{ib}(\frac{Y_l}{1+\sum_{k=1}^{l-1}Y_k}|\theta_{jl})$, which the first term is $p_{ib}(Y_1|\theta_{jl})$. So, we finally have:

$$p(j|\vec{Y}_i) \propto p_j p(\vec{Y}_i|\vec{\theta}_j) \propto p_j p_{ib}(Y_1|\theta_{jl})\prod_{l=2}^{D}p_{ib}(\frac{Y_l}{1+\sum_{k=1}^{l-1}Y_k}|\theta_{jl}) \tag{A.6}$$

Appendix B

Here, we show how to derive the message length formula by substituting $p(\Theta^{**})$ and $|I(\Theta^{**})|$ into Equation 2.15:

$$
\begin{aligned}
MML(M) = {} & \frac{1}{2}\sum_{j=1}^{M+1}\log p_j + \frac{1}{2}\sum_{l=1}^{D}\big(\log(\rho_{l_1}) + \log(\rho_{l_2})\big) + \frac{c}{2}(1+\log\frac{1}{12}) - \log p(\mathscr{X}|\Theta^{**}) \\
& + \frac{1}{2}\Bigg[M\log N - \sum_{j=1}^{M+1}\log p_j + \sum_{l=1}^{D}(\log N - \log\rho_{l_1} - \log\rho_{l_2})\Bigg] \\
& + \frac{1}{2}\Bigg[\sum_{j=1}^{M}\sum_{l=1}^{D}\Bigg(2\log(Np_j\rho_{l_1}) + \log|\Psi'(\alpha_{jl})\Psi'(\beta_{jl}) - \Psi'(\alpha_{jl}+\beta_{jl})(\Psi'(\alpha_{jl})\Psi'(\beta_{jl}))|)\Bigg) \\
& + \sum_{l=1}^{D}\Bigg(2\log(N\rho_{l_2}) + \log|\Psi'(\alpha_{\lambda|l})\Psi'(\beta_{\lambda|l}) - \Psi'(\alpha_{\lambda|l}+\beta_{\lambda|l})(\Psi'(\alpha_{\lambda|l})\Psi'\beta_{\lambda|l}))|\Bigg)\Bigg] \\
& + \sum_{l=1}^{D}\Bigg(\log\alpha_{\lambda|l} + \log\beta_{\lambda|l} + \log(\hat{A}^{\lambda_l} - \alpha_{\lambda|l} - \beta_{\lambda|l})\Bigg) \\
& + \sum_{j=1}^{M}\sum_{l=1}^{D}\Bigg(\log\alpha_{jl} + \log\beta_{jl} + \log(\hat{A}^{\theta_{jl}} - \alpha_{jl} - \beta_{jl})\Bigg) \\
& = \frac{M+D+2MD+2D}{2}\log N + D\sum_{j=1}^{M}\log p_j + M\sum_{l=1}^{D}\log(\rho_{l_1}) + \sum_{l=1}^{D}\log(\rho_{l_2}) \\
& + \frac{1}{2}\Bigg[\sum_{j=1}^{M}\sum_{l=1}^{D}\Bigg(\log|\Psi'(\alpha_{jl})\Psi'(\beta_{jl}) - \Psi'(\alpha_{jl}+\beta_{jl})(\Psi'(\alpha_{jl})\Psi'(\beta_{jl}))|)\Bigg) \\
& + \sum_{l=1}^{D}\Bigg(\log|\Psi'(\alpha_{\lambda|l})\Psi'(\beta_{\lambda|l}) - \Psi'(\alpha_{\lambda|l}+\beta_{\lambda|l})(\Psi'(\alpha_{\lambda|l})\Psi'\beta_{\lambda|l}))|\Bigg)\Bigg]
\end{aligned}
$$

$$+\sum_{l=1}^{D}\left(\log\alpha_{\lambda|l}+\log\beta_{\lambda|l}+\log(\hat{A}^{\lambda_l}-\alpha_{\lambda|l}-\beta_{\lambda|l})\right)$$
$$+\sum_{j=1}^{M}\sum_{l=1}^{D}\left(\log\alpha_{jl}+\log\beta_{jl}+\log(\hat{A}^{\theta_{jl}}-\alpha_{jl}-\beta_{jl})\right)+\frac{c}{2}(1+\log\frac{1}{12})-\log p(\mathscr{X}|\Theta^{**}) \tag{B.1}$$

