

High-Dimensional Non-Gaussian Data Clustering using Variational Learning of Mixture Models

Wentao Fan

A Thesis

in

Department of Electrical and Computer Engineering

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

December 2013

© **Wentao Fan, 2013**

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Wentao Fan**

Entitled: **High-Dimensional Non-Gaussian Data Clustering using Variational Learning of Mixture Models**

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

DOCTOR OF PHILOSOPHY (Electrical & Computer Engineering)

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

Signed by the final examining committee:

	_____	Chair
	Dr. F. Haghigat	
Examiner	_____	External
	Dr. G.-A. Bilodeau	
Program	_____	External to
	Dr. P. Grogono	
	_____	Examiner
	Dr. A. Ben Hamza	
	_____	Examiner
	Dr. A. Youssef	
	_____	Thesis Supervisor
	Dr. N. Bouguila	

Approved by _____
Chair of Department or Graduate Program Director
Dr. A.R. Sebak, Graduate Program Director

December 11, 2013

Dr. C. Trueman, Interim Dean
Faculty of Engineering & Computer Science

Abstract

High-Dimensional Non-Gaussian Data Clustering using Variational Learning of Mixture Models

Wentao Fan, Ph.D.

Concordia University, 2013

Clustering has been the topic of extensive research in the past. The main concern is to automatically divide a given data set into different clusters such that vectors of the same cluster are as similar as possible and vectors of different clusters are as different as possible. Finite mixture models have been widely used for clustering since they have the advantages of being able to integrate prior knowledge about the data and to address the problem of unsupervised learning in a formal way. A crucial starting point when adopting mixture models is the choice of the components densities. In this context, the well-known Gaussian distribution has been widely used. However, the deployment of the Gaussian mixture implies implicitly clustering based on the minimization of Euclidean distortions which may yield to poor results in several real applications where the per-components densities are not Gaussian. Recent works have shown that other models such as the Dirichlet, generalized Dirichlet and Beta-Liouville mixtures may provide better clustering results in applications containing non-Gaussian data, especially those involving proportional data (or normalized histograms) which are naturally generated by many applications. Two other challenging aspects that should also be addressed when considering mixture models are: how to determine the model's complexity (i.e. the number of mixture components) and how to estimate the model's parameters. Fortunately, both problems can be tackled simultaneously within a principled elegant learning framework namely variational inference. The main idea of variational inference is to approximate the model posterior distribution by minimizing the Kullback-Leibler divergence between the exact (or true) posterior and an approximating distribution. Recently, variational inference has provided good generalization performance and computational tractability in many applications including learning mixture models.

In this thesis, we propose several approaches for high-dimensional non-Gaussian data clustering based on various mixture models such as Dirichlet, generalized Dirichlet and Beta-Liouville. These mixture models are learned using variational inference which main advantages are computational efficiency and guaranteed convergence. More specifically, our contributions are four-fold. Firstly, we develop a variational inference algorithm for learning the finite Dirichlet mixture model, where model parameters and the model complexity can be determined automatically and simultaneously as part of the Bayesian inference procedure; Secondly, an unsupervised feature selection scheme is integrated with finite generalized Dirichlet mixture model for clustering high-dimensional non-Gaussian data; Thirdly, we extend the proposed finite generalized mixture model to the infinite case using a nonparametric Bayesian framework known as Dirichlet process, so that the difficulty of choosing the appropriate number of clusters is sidestepped by assuming that there are an infinite number of mixture components; Finally, we propose an online learning framework to learn a Dirichlet process mixture of Beta-Liouville distributions (i.e. an infinite Beta-Liouville mixture model), which is more suitable when dealing with sequential or large scale data in contrast to batch learning algorithm. The effectiveness of our approaches is evaluated using both synthetic and real-life challenging applications such as image databases categorization, anomaly intrusion detection, human action videos categorization, image annotation, facial expression recognition, behavior recognition, and dynamic textures clustering.

Acknowledgements

First, I would like to express my greatest gratitude to my supervisor, Dr. Nizar Bouguila for opening the door of the academic world to me. Within six years, from my Master to Ph.D. study, he has always been a wonderful advisor, mentor and motivator. I learned a lot from his valuable tutoring, not only technical knowledge but also about dealing in real life. I will be always grateful to him for his support and persistent encouragement.

Special thanks goes to Dr. A. Ben Hamza for his patience and guidance through my INSE6510 project. I also benefit a lot from being his teaching assistant for the course Comp471.

It is a great pleasure to work with the former and current colleagues in our lab. I would like to thank them for their helpful suggestions and discussions during my research. I am lucky to share six years' time with them.

I would like to thank Fonds de recherche du Québec-Nature et technologies (FQRNT) for the scholarship at the doctorate-level. I am also grateful to the Faculty of Engineering and Computer Science of Concordia University for the Special Scholarship for New High Caliber Ph.D. Students. In addition, I would like to express my gratitude to the Chinese government for the Chinese Government Award for Outstanding Self-financed Students Abroad.

Last, but by no means least I would like to thank my family for their love and support in all my years, and especially my wife for her love, encouragement and endless patience with me.

Table of Contents

List of Tables	ix
List of Figures	xi
1 Introduction	1
1.1 Clustering via Finite Mixture Models	1
1.2 Variational Inference	4
1.3 Contributions	7
1.4 Thesis Overview	8
2 Variational Learning for Finite Dirichlet Mixture Models	10
2.1 The Finite Dirichlet Mixture Model	10
2.2 Variational Inference for Finite Dirichlet Mixture Model	12
2.2.1 Variational Approximation	12
2.2.2 Determining The Number of Components	15
2.2.3 Complete Variational Learning Algorithm	16
2.3 Experimental Results	17
2.3.1 Synthetic Data	19
2.3.2 Images Categorization	21
2.3.3 Anomaly Intrusion Detection	24
3 Unsupervised Feature Selection for High-Dimensional Non-Gaussian Data Clustering with Variational Inference	27
3.1 Model specification	27
3.2 Variational Learning of the Model	30
3.3 Experimental Results	33
3.3.1 Synthetic Data	34
3.3.2 Human Action Videos Categorization	35
4 Variational Learning of a Dirichlet Process of Generalized Dirichlet Distributions for Simultaneous Clustering and Feature Selection	42
4.1 The Infinite GD Mixture Model with Feature Selection	42
4.1.1 The Finite GD Mixture Model	43

4.1.2	Infinite GD Mixture Model With Feature Selection	44
4.1.3	Prior Distributions of The Proposed Model	46
4.2	Variational Inference	49
4.3	Experimental Results	52
4.3.1	Synthetic data	54
4.3.2	Visual Scenes Categorization	54
4.3.3	Image Auto-Annotation	59
5	Online Learning of a Dirichlet Process Mixture of Beta-Liouville Distributions via Variational Inference	66
5.1	Beta-Liouville Mixture Model	67
5.1.1	Finite Beta-Liouville Mixture Model	67
5.1.2	Infinite Beta-Liouville Mixture Model	68
5.2	Online Variational Model Learning	69
5.2.1	Batch Variational Learning	69
5.2.2	Online Variational Inference	72
5.3	Experimental Results	76
5.3.1	Design of Experiments	76
5.3.2	Facial Expression Recognition	76
5.3.3	Behavior Modeling and Recognition	79
5.3.4	Dynamic Textures Clustering	81
6	Conclusions	86
	List of References	90
A	Proof of Equations (2.14) and (2.15)	109
A.1	Proof of Equation (2.14): Variational Solution to $Q(\mathcal{Z})$	109
A.2	Proof of Equation (2.15): Variational Solution to $Q(\vec{\alpha})$	110
B	Proof of Equations (2.18) and (A.12)	113
B.1	Lower Bound of \mathcal{R}_j : Proof of Equation (2.18)	113
B.2	Lower Bound of $\mathcal{J}(\alpha_{js})$: Proof of Equation (A.12)	115
B.2.1	Convexity of $\mathcal{F}(\alpha_{js})$	115
B.2.2	Evaluating Lower Bound by The First Order Taylor Expansion	116
C	Variational Learning of Online Infinite Beta-Liouville Mixture	118
C.1	Variational lower bound $\mathcal{L}(Q)$	118
C.2	Variational solution to $Q(\mathcal{Z})$	118
C.3	Variational solution to $Q(\vec{\lambda})$	120

C.4 Variational solutions to $Q(\vec{\alpha}_d)$, $Q(\vec{\alpha})$ and $Q(\vec{\beta})$ 120

List of Tables

2.1	Parameters of the different generated data sets. N denotes the total number of elements, n_j denotes the number of elements in cluster j . $\alpha_{j1}, \alpha_{j2}, \alpha_{j3}$ and π_j are the real parameters. $\hat{\alpha}_{j1}, \hat{\alpha}_{j2}, \hat{\alpha}_{j3}$ and $\hat{\pi}_j$ are the estimated parameters by variational inference. $\check{\alpha}_{j1}, \check{\alpha}_{j2}, \check{\alpha}_{j3}$ and $\check{\pi}_j$ are the estimated parameters using DM. We can observe that both algorithms are able to estimate unknown parameters, yet the variational algorithm always gives more accurate values.	18
2.2	Rum time (in seconds) and number of iterations required before convergence for varDM and DM.	22
2.3	Clustering Accuracies with varDM Model and varGM Model. M^* denotes the average number of clusters.	22
2.4	Average Rounded Confusion Matrix using the varDM Model to categorize Data Set A.	24
2.5	Average Rounded Confusion Matrix using the varDM Model to categorize Data Set B.	25
2.6	Confusion Matrix for Intrusion Detection with Variational Dirichlet Mixture Model.	26
2.7	Intrusion Detection Results Using different approaches.	26
3.1	Parameters of the different generated data sets. N denotes the total number of elements, n_j denotes the number of elements in cluster j for the relevant features. $\alpha_{j1}, \beta_{j1}, \alpha_{j2}, \beta_{j2}, \alpha_{j3}, \beta_{j3}$ and π_j are the real parameters of the mixture models of relevant features. $\hat{\alpha}_{j1}, \hat{\beta}_{j1}, \hat{\alpha}_{j2}, \hat{\beta}_{j2}, \hat{\alpha}_{j3}, \hat{\beta}_{j3}$ and $\hat{\pi}_j$ are the estimated parameters from variational inference.	34
3.2	The average classification accuracy and the number of components (\hat{M}) computed on the KTH data set using varFsGD, MMLFsGD, varGD and varFsGau over 30 random runs.	39
4.1	Parameters of the generated data sets. N denotes the total number of elements, N_j denotes the number of elements in cluster j . $\alpha_{j1}, \alpha_{j2}, \beta_{j1}, \beta_{j2}$ and π_j are the real parameters. $\hat{\alpha}_{j1}, \hat{\alpha}_{j2}, \hat{\beta}_{j1}, \hat{\beta}_{j2}$ and $\hat{\pi}_j$ are the estimated parameters by the proposed algorithm.	53
4.2	The average classification accuracy and the number of categories (\hat{M}) computed by different algorithms for the Caltech data set.	59
4.3	The average classification accuracy computed by different algorithms.	62

4.4	Performance evaluation on the automatic annotation system based on different categorization methods.	63
4.5	Sample annotation results by using <i>InFsGD</i> classification method.	64
4.6	The comparison of image retrieval performance.	64
5.1	The average recognition accuracy (%) and the number of categories (\widehat{M}) computed by different algorithms for the JAFFE data set. The numbers in parenthesis are the standard deviations of the corresponding quantities.	78
5.2	The average recognition accuracy rate (Acc) and the average estimated number of categories (\widehat{M}) computed using different algorithms on the three data sets: facial expression (face), mouse behavior (mouse) and human activity (UCF11).	79
5.3	The average accuracy and the number of categories (\widehat{M}) computed by different algorithms when clustering the DynTex data set.	83

List of Figures

2.1	Graphical model representation of the finite Dirichlet mixture. Symbols in circles denote random variables; otherwise, they denote model parameters. Plates indicate repetition (with the number of repetitions in the lower right), and arcs describe conditional dependencies between variables.	13
2.2	Mixture densities for the synthetic data sets. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4, (e) Data set 5, (f) Data set 6.	19
2.3	Variational likelihood bound for each iteration for the different generated data sets. The initial number of components is 15. Vertical dash lines indicate cancelation of components. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4, (e) Data set 5, (f) Data set 6.	20
2.4	Variational likelihood bound as a function of the fixed assumed number of mixture components for the different generated data sets. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4, (e) Data set 5, (f) Data set 6.	21
2.5	Sample images from each group of sports event data set: (a) Rowing. (b) Badminton. (c) polo. (d) Bocce. (e) Snow Boarding. (f) Croquet. (g) Sailing. (h) Rock climbing.	23
3.1	Feature saliency for synthetic data sets with one standard deviation over ten runs. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4.	35
3.2	Examples of frames, representing different human actions in different scenarios, from video sequences in the KTH data set.	36
3.3	Confusion matrix for the KTH data set.	39
3.4	(a) Classification accuracy vs. vocabulary size for the KTH data set; (b) Classification accuracy vs. the number of aspects for the KTH data set.	40
3.5	Feature saliencies of the different aspect features over 30 runs for the KTH data set.	41
4.1	Graphical model representation of the infinite GD mixture model with feature selection. Symbols in circles denote random variables; otherwise, they denote model parameters. Plates indicate repetition (with the number of repetitions in the lower right), and arcs describe conditional dependencies between variables.	48
4.2	Mixing probabilities of components, π_j , found for each synthetic data set after convergence. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4.	55
4.3	Features saliencies for synthetic data sets with one standard deviation over ten runs. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4.	56

4.4	Sample images from the four categories of the Caltech data set.	57
4.5	Sample segmentation results from the four categories of the Caltech data sets . . .	58
4.6	(a) Classification accuracy vs. the number of aspects; (b) Feature saliency for each aspect.	59
5.1	Sample images from the JAFFE data set: (a) Anger, (b) Disgust, (c) Fear, (d) Happiness, (e) Sadness, (f) Surprise, (g) Neutral.	77
5.2	Confusion matrix obtained by <i>OIBLM</i> for the JAFFE data set.	79
5.3	Sample frames from the each data set. (a): facial expression; (b): mouse behavior; (c): human action.	80
5.4	Performance comparison on the three data sets: facial expression, mouse behavior and human activity using different algorithms.	82
5.5	Sample frames from the DynTex data set.	83
5.6	Confusion matrix obtained by <i>OIBLM</i> for the DynTex data set.	84
5.7	Performance comparison in terms of classification accuracy provided different algorithms for the DynTex data set.	85

Chapter 1

Introduction

1.1 Clustering via Finite Mixture Models

Data clustering is the unsupervised partitioning of data into homogeneous components. It is an important problem in several fields, such as signal and image processing, and has been the topic of extensive research in the past [1–4]. There are a myriad of clustering methods (see [5] for a review). Among all these methods, finite mixture models have been shown to provide flexibility for data clustering [6] and have been successfully applied in several domains and applications. Examples include cognitive understanding [7], epidemiological studies [8], speaker’s location detection [9], person authentication [10], and so forth. Indeed, they have been proven to be a powerful way to capture hidden structure in data and to take uncertainty into account.

A finite mixture model is formed by taking linear combinations of a finite number of basic distributions. These basic distributions are called components of the mixture model. For instance, a finite mixture model with M components is given by

$$p(X) = \sum_{j=1}^M \pi_j p(X|\theta_j) \quad (1.1)$$

where $p(X|\theta_j)$ is a component of the mixture and has its own parameter θ_j . In general, mixture models can comprise linear combinations of any distributions, such as Gaussian, Beta, Dirichlet, etc. The parameters $\{\pi_j\}$ are called mixing coefficients and are subject to the constraints: $0 \leq \pi_j \leq 1$ and $\sum_{j=1}^M \pi_j = 1$. In mixture modeling, three challenging aspects should be carefully addressed: how to choose the proper basic distribution, how to estimate the model’s parameters and how to select the model’s complexity. Each of these aspects has a significant impact on the performance of model learning.