Appendix C

In this appendix, we show how to obtain the update formulae for $\rho_{l_1} = \rho_l$ and $\rho_{l_2} = (1-\rho_l)$. By computing the derivative of $L(\Theta^{**}, \mathscr{X})$ w.r.t ρ_{l_1}, we obtain:

$$\begin{aligned}
\frac{\partial L(\Theta^{**}, \mathscr{X})}{\partial \rho_{l_1}} &= \frac{\partial \log p(\mathscr{X}|\Theta^{**})}{\partial \rho_{l_1}} - \frac{M}{\rho_{l_1}} - \Lambda_2 = 0 \\
&= \frac{\partial \sum_{i=1}^{N} \log \left(\sum_{j=1}^{M} p_j \prod_{l=1}^{D} \left[\rho_{l_1} p_{ib}(X_{il}|\theta_{jl}) + \rho_{l_2} p_{ib}(X_{il}|\lambda_l)\right] + p_{M+1} U(\vec{X}_i) \right)}{\partial \rho_{l_1}} \\
&- \frac{M}{\rho_{l_1}} - \Lambda_2 = 0 \\
&= \sum_{i=1}^{N} \sum_{j=1}^{M} p(j|\vec{X}_i) \left(\frac{\partial \left[\rho_{l_1} p_{ib}(X_{il}|\theta_{jl}) + \rho_{l_2} p_{ib}(X_{il}|\lambda_l)\right]}{\partial \rho_{l_1}} \right) - \frac{M}{\rho_{l_1}} - \Lambda_2 = 0 \\
&= \sum_{i=1}^{N} \sum_{j=1}^{M} p(j|\vec{X}_i) \left(\frac{p_{ib}(X_{il}|\theta_{jl})}{\rho_{l_1} p_{ib}(X_{il}|\theta_{jl}) + \rho_{l_2} p_{ib}(X_{il}|\lambda_l)} \right) - \frac{M}{\rho_{l_1}} - \Lambda_2 = 0
\end{aligned} \tag{C.1}$$

Multiplying by ρ_{l_1}, we obtain:

$$\sum_{i=1}^{N} \sum_{j=1}^{M} \rho_{l_1} p(j|\vec{X}_i) \left(\frac{p_{ib}(X_{il}|\theta_{jl})}{\rho_{l_1} p_{ib}(X_{il}|\theta_{jl}) + \rho_{l_2} p_{ib}(X_{il}|\lambda_l)} \right) - M - \Lambda_2 \rho_{l_1} = 0$$

By computing the derivative of $L(\Theta^{**}, \mathscr{X})$ w.r.t ρ_{l_2}, we obtain:

$$\sum_{i=1}^{N} \sum_{j=1}^{M} p(j|\vec{X}_i) \left(\frac{p_{ib}(X_{il}|\lambda_l)}{\rho_{l_1} p_{ib}(X_{il}|\theta_{jl}) + \rho_{l_2} p_{ib}(X_{il}|\lambda_l)} \right) - \frac{1}{\rho_{l_2}} - \Lambda_2 = 0 \tag{C.2}$$

Multiplying by ρ_{l_2}, we obtain:

$$\sum_{i=1}^{N}\sum_{j=1}^{M}\rho_{l_2}p(j|\vec{X}_i)\left(\frac{p_{ib}(X_{il}|\lambda_l)}{\rho_{l_1}p_{ib}(X_{il}|\theta_{jl})+\rho_{l_2}p_{ib}(X_{il}|\lambda_l)}\right)-1-\rho_{l_2}\Lambda_2=0 \tag{C.3}$$