Selecting the most accurate probability density functions (pdfs) that best represent the mixture components is important when modeling and clustering data. The Gaussian assumption has been widely adopted (i.e. assuming that each per-class density is Gaussian) due to its simplicity. In several real-world applications, however, when the data clearly appear with a non-Gaussian structure, this assumption fails. For instance, recent works have shown that other models such as the Dirichlet [11–16], the generalized Dirichlet [17–23] and the Beta-Liouville mixtures [24–27] provide better clustering results in several applications, especially those involving normalized count data (i.e. proportional vectors) which naturally appear in many applications such as text, image and video modeling.

The majority of parameter estimation approaches in mixture modeling consider either deterministic or Bayesian techniques [6]. Deterministic techniques aim at optimizing the model likelihood function, are generally implemented within the expectation-maximization (EM) [28] framework, and are well documented [29, 30]. On the other hand, Bayesian techniques have been proposed to avoid drawbacks related to deterministic techniques such as their suboptimal generalization performance, dependency on initialization, over-fitting and noise level under-estimation problems of classic likelihood-based inference [31, 32]. These drawbacks are avoided via the incorporation of prior knowledge (or belief) in a principled way and then marginalizing over parameter uncertainty. Bayesian methods [33, 34] have considered either Laplace’s approximation [35] or *Markov chain Monte Carlo* (MCMC) simulation techniques [36, 37]. While MCMC techniques are computationally expensive, Laplace’s approximation is generally imprecise, since it is based on the strong assumption that the likelihood function is unimodal which is not generally the case for finite mixtures of distributions [38]. Recently, *Variational inference* (also known as *variational Bayes*) [39, 40] framework has been widely used as an efficient alternative and as a more controllable way to approximate Bayesian learning. The variational learning approach was introduced in the context of the multi-layer perceptron in [41] where it was called ensemble learning and developed further in [42, 43]. The main idea is to approximate the model posterior distribution by minimizing the Kullback-Leibler divergence between the exact (or true) posterior and an approximating distribution. The variational inference has received a lot of attention and has provided good generalization performance and computational tractability in various applications including finite

mixture learning [44–46]. For instance, the authors in [39, 40, 47, 48] have developed comprehensive frameworks for variational learning, in the case of Gaussian mixture models, which have been shown to provide better parameter estimates than the *maximum likelihood* (ML) approach.

Another crucial issue when using mixture models is the model complexity (i.e. model structure or number of mixture components) determination problem. Indeed, it is important to estimate the number of clusters that best describes the data without over-fitting or under-fitting it [6]. In general, this problem is tackled using ML method in conjunction with a given model selection criterion, such as minimum description length (MDL) and minimum message length (MML) [6, 11], in frequentist frameworks or by considering Bayes factors in the case of fully Bayesian approaches. However, these approaches are clearly time-consuming since they have to evaluate a given selection criterion for several numbers of mixture components. This is especially true in the case of the Bayesian approach because it requests the evaluation of multi-dimensional integrals which is generally tackled via MCMC techniques (e.g. Gibbs sampling, Metropolis Hastings). Despite the fact that MCMC techniques have revolutionized Bayesian statistics by accommodating situations characterized by uncertainty of the statistical model structure [49–51], their use is often limited to small-scale problems in practice because of its high computational cost and the difficulty in tracking convergence. Apart from the elegant way to estimate the parameters of mixture models, another advantage of variational inference is that it is able to automatically determine the number of mixture components as part of the Bayesian inference procedure.

Data clustering is known to be a challenging task in modern knowledge discovery and data mining. This is especially true in high-dimensional spaces mainly because of data sparsity. Thus, feature selection is a crucial factor to improve the clustering performance [52–54]. Its primary objective is the identification and the reduction of the influence of extraneous (or irrelevant) features which do not contribute information about the true clusters structure. The automatic selection of relevant features in the context of unsupervised learning is challenging and is far from trivial because inference has to be made on both the selected features and the clustering structure [54–61]. [54] is an early influential paper advocating the use of finite mixture models for unsupervised feature selection. The main idea is to suppose that a given feature is generated from a mixture of two univariate distributions. The first one is assumed to generate relevant features and is different

for each cluster and the second one is common to all clusters (i.e. independent from class labels) and assumed to generate irrelevant features ¹. The unsupervised feature selection models in [54, 60] have been trained using a MML objective function with the EM algorithm. Despite the fact that the EM algorithm is the procedure of choice for parameter estimation in the case of incomplete data problems where part of the data is hidden, several studies have shown theoretically and experimentally that the EM algorithm, in deterministic settings (e.g. ML estimation), converges either to a local maximum or to a saddle point solution and depends on an appropriate initialization (see, for instance, [29, 63, 64]) which may compromise the modeling capabilities. Recently, variational inference have shown promising results in learning mixture models with integrated unsupervised feature selection [57, 65], by providing parameters estimation and features selection in a single optimization framework.

1.2 Variational Inference

In this section, a brief introduction to variational inference is presented. Assume that we have a fully Bayesian model in which all parameters are given proper prior distributions. Let Θ represents the set of all non-observed variables (including latent variables) and \mathcal{X} denotes the set of observations. The goal of variational inference is to find a proper approximation $q(\Theta)$ for the true posterior distribution $p(\Theta|\mathcal{X})$. In order to do this, we can write the following decomposition of the log marginal probability of the observed data \mathcal{X} , which holds for any choice of distribution $q(\Theta)$

$$\ln p(\mathcal{X}) = \mathcal{L}(q) + \mathbf{KL}(q||p) \quad (1.2)$$

where

$$\mathcal{L}(q) = \int q(\Theta) \ln \frac{p(\Theta, \mathcal{X})}{q(\Theta)} d\Theta \quad (1.3)$$

$$\mathbf{KL}(q||p) = - \int q(\Theta) \ln \frac{p(\Theta|\mathcal{X})}{q(\Theta)} d\Theta \quad (1.4)$$

here, $\mathbf{KL}(q||p)$ is the Kullback-Leibler (KL) divergence which represents the dissimilarity between the true posterior $p(\Theta|\mathcal{X})$ and the variational approximation $q(\Theta)$. We know that $\mathbf{KL}(q||p) \geq 0$

¹Several other quantitative formalisms for relevance in the case of feature selection have been proposed in the past (see, for instance, [62]).

(according to Jensen’s inequality), and that the equality is achieved when if and only if $q(\Theta) = p(\Theta|\mathcal{X})$. Then, we can conclude that $\mathcal{L}(q) \leq \ln p(\mathcal{X})$ from Eq. (1.2), which means that $\mathcal{L}(q)$ forms a lower bound on $\ln p(\mathcal{X})$.

Suppose that we allow any possible choice for $q(\Theta)$. Then, the lower bound of $\ln p(\mathcal{X})$ can be maximized with respect to $q(\Theta)$ when the KL divergence is minimized, that is when $q(\Theta) = p(\Theta|\mathcal{X})$. However, in practice the true posterior distribution is normally computationally intractable and can not be directly used for variational inference. Thus, a restricted family of distributions $q(\Theta)$ needs to be considered. An ideal restriction should have the property that, the family of $q(\Theta)$ comprises only tractable distributions, and at meanwhile is still flexible enough to provide a good approximation to the true posterior distribution. A common approach in variational inference literatures is to adopt factorization assumptions for restricting the form of $q(\Theta)$ [66]. This approximation framework is known as *mean field theory* [67, 68] which was developed in the filed of physics [69]. With the factorization assumption, the posterior distribution $q(\Theta)$ can be factorized into T disjoint tractable distributions as

$$q(\Theta) = \prod_{i=1}^T q_i(\Theta_i) \tag{1.5}$$

Notice that this is the only assumption about the distribution, and no further restriction is placed on the functional forms of the individual factors $q_i(\Theta_i)$. In order to maximize the lower bound $\mathcal{L}(q)$, we need to make a variational optimization of $\mathcal{L}(q)$ with respect to each of the distributions $q_i(\Theta_i)$ in turn. Let us substitute Eq. (1.5) into Eq. (1.3), and use q_i to denote $q_i(\Theta_i)$ for simplification,

then the optimization of $\mathcal{L}(q)$ with respect to a specific factor $q_s(\Theta_s)$ can be given as

$$\begin{aligned}
\mathcal{L}(q) &= \int \prod_{i=1}^T q_i \ln \left[\frac{p(\Theta, \mathcal{X})}{\prod_{i=1}^T q_i} \right] d\Theta \\
&= \int q_s \prod_{i \neq s}^T q_i \left[\ln p(\Theta, \mathcal{X}) - \sum_{i=1}^T \ln q_i \right] d\Theta \\
&= \int q_s \left[\int \prod_{i \neq s}^T q_i \ln p(\Theta, \mathcal{X}) d\Theta_i \right] d\Theta_s - \int q_s \ln q_s d\Theta_s + \text{const.} \\
&= \int q_s \ln f(\Theta_s, \mathcal{X}) d\Theta_s - \int q_s \ln q_s d\Theta_s + \text{const.}
\end{aligned} \tag{1.6}$$

where any terms that are independent of $q_s(\Theta_s)$ are absorbed into the additive constant. A new distribution $f(\Theta_s, \mathcal{X})$ in Eq. (1.6) is introduced as

$$\ln f(\Theta_s, \mathcal{X}) = \int \prod_{i \neq s}^T q_i \ln p(\Theta, \mathcal{X}) d\Theta_i = \langle \ln p(\Theta, \mathcal{X}) \rangle_{i \neq s} \tag{1.7}$$

Here, we use the notation $\langle \dots \rangle_{i \neq s}$ to represent the expectation with respect to all the distributions of $q_i(\Theta_i)$ except for $i = s$. We can also notice that Eq. (1.6) is actually a minus KL divergence between $q_s(\Theta_s)$ and $f(\Theta_s, \mathcal{X})$. Therefore, maximizing $\mathcal{L}(q)$ in Eq. (1.6) is equivalent to minimizing the KL divergence. We know that the KL divergence reaches its minimum when $q_s(\Theta_s) = f(\Theta_s, \mathcal{X})$. Thus, a general expression for the optimal solution $q_s^*(\Theta_s)$ can be given by

$$\ln q_s^*(\Theta_s) = \langle \ln p(\mathcal{X}, \Theta) \rangle_{i \neq s} + \text{const.} \tag{1.8}$$

Here, the additive constant denotes the normalization coefficient for the distribution. By taking the exponential of both sides of Eq. (1.8) and normalize, we can obtain the variational solution of $q_s^*(\Theta_s)$ as

$$q_s^*(\Theta_s) = \frac{\exp(\langle \ln p(\mathcal{X}, \Theta) \rangle_{i \neq s})}{\int \exp(\langle \ln p(\mathcal{X}, \Theta) \rangle_{i \neq s}) d\Theta} \tag{1.9}$$

Since the expression for $q_s^*(\Theta_s)$ depends on calculating the expectations with respect to the other factors $q_i^*(\Theta_i)$ for $i \neq s$, we need to cycle through all the factors to find the maximum of the

lower bound. In general, in order to perform the variational inference, all the factors $q_i(\Theta_i)$ need to be suitably initialized first, then each factor is updated in turn with a revised value obtained by Eq. (1.9) using the current values of all of the other factors. Convergence is guaranteed since bound is convex with respect to each of the factors $q_i(\Theta_i)$ [66, 70].

1.3 Contributions

The goal of this thesis is to propose several novel approaches for high-dimensional non-Gaussian data clustering based on variational inference framework in the context of various mixture models including Dirichlet, generalized Dirichlet and Beta-Liouville. The contributions of this thesis are listed as the following:

☞ **Finite Dirichlet Mixture Models with Variational Bayes Learning:**

We propose a variational inference framework for learning finite Dirichlet mixture models. Compared with other algorithms which are commonly used for mixture models (such as EM), our approach has several advantages: first, the problem of over-fitting is prevented; furthermore, the complexity of the mixture model (i.e. the number of components) can be determined automatically and simultaneously with the parameters estimation as part of the Bayesian inference procedure; finally, since the whole inference process is analytically tractable with closed-form solutions, it may scale well to large applications.

☞ **Finite Generalized Dirichlet Mixture Models with Unsupervised Feature Selection:**

A variational inference framework is developed for unsupervised non-Gaussian feature selection, in the context of finite generalized Dirichlet mixture-based clustering. Under the proposed principled variational framework, we simultaneously estimate, in a closed-form, all the involved parameters and determine the complexity (i.e. both model and feature selection) of the finite generalized Dirichlet mixture model.

☞ **Infinite Generalized Dirichlet Mixture Models via Dirichlet Process:**

We extend the finite generalized Dirichlet mixture model to an infinite case through a non-parametric Bayesian framework namely Dirichlet process. The infinite assumption is used

to avoid problems related to model selection (i.e. determination of the number of clusters) and allows simultaneous separation of data in to similar clusters and selection of relevant features.

☞ **Online Learning of Infinite Beta-Liouville Mixture Models:**

We propose a novel online clustering approach based on a Dirichlet process mixture of Beta-Liouville distributions (i.e. an infinite Beta-Liouville mixture model). We are mainly motivated by the fact that online algorithms allow data instances to be processed in a sequential way, which is important for large-scale and real-time applications.

1.4 Thesis Overview

The organization of this thesis is as follows:

- ❑ Chapter 1 introduced the background knowledge regarding finite mixture models and variational inference learning framework.
- ❑ In Chapter 2, we propose a variational inference framework approach to learn finite Dirichlet mixture models. Both synthetic and real data, generated from real-life challenging applications namely image databases categorization and anomaly intrusion detection, are experimented to verify the effectiveness of the proposed approach. This work has been published in the *IEEE Transactions on Neural Networks and Learning Systems* [71].
- ❑ In Chapter 3, we develop a novel statistical approach of simultaneous clustering and feature selection for unsupervised learning. The proposed approach is based on finite generalized mixture models and variational inference learning. We apply the proposed approach to both synthetic data and a challenging application which concerns human action videos categorization. This contribution has been published in the *IEEE Transactions on Knowledge and Data Engineering* [72].
- ❑ In Chapter 4, we propose a novel unsupervised clustering approach based on an infinite generalized mixture model with variational framework. We test the proposed approach using

both synthetic data and real-world applications involving visual scenes categorization, auto-annotation and retrieval. This research work has been published in *Pattern Recognition* [73].

- In Chapter 5, a novel online clustering approach based on infinite Beta-Liouville mixture models is proposed. The effectiveness of the proposed work is evaluated on three challenging real applications namely facial expression recognition, behavior modeling and recognition, and dynamic textures clustering. This work has been published in the *IEEE Transactions on Neural Networks and Learning Systems* [74].
- In Conclusions, we summarize our contributions and present some promising future works.

Chapter 2

Variational Learning for Finite Dirichlet Mixture Models

In this chapter, we focus on the variational learning of finite Dirichlet mixture models. Compared to other algorithms which are commonly used for mixture models (such as expectation-maximization), our approach has several advantages: first, the problem of over-fitting is prevented; furthermore, the complexity of the mixture model (i.e. the number of components) can be determined automatically and simultaneously with the parameters estimation as part of the Bayesian inference procedure; finally, since the whole inference process is analytically tractable with closed-form solutions, it may scale well to large applications. Both synthetic and real data, generated from real-life challenging applications namely image databases categorization and anomaly intrusion detection, are experimented to verify the effectiveness of the proposed approach.