By summing Eqs. C.2 and C.3, we obtain:

$$\sum_{i=1}^{N}\sum_{j=1}^{M}p(j|\vec{X}_i)-M-1=N-M-1=\Lambda_2 \tag{C.4}$$

then, according to Eq. C.2, we have:

$$\rho_{l_1}=\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}p(j|\vec{X}_i)\frac{\rho_{l_1}p_{ib}(X_{il}|\theta_{jl})}{\rho_{l_1}p_{ib}(X_{il}|\theta_{jl})+\rho_{l_2}p_{ib}(X_{il}|\lambda_l)}-M}{N-M-1} \tag{C.5}$$

$$\rho_{l_2}=\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}p(j|\vec{X}_i)\frac{\rho_{l_2}p_{ib}(X_{il}|\lambda_l)}{\rho_{l_1}p_{ib}(X_{il}|\theta_{jl})+\rho_{l_2}p_{ib}(X_{il}|\lambda_l)}-1}{N-M-1} \tag{C.6}$$

It is noteworthy that both previous equations can be summarized as follows. We have $\rho_{l_1}+\rho_{l_2}=1$, thus $\frac{1}{\rho_{l_1}}=1+\frac{\rho_{l_2}}{\rho_{l_1}}$, then since ρ_{l_1} and ρ_{l_2} must be positive, we obtain:

$$\frac{1}{\rho_{l_1}}=1+\frac{\max(\sum_{i=1}^{N}\sum_{j=1}^{M}p(j|\vec{X}_i)\frac{\rho_{l_2}p_{ib}(X_{il}|\lambda_l)}{\rho_{l_1}p_{ib}(X_{il}|\theta_{jl})+\rho_{l_2}p_{ib}(X_{il}|\lambda_l)}-1,0)}{\max(\sum_{i=1}^{N}\sum_{j=1}^{M}p(j|\vec{X}_i)\frac{\rho_{l_1}p_{ib}(X_{il}|\theta_{jl})}{\rho_{l_1}p_{ib}(X_{il}|\theta_{jl})+\rho_{l_2}p_{ib}(X_{il}|\lambda_l)}-M,0)} \tag{C.7}$$

Appendix D

The first derivatives of $L(\Theta^{**}, \mathscr{X})$ w.r.t θ_{jl} are given by:

$$\frac{\partial L(\Theta^{**}, \mathscr{X})}{\partial \alpha_{jl}} = \sum_{i=1}^{N} p(j|\vec{X}_i)\left(\frac{\rho_{l_1}\frac{\partial p_{IBeta}(X_{il}|\theta_{jl})}{\partial \alpha_{jl}}}{\rho_{l_1} p_{IBeta}(X_{il}|\theta_{jl}) + \rho_{l_2} p_{IBeta}(X_{il}|\lambda_l)}\right) \tag{D.1}$$

where $\frac{\partial p_{IBeta}(X_{il}|\theta_{jl})}{\partial \alpha_{jl}} = p_{IBeta}(X_{il}|\theta_{jl})\big[\Psi(\alpha_{jl}+\beta_{jl}) - \Psi(\alpha_{jl}) + \log X_{il} - \log(1+X_{il})\big]$. Moreover,

$$\frac{\partial L(\Theta^{**}, \mathscr{X})}{\partial \beta_{jl}} = \sum_{i=1}^{N} p(j|\vec{X}_i)\left(\frac{\rho_{l_1}\frac{\partial p_{IBeta}(X_{il}|\theta_{jl})}{\partial \beta_{jl}}}{\rho_{l_1} p_{IBeta}(X_{il}|\theta_{jl}) + \rho_{l_2} p_{IBeta}(X_{il}|\lambda_l)}\right) \tag{D.2}$$

where $\frac{\partial p_{IBeta}(X_{il}|\theta_{jl})}{\partial \beta_{jl}} = p_{IBeta}(X_{il}|\theta_{jl})\big[\Psi(\alpha_{jl}+\beta_{jl}) - \Psi(\alpha_{jl}) - \log(1+X_{il})\big]$. The second derivative w.r.t α_{jl} is given by:

$$\begin{aligned}\frac{\partial^2 L(\Theta^{**}, \mathscr{X})}{\partial^2 \alpha_{jl}} = \sum_{i=1}^{N} p(j|\vec{X}_i)\Bigg[&\frac{\rho_{l_1}\frac{\partial^2 p_{IBeta}(X_{il}|\theta_{jl})}{\partial^2 \alpha_{jl}}}{\rho_{l_1} p_{IBeta}(X_{il}|\theta_{jl}) + \rho_{l_2} p_{IBeta}(X_{il}|\lambda_l)} \\ &- \frac{\big(\rho_{l_1}\frac{\partial p_{IBeta}(X_{il}|\theta_{jl})}{\partial \alpha_{jl}}\big)^2}{\big(\rho_{l_1} p_{IBeta}(X_{il}|\theta_{jl}) + \rho_{l_2} p_{IBeta}(X_{il}|\lambda_l)\big)^2}\Bigg]\end{aligned} \tag{D.3}$$