2.1 The Finite Dirichlet Mixture Model

The Dirichlet distribution is the multivariate generalization of the Beta distribution, which offers considerable flexibility and ease of use. In contrast to Gaussian distribution which only contains symmetric modes, the Dirichlet distribution may have multiple symmetric and asymmetric modes. Additionally, the Dirichlet distribution is defined in the compact support $[0, 1]$ and can be easily generalized to be defined in a compact support of the form $[A, B]$, where $(A, B) \in \mathbb{R}^2$. Thus, the Dirichlet distribution is a better choice for modeling compactly supported data, such as images, text or videos [11].

A finite mixture of Dirichlet distributions with M components is represented by [75]

$$p(\vec{X}|\vec{\pi}, \vec{\alpha}) = \sum_{j=1}^M \pi_j \text{Dir}(\vec{X}|\vec{\alpha}_j) \quad (2.1)$$

where $\vec{\pi} = (\pi_1, \dots, \pi_M)$ denotes the mixing coefficients which are positive and sum to one. $\text{Dir}(\vec{X}|\vec{\alpha}_j)$ in Eq. (2.1) is the Dirichlet distribution of component j with its own positive parameters $\vec{\alpha}_j = (\alpha_{j1}, \dots, \alpha_{jD})$, and is defined by:

$$\text{Dir}(\vec{X}|\vec{\alpha}_j) = \frac{\Gamma(\sum_{l=1}^D \alpha_{jl})}{\prod_{l=1}^D \Gamma(\alpha_{jl})} \prod_{l=1}^D X_l^{\alpha_{jl}-1} \quad (2.2)$$

where $\vec{X} = (X_1, \dots, X_D)$ and $\sum_{l=1}^D X_l = 1$, $0 \leq X_l \leq 1$ for $l = 1, \dots, D$. It is noteworthy that the Dirichlet distribution is used here as a parent distribution to model directly the data and not as a prior to the multinomial.

Consider a set of N independent identically distributed vectors $\mathcal{X} = \{\vec{X}_1, \dots, \vec{X}_N\}$ assumed to be generated from the mixture distribution in Eq. (2.1), the likelihood function of the Dirichlet mixture model is given by

$$p(\mathcal{X}|\vec{\pi}, \vec{\alpha}) = \prod_{i=1}^N \left\{ \sum_{j=1}^M \pi_j \text{Dir}(\vec{X}_i|\vec{\alpha}_j) \right\} \quad (2.3)$$

It is convenient to interpret the finite Dirichlet mixture model in Eq. (2.1) as a latent variable model. Thus, for each vector \vec{X}_i , we introduce a M -dimensional binary random vector $\vec{Z}_i = \{Z_{i1}, \dots, Z_{iM}\}$, 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. The latent variables $\mathcal{Z} = \{\vec{Z}_1, \dots, \vec{Z}_N\}$ are actually hidden variables, so that do not appear explicitly in the model. The conditional distribution of \mathcal{Z} given the mixing coefficients $\vec{\pi}$ is defined as

$$p(\mathcal{Z}|\vec{\pi}) = \prod_{i=1}^N \prod_{j=1}^M \pi_j^{Z_{ij}} \quad (2.4)$$

Then, the likelihood function with latent variables, which is actually the conditional distribution of data set \mathcal{X} given the class labels \mathcal{Z} , can be written as

$$p(\mathcal{X}|\mathcal{Z}, \vec{\alpha}) = \prod_{i=1}^N \prod_{j=1}^M \text{Dir}(\vec{X}_i|\vec{\alpha}_j)^{Z_{ij}} \quad (2.5)$$

Having the data set \mathcal{X} , an important problem is the learning of the mixture parameters. By learning, we mean both the estimation of the parameters and the selection of the number of components M . In the following, we describe a variational inference approach, for finite Dirichlet mixture models, that can handle these two issues simultaneously.

2.2 Variational Inference for Finite Dirichlet Mixture Model

2.2.1 Variational Approximation

In order to estimate the parameters of the finite Dirichlet mixture model and to select the number of components correctly, we adopt the variational inference methodology proposed in [47] for finite Gaussian mixtures. The main idea of this framework is based on the estimation of the mixing coefficients $\vec{\pi}$ by maximizing the marginal likelihood $p(\mathcal{X}|\vec{\pi})$ given by

$$p(\mathcal{X}|\vec{\pi}) = \sum_{\mathcal{Z}} \int p(\mathcal{X}, \mathcal{Z}, \vec{\alpha}|\vec{\pi}) d\vec{\alpha} \quad (2.6)$$

where $p(\mathcal{X}, \mathcal{Z}, \vec{\alpha}|\vec{\pi})$ is the joint distribution of all the mixture model random variables conditioned on the mixing coefficients as

$$p(\mathcal{X}, \mathcal{Z}, \vec{\alpha}|\vec{\pi}) = p(\mathcal{X}|\mathcal{Z}, \vec{\alpha})p(\mathcal{Z}|\vec{\pi})p(\vec{\alpha}) \quad (2.7)$$

An important step now is to define a conjugate prior $p(\vec{\alpha})$ over the $\vec{\alpha}$ parameters. Since the Dirichlet belongs to the exponential family of distributions [76], a conjugate prior can be derived as follows [77]:

$$p(\vec{\alpha}_j) = f(\nu, \lambda) \left[\frac{\Gamma(\sum_{l=1}^D \alpha_{jl})}{\prod_{l=1}^D \Gamma(\alpha_{jl})} \right]^\nu \prod_{l=1}^D e^{-\lambda_l(\alpha_{jl}-1)} \quad (2.8)$$

where $f(\nu, \lambda)$ is a normalization coefficient and (ν, λ) are hyperparameters. Unfortunately, this formal conjugate prior for the Dirichlet distribution is intractable, mainly because of the difficulty to evaluate the normalization coefficient, and cannot be applied for the variational inference directly as it shall be clearer later. We decided, *faut de mieux*, to tackle this problem in a similar way as in [78] where the authors proposed a conjugate prior for the Beta distribution (i.e.

one-dimensional Dirichlet) within a variational framework. Indeed, we assume that the Dirichlet parameters are statistically independent and for each parameter α_{jl} , the Gamma distribution is adopted to approximate the conjugate prior as

$$p(\alpha_{jl}) = \mathcal{G}(\alpha_{jl}|u_{jl}, v_{jl}) = \frac{v_{jl}^{u_{jl}}}{\Gamma(u_{jl})} \alpha_{jl}^{u_{jl}-1} e^{-v_{jl}\alpha_{jl}} \quad (2.9)$$

where u_{jl} and v_{jl} are hyperparameters, subject to the constraints $u_{jl} > 0$ and $v_{jl} > 0$. Therefore, we have

$$p(\vec{\alpha}) = \prod_{j=1}^M \prod_{l=1}^D p(\alpha_{jl}) \quad (2.10)$$

By substituting Eqs. (2.4), (2.5) and (2.10) into Eq. (2.7), we obtain the joint distribution of all the random variables, conditioned on the mixing coefficients as

$$p(\mathcal{X}, \mathcal{Z}, \vec{\alpha}|\vec{\pi}) = \prod_{i=1}^N \prod_{j=1}^M \left[\pi_j \frac{\Gamma(\sum_{l=1}^D \alpha_{jl})}{\prod_{l=1}^D \Gamma(\alpha_{jl})} \prod_{l=1}^D X_{il}^{\alpha_{jl}-1} \right]^{Z_{ij}} \prod_{j=1}^M \prod_{l=1}^D \frac{v_{jl}^{u_{jl}}}{\Gamma(u_{jl})} \alpha_{jl}^{u_{jl}-1} e^{-v_{jl}\alpha_{jl}} \quad (2.11)$$

A directed graphical representation of this model is illustrated in Figure. 2.1.

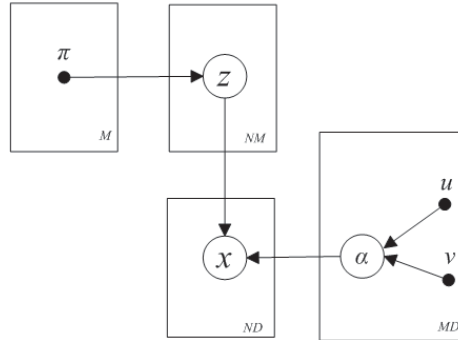


Figure 2.1: Graphical model representation of the finite Dirichlet mixture. Symbols in circles denote random variables; otherwise, they denote model parameters. Plates indicate repetition (with the number of repetitions in the lower right), and arcs describe conditional dependencies between variables.

Since the marginalization in Eq. (2.6) is intractable, we use the variational inference to find a tractable lower bound on $p(\mathcal{X}|\vec{\pi})$. To simplify the notation without loss of generality we define $\Theta = \{\mathcal{Z}, \vec{\alpha}\}$. The variational lower bound \mathcal{L} of the logarithm of the marginal likelihood $p(\mathcal{X}|\vec{\pi})$

can be found as

$$\mathcal{L}(Q) = \int Q(\Theta) \ln \frac{p(\mathcal{X}, \Theta | \vec{\pi})}{Q(\Theta)} d\Theta \quad (2.12)$$

where $Q(\Theta)$ is an approximation to the true posterior distribution $p(\Theta | \mathcal{X}, \vec{\pi})$. In this work, we adopt the factorization assumption for restricting the form of $Q(\Theta)$ as mentioned in Section 1.2. With this factorized approximation, the posterior distribution $Q(\Theta)$ can be factorized into disjoint tractable distributions as follows

$$Q(\Theta) = Q(\mathcal{Z})Q(\vec{\alpha}) = \left[\prod_{i=1}^N \prod_{j=1}^M Q(Z_{ij}) \right] \left[\prod_{j=1}^M \prod_{l=1}^D Q(\alpha_{jl}) \right] \quad (2.13)$$

By applying the general variational formula as shown in Eq. (1.9), we obtain the variational solutions for the factors of the variational posterior as (see Appendix A for details)

$$Q(\mathcal{Z}) = \prod_{i=1}^N \prod_{j=1}^M r_{ij}^{Z_{ij}} \quad (2.14)$$

$$Q(\vec{\alpha}) = \prod_{j=1}^M \prod_{l=1}^D \mathcal{G}(\alpha_{jl} | u_{jl}^*, v_{jl}^*) \quad (2.15)$$

where

$$r_{ij} = \frac{\rho_{ij}}{\sum_{j=1}^M \rho_{ij}} \quad (2.16)$$

$$\rho_{ij} = \exp \left\{ \ln \pi_j + \tilde{\mathcal{R}}_j + \sum_{l=1}^D (\bar{\alpha}_{jl} - 1) \ln X_{il} \right\} \quad (2.17)$$

$$\begin{aligned} \tilde{\mathcal{R}}_j = & \ln \frac{\Gamma(\sum_{l=1}^D \bar{\alpha}_{jl})}{\prod_{l=1}^D \Gamma(\bar{\alpha}_{jl})} + \sum_{l=1}^D \bar{\alpha}_{jl} \left[\Psi(\sum_{l=1}^D \bar{\alpha}_{jl}) - \Psi(\bar{\alpha}_{jl}) \right] \left[\langle \ln \alpha_{jl} \rangle - \ln \bar{\alpha}_{jl} \right] \\ & + \frac{1}{2} \sum_{l=1}^D \bar{\alpha}_{jl}^2 \left[\Psi'(\sum_{l=1}^D \bar{\alpha}_{jl}) - \Psi'(\bar{\alpha}_{jl}) \right] \langle (\ln \alpha_{jl} - \ln \bar{\alpha}_{jl})^2 \rangle \\ & + \frac{1}{2} \sum_{a=1}^D \sum_{b=1, a \neq b}^D \bar{\alpha}_{ja} \bar{\alpha}_{jb} \left[\Psi'(\sum_{l=1}^D \bar{\alpha}_{jl}) (\langle \ln \alpha_{ja} \rangle - \ln \bar{\alpha}_{ja}) (\langle \ln \alpha_{jb} \rangle - \ln \bar{\alpha}_{jb}) \right] \end{aligned} \quad (2.18)$$

$$u_{jl}^* = u_{jl} + \varphi_{jl}, \quad v_{jl}^* = v_{jl} - \vartheta_{jl} \quad (2.19)$$

$$\varphi_{jl} = \sum_{i=1}^N \langle Z_{ij} \rangle \bar{\alpha}_{jl} \left[\Psi\left(\sum_{k=1}^D \bar{\alpha}_{jk}\right) - \Psi(\bar{\alpha}_{jl}) + \sum_{k \neq l}^D \Psi'\left(\sum_{k=1}^D \bar{\alpha}_k\right) \bar{\alpha}_k (\langle \ln \alpha_k \rangle - \ln \bar{\alpha}_k) \right] \quad (2.20)$$

$$\vartheta_{jl} = \sum_{i=1}^N \langle Z_{ij} \rangle \ln X_{il} \quad (2.21)$$

where $\Psi(\cdot)$ and $\Psi'(\cdot)$ are the digamma and trigamma functions, respectively. The expected values in the above formulas are

$$\langle Z_{ij} \rangle = r_{ij}, \quad \bar{\alpha}_{jl} = \langle \alpha_{jl} \rangle = \frac{u_{jl}}{v_{jl}} \quad (2.22)$$

$$\langle \ln \alpha_{jl} \rangle = \Psi(u_{jl}^*) - \ln v_{jl}^* \quad (2.23)$$

$$\langle (\ln \alpha_{jl} - \ln \bar{\alpha}_{jl})^2 \rangle = [\Psi(u_{jl}^*) - \ln v_{jl}^*]^2 + \Psi'(u_{jl}^*) \quad (2.24)$$

2.2.2 Determining The Number of Components

Most conventional approaches tackle model selection problems via *cross-validation*. However, this approach is computational demanding and wasteful of data. In our work, the mixing coefficients $\vec{\pi}$ are treated as parameters, and point estimations of their values are evaluated by maximizing the variational likelihood bound $\mathcal{L}(Q)$. Setting the derivative of this lower bound with respect to $\vec{\pi}$ to zero gives:

$$\pi_j = \frac{1}{N} \sum_{i=1}^N r_{ij} \quad (2.25)$$

Note that this maximization is interleaved with the variational optimizations for $Q(\mathcal{Z})$ and $Q(\vec{\alpha})$. Indeed, components that provide insufficient contribution to explain the data would have their mixing coefficients driven to zero during the variational optimization, and so they can be effectively eliminated from the model through *automatic relevance determination* [79]. Thus, by starting with a relatively large initial value of M and then remove the redundant components after convergence, we can obtain the correct number of components in a single training run. It is also noteworthy that some works have shown that the variational objective is reduced to the Bayesian information criterion (BIC) as $N \rightarrow \infty$ [39, 40] which justifies the fact that the variational Bayes approach is more accurate than BIC for model selection (i.e. determination of the optimal number of mixture components) in practical settings [46].