where:

$$\begin{aligned}\frac{\partial^2 p_{IBeta}(X_{il}|\theta_{jl})}{\partial^2 \alpha_{jl}} = &\frac{\partial p_{IBeta}(X_{il}|\theta_{jl})}{\partial \alpha_{jl}}\big[\Psi(\alpha_{jl}+\beta_{jl}) - \Psi(\alpha_{jl}) - \log(1+X_{il})\big] \\ &+ p_{IBeta}(X_{il}|\theta_{jl})\big[\Psi'(\alpha_{jl}+\beta_{jl}) - \Psi'(\alpha_{jl})\big]\end{aligned} \tag{D.4}$$

The same development can be straightforwardly followed for the second derivative w.r.t β_{jl}. As for the mixed derivative, it is given by the following:

$$\begin{aligned}\frac{\partial^2 L(\Theta^{**},\mathscr{X})}{\partial\alpha_{jl}\partial\beta_{jl}} &= \frac{\partial^2 L(\Theta^{**},\mathscr{X})}{\partial\beta_{jl}\partial\alpha_{jl}} \\ &= \sum_{i=1}^{N}\rho_{l_1}p(j|\vec{X}_i)\Bigg[\frac{\frac{\partial^2 p_{IBeta}(X_{il}|\theta_{jl})}{\partial\alpha_{jl}\beta_{jl}}}{\rho_{l_1}p_{IBeta}(X_{il}|\theta_{jl})+\rho_{l_2}p_{IBeta}(X_{il}|\lambda_l)} \\ &\quad - \frac{\rho_{l_1}\frac{\partial p_{IBeta}(X_{il}|\theta_{jl})}{\partial\alpha_{jl}}\frac{\partial p_{IBeta}(X_{il}|\theta_{jl})}{\partial\beta_{jl}}}{\big(\rho_{l_1}p_{IBeta}(X_{il}|\theta_{jl})+\rho_{l_2}p_{IBeta}(X_{il}|\lambda_l)\big)^2}\Bigg]\end{aligned} \tag{D.5}$$

where

$$\begin{aligned}\frac{\partial^2 p_{IBeta}(X_{il}|\theta_{jl})}{\partial\alpha_{jl}\partial\beta_{jl}} &= \frac{\partial p_{IBeta}(X_{il}|\theta_{jl})}{\partial\beta_{jl}}\big[\Psi(\alpha_{jl}+\beta_{jl}) \\ &\quad -\Psi(\alpha_{jl})-\log(1+X_{il})\big]+p_{IBeta}(X_{il}|\theta_{jl})\big[\Psi(\alpha_{jl}+\beta_{jl})\big]\end{aligned} \tag{D.6}$$

Similarly, we can obtain the derivatives w.r.t λ_l.