2.2.3 Complete Variational Learning Algorithm

In variational learning, it is able to trace the convergence systematically by monitoring the variational lower bound during the re-estimation step [40]. Indeed, at each step of the iterative re-estimation procedure, the value of this bound should never decrease. Specifically, we evaluate the bound $\mathcal{L}(Q)$ at each interaction and terminate optimization if the amount of increase from one iteration to the next is less than a criterion. For the variational Dirichlet mixture model, the lower bound in Eq. (2.12) is evaluated as

$$\begin{aligned}
\mathcal{L}(Q) &= \sum_{\mathcal{Z}} \int Q(\mathcal{Z}, \vec{\alpha}) \ln \left\{ \frac{p(\mathcal{X}, \mathcal{Z}, \vec{\alpha} | \vec{\pi})}{Q(\mathcal{Z}, \vec{\alpha})} \right\} d\vec{\alpha} \\
&= \langle \ln p(\mathcal{X} | \mathcal{Z}, \vec{\alpha}) \rangle + \langle \ln p(\mathcal{Z} | \vec{\pi}) \rangle + \langle \ln p(\vec{\alpha}) \rangle - \langle \ln Q(\mathcal{Z}) \rangle - \langle \ln Q(\vec{\alpha}) \rangle \\
&= \sum_{i=1}^N \sum_{j=1}^M r_{ij} [\tilde{\mathcal{R}}_j + \sum_{l=1}^D (\bar{\alpha}_{jl}) \ln X_{il}] + \sum_{i=1}^N \sum_{j=1}^M r_{ij} \ln \pi_j - \sum_{i=1}^N \sum_{j=1}^M r_{ij} \ln r_{ij} \\
&\quad + \sum_{j=1}^M \sum_{l=1}^D \left\{ u_{jl} \ln v_{jl} - \ln \Gamma(u_{jl}) + (u_{jl} - 1) \langle \ln \alpha_{jl} \rangle - v_{jl} \bar{\alpha}_{jl} \right\} \\
&\quad - \sum_{j=1}^M \sum_{l=1}^D \left\{ u_{jl}^* \ln v_{jl}^* - \ln \Gamma(u_{jl}^*) + (u_{jl}^* - 1) \langle \ln \alpha_{jl} \rangle - v_{jl}^* \bar{\alpha}_{jl} \right\}
\end{aligned} \tag{2.26}$$

Since the solutions for the variational posterior Q and the value of the lower bound depend on

Algorithm 1 Variational Dirichlet mixtures

- 1: Choose the initial number of components M and the initial values for hyperparameters $\{u_{jl}\}$ and $\{v_{jl}\}$.
 - 2: Initialize the value of r_{ij} by K -Means algorithm.
 - 3: **repeat**
 - 4: The variational E-step: Update the variational solutions for $Q(\mathcal{Z})$ Eq. (2.14) and $Q(\vec{\alpha})$ Eq. (2.15).
 - 5: The variational M-step: maximize lower bound $\mathcal{L}(Q)$ with respect to the current value of $\vec{\pi}$ Eq. (2.25).
 - 6: **until** Convergence criteria is reached.
 - 7: Detect the optimal number of components M by eliminating the components with small mixing coefficients close to 0.
-

$\vec{\pi}$, the optimization of the variational Dirichlet mixture model can be solved using an EM-like algorithm with a guaranteed convergence (see, for instance, [39] for an empirical study and [48, 80] for a theoretical one). Indeed, local convergence has been formally and analytically proven in the case of the exponential family models with missing values [80] to which the finite Dirichlet mixture belongs. This local convergence is due to the convexity property of the exponential family of distributions. The complete algorithm can be summarized in Algorithm 1.

2.3 Experimental Results

In this section, we describe results that evaluate and indicate the effectiveness of the proposed approach using both synthetic and two real applications namely images categorization and anomaly intrusion detection. While the goal of the synthetic data is to investigate the accuracy of the variational approach as compared to the deterministic technique proposed in [75], the target of the real applications is to compare the performances of finite Dirichlet with finite Gaussian mixture models both learned in a variational way. In our experiments, we initialize the number of components to 15 with equal mixing coefficients. It is worth mentioning that multiple maxima in the variational bound may exist and therefore running the optimization several times with different initializations is helpful for discovering a good maximum in principle [47]. In practice we have perceived that, for the experiments involved in this chapter, poor initialization values of the hyperparameters $\{u_{jl}\}$ and $\{v_{jl}\}$ will considerably slow down the convergence speed. Based on our experiments, an optimal choice of the initial values of the hyperparameters $\{u_{jl}\}$ and $\{v_{jl}\}$ is to set them as 1 and 0.01, respectively. We have also considered hyperparameters initialization strategy previously proposed in [81] in the case of finite Gaussian mixture models. This approach is based on estimating the hyperparameters using maximum likelihood estimation of the parameters that result from successive runs of the EM algorithm. However, we have not observed, according to our experiments, significant improvement or influence on the learning process.

Table 2.1: Parameters of the different generated data sets. N denotes the total number of elements, n_j denotes the number of elements in cluster j . α_{j1} , α_{j2} , α_{j3} and π_j are the real parameters. $\hat{\alpha}_{j1}$, $\hat{\alpha}_{j2}$, $\hat{\alpha}_{j3}$ and $\hat{\pi}_j$ are the estimated parameters by variational inference. $\check{\alpha}_{j1}$, $\check{\alpha}_{j2}$, $\check{\alpha}_{j3}$ and $\check{\pi}_j$ are the estimated parameters using DM. We can observe that both algorithms are able to estimate unknown parameters, yet the variational algorithm always gives more accurate values.

	n_j	j	α_{j1}	α_{j2}	α_{j3}	π_j	$\hat{\alpha}_{j1}$	$\hat{\alpha}_{j2}$	$\hat{\alpha}_{j3}$	$\hat{\pi}_j$	$\check{\alpha}_{j1}$	$\check{\alpha}_{j2}$	$\check{\alpha}_{j3}$	$\check{\pi}_j$
Data set 1	200	1	12	30	45	0.5	12.59	31.29	45.56	0.50	11.08	31.33	45.28	0.482
($N = 400$)	200	2	32	50	16	0.5	33.58	50.20	15.64	0.50	31.27	50.64	16.38	0.518
Data set 2	200	1	12	30	45	0.4	13.91	35.40	51.07	0.398	13.96	32.41	48.53	0.327
($N = 500$)	200	2	32	50	16	0.4	32.68	51.71	16.81	0.401	32.53	48.96	16.79	0.451
	100	3	55	28	35	0.2	50.43	25.99	31.95	0.201	51.66	30.03	37.85	0.222
Data set 3	200	1	12	30	45	0.25	13.07	31.96	46.63	0.247	13.58	28.85	46.54	0.225
($N = 800$)	200	2	25	18	90	0.25	24.02	17.76	85.44	0.253	25.96	17.69	93.51	0.231
	200	3	55	28	35	0.25	54.89	27.73	34.13	0.249	56.43	29.72	33.93	0.286
	200	4	32	50	16	0.25	31.63	48.73	14.45	0.251	34.68	51.34	14.18	0.258
Data set 4	200	1	12	30	45	0.2	11.46	27.97	41.98	0.198	11.28	32.59	46.84	0.231
($N = 1000$)	100	2	25	18	90	0.1	25.16	19.23	93.36	0.098	23.13	19.50	87.92	0.145
	300	3	55	28	35	0.3	54.45	28.58	34.40	0.300	53.57	29.08	36.77	0.286
	200	4	32	50	16	0.2	36.23	55.47	18.04	0.198	35.31	53.09	19.61	0.174
	200	5	3	118	60	0.2	3.22	130.15	65.89	0.206	2.84	109.37	63.32	0.164
Data set 5	200	1	12	30	45	0.22	12.21	31.24	47.06	0.223	12.50	28.96	46.89	0.258
($N = 900$)	200	2	32	50	16	0.22	36.86	57.47	18.96	0.222	34.58	52.67	18.71	0.204
	200	3	55	28	35	0.22	55.83	28.75	34.84	0.221	57.62	27.04	36.18	0.237
	100	4	3	118	60	0.11	3.03	124.93	63.46	0.111	3.19	122.75	58.14	0.125
	100	5	25	18	90	0.11	25.72	17.96	90.71	0.112	26.15	18.03	88.96	0.092
	100	6	75	2	80	0.11	68.48	1.69	74.98	0.111	72.54	2.37	83.28	0.084
Data set 6	200	1	12	30	45	0.2	13.60	33.37	49.51	0.199	11.13	32.66	47.35	0.218
($N = 1000$)	200	2	32	50	16	0.2	33.72	53.26	16.67	0.201	35.68	47.52	17.31	0.207
	200	3	80	130	5	0.2	86.35	139.96	5.21	0.199	84.93	136.49	3.98	0.179
	100	4	3	118	60	0.1	2.98	124.84	64.67	0.100	3.50	115.03	66.37	0.081
	100	5	25	18	90	0.1	21.57	15.43	79.37	0.100	22.86	16.71	83.92	0.092
	100	6	75	2	80	0.1	64.15	1.69	67.23	0.101	81.63	2.67	70.38	0.135
	100	7	6	50	118	0.1	5.84	49.48	115.82	0.100	8.19	53.75	128.17	0.088