Appendix E

$$
\begin{aligned}
p(\vec{X}|\vec{\xi}_k) &= \prod_{d=1}^{D} \frac{\Gamma(\alpha_{kd}+\beta_{kd})}{\Gamma(\alpha_{kd})\Gamma(\beta_{kd})} \frac{X_d^{\alpha_{kd}-1}}{(1+\sum_{l=1}^{d} X_l)^{\gamma_{kd}}} = \exp\Bigg[\sum_{d=1}^{D} \log\big(\Gamma(\alpha_{kd}+\beta_{kd})\big) - \log\big(\Gamma(\alpha_{kd})\big) \\
&- \log\big(\Gamma(\beta_{kd})\big) + (\alpha_{kd}-1)\log(X_d) - \gamma_{kd}\log(1+\sum_{l=1}^{d} X_l)\Bigg] \\
&= \exp\Bigg[\sum_{d=1}^{D}\Bigg(\log\big(\Gamma(\alpha_{kd}+\beta_{kd})\big) - \log\big(\Gamma(\alpha_{kd})\big) - \log\big(\Gamma(\beta_{kd})\big)\Bigg) \\
&+ \sum_{d=1}^{D}\Bigg((\alpha_{kd}-1)\log(X_d) - \gamma_{kd}\log(1+\sum_{l=1}^{d} X_l)\Bigg)\Bigg] \\
&= \exp\Bigg[\sum_{d=1}^{D}\Bigg(\log\big(\Gamma(\alpha_{kd}+\beta_{kd})\big) - \log\big(\Gamma(\alpha_{kd})\big) - \log\big(\Gamma(\beta_{kd})\big)\Bigg) \\
&+ \sum_{d=1}^{D}\Bigg((\alpha_{kd}-1)\log(X_d)\Bigg) - \sum_{d=D+1}^{2D-1}\Bigg((\beta_{kd-D}+\alpha_{kd-D}-\beta_{kd-D+1})\log(1+\sum_{l=1}^{d-l} X_l)\Bigg) \\
&- (\alpha_{kD}+\beta_{kD})\log(1+\sum_{l=1}^{D} X_l)\Bigg]
\end{aligned}
\tag{E.1}
$$

Appendix F

$$
\begin{aligned}
p(\xi_k|\mathscr{Z},\mathscr{X}) &\propto p(\xi_k)\prod_{Z_{ik}=1} p(\vec{X}_i|\xi_k) \\
&\propto \exp\Big[\kappa \sum_{d=1}^{D} \Big(\log\big(\Gamma(\alpha_{kd}+\beta_{kd})\big) - \log\big(\Gamma(\alpha_{kd})\big) - \log\big(\Gamma(\beta_{kd})\big)\Big) + \sum_{d=1}^{D} \rho_d \alpha_{kd} \\
&+ \sum_{d=1}^{D} \rho_{d+D}\gamma_{kd}\Big] \times \Big(\prod_{d=1}^{D} \frac{\Gamma(\alpha_{kd}+\beta_{kd})}{\Gamma(\alpha_{kd})\Gamma(\beta_{kd})}\Big)^{n_k} \prod_{Z_{ik}=1}\Big(\prod_{d=1}^{D} \frac{X_{id}^{\alpha_{kd}-1}}{(1+\sum_{l=1}^{d} X_{il})^{\gamma_{kd}}}\Big) \\
&\propto \exp\Big[\sum_{d=1}^{D} \rho_d \alpha_{kd} + \sum_{d=1}^{D} \rho_{d+D}\gamma_{kd} + \kappa \sum_{d=1}^{D} \Big(\log\big(\Gamma(\alpha_{kd}+\beta_{kd})\big) \\
&- \log\big(\Gamma(\alpha_{kd})\big) - \log\big(\Gamma(\beta_{kd})\big)\Big)\Big] \times \exp\Big[n_k \sum_{d=1}^{D} \Big(\log\big(\Gamma(\alpha_{kd}+\beta_{kd})\big) \\
&- \log\big(\Gamma(\alpha_{kd})\big) - \log\big(\Gamma(\beta_{kd})\big)\Big) + \sum_{Z_{ik}=1} \Big[\sum_{d=1}^{D} \Big((\alpha_{kd}-1)\log(X_{id})\Big) \\
&- \sum_{d=1}^{D} \Big(\gamma_{kd}\log(1+\sum_{l=1}^{d} X_{il})\Big)\Big]\Big] \\
&\propto \exp\Big[\sum_{d=1}^{D} \alpha_{kd}\Big(\rho_d + \sum_{Z_{ik}=1} \log(X_{id})\Big) + \sum_{d=1}^{D} \gamma_{kd}\Big(\rho_{d+D} - \sum_{Z_{ik}=1} \log(1+\sum_{l=1}^{d} X_{il})\Big) \\
&+ (\kappa+n_k)\sum_{d=1}^{D} \Big(\log\big(\Gamma(\alpha_{kd}+\beta_{kd})\big) - \log\big(\Gamma(\alpha_{kd})\big) - \log\big(\Gamma(\beta_{kd})\big)\Big)\Big]
\end{aligned}
\tag{F.1}
$$