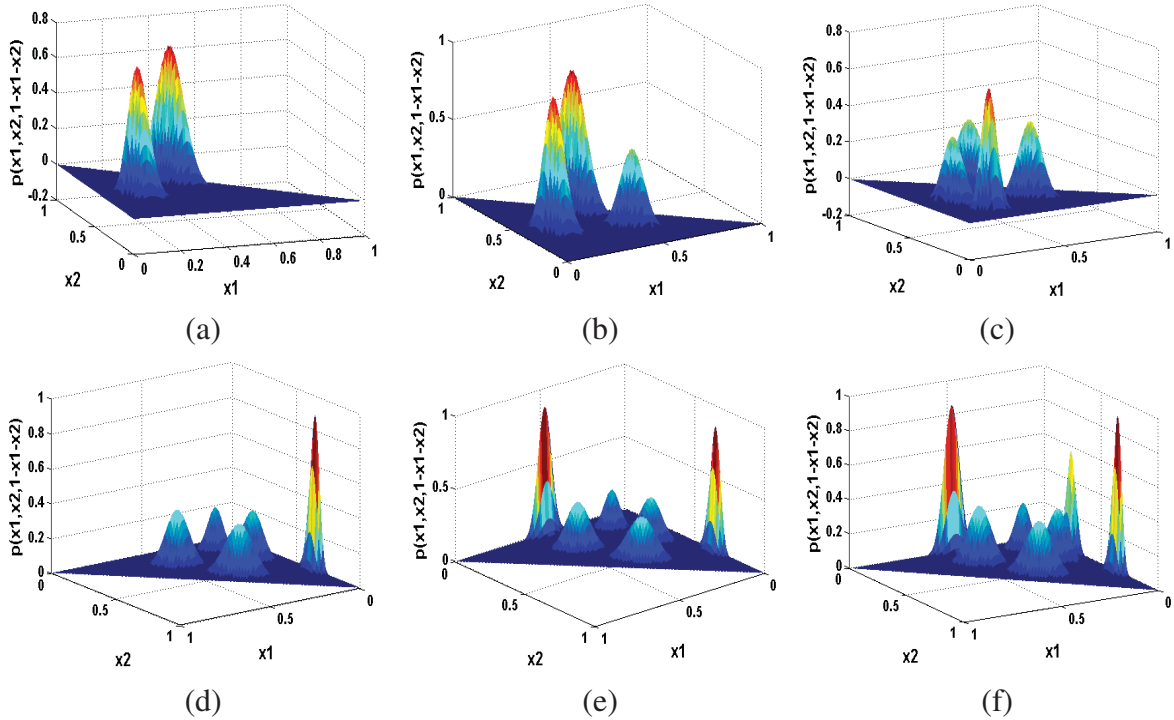


Figure 2.2: Mixture densities for the synthetic data sets. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4, (e) Data set 5, (f) Data set 6.

2.3.1 Synthetic Data

We first present the performance of our variational algorithm (varDM) in terms of estimation and selection, on six three-dimensional synthetic data. Please notice that, here we choose $D = 3$ purely for ease of representation. We tested the effectiveness of our algorithm for estimating the mixture’s parameters and selecting the number of components on generated data sets with different parameters. Table 2.1 shows the real and estimated parameters of each data set using both our variational algorithm and the deterministic approach (DM) proposed in [75]. Figure 2.2 represents the resultant mixtures with different shapes (symmetric and asymmetric modes).

In order to estimate the number of components, we apply directly our algorithm on these data sets (by starting with 15 components). The redundant components have estimated mixing coefficients close to 0 after convergence. By removing these redundant components, we obtain the correct number of components for each generated data set. Figure 2.3 illustrates the value of the

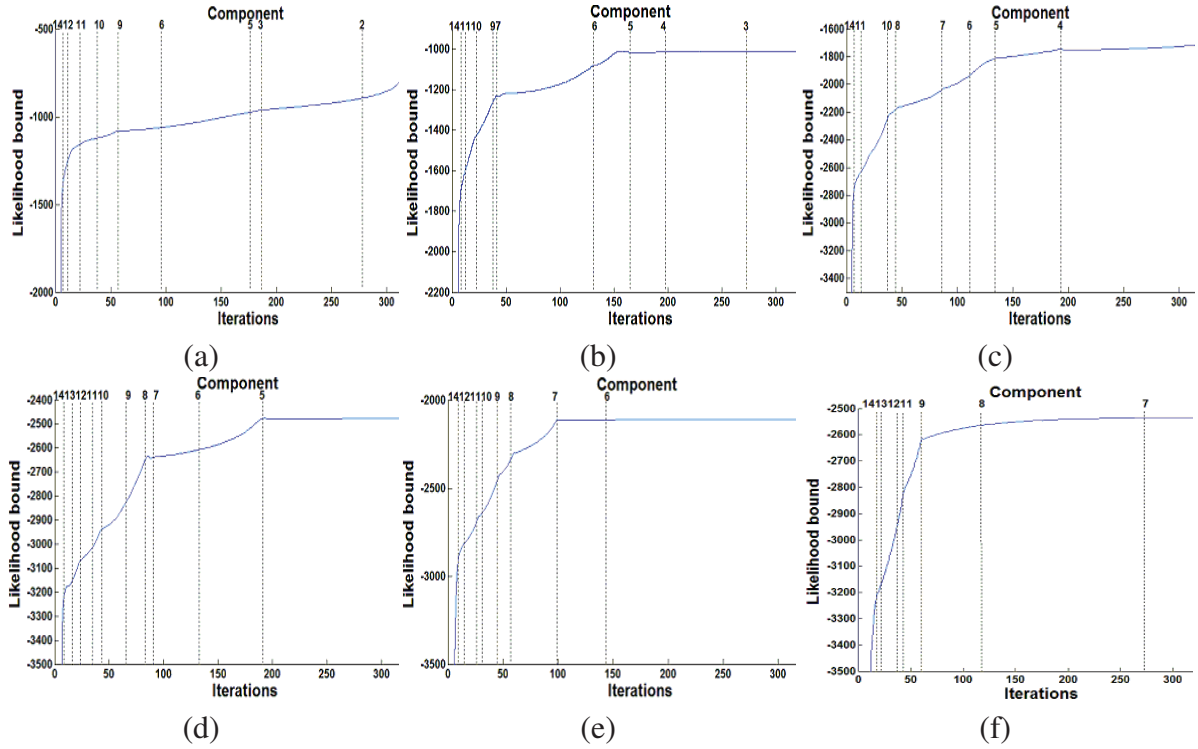


Figure 2.3: Variational likelihood bound for each iteration for the different generated data sets. The initial number of components is 15. Vertical dash lines indicate cancelation of components. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4, (e) Data set 5, (f) Data set 6.

variational likelihood bound in each iteration and shows that the likelihood bound increases at each iteration and in most cases it increases very fast when one of the mixing coefficients is close to 0 (i.e. shall be removed). We can verify the results of estimating the number of components by performing our variational optimization on a fixed number of components (i.e. without components elimination). Thus, the variational likelihood bound becomes a model selection score. As shown in Figure 2.4, we ran our algorithm by varying the number of mixture components from 2 to 15. According to this figure, it is clear that for each data set, the variational likelihood bound is maximum at the correct number of components which indicates that the variational likelihood bound can be used as an efficient criterion for model selection.

Moreover, we have performed a comparison between the numerical complexity of the proposed variational algorithm and the DM approach, in terms of overall computation time and number of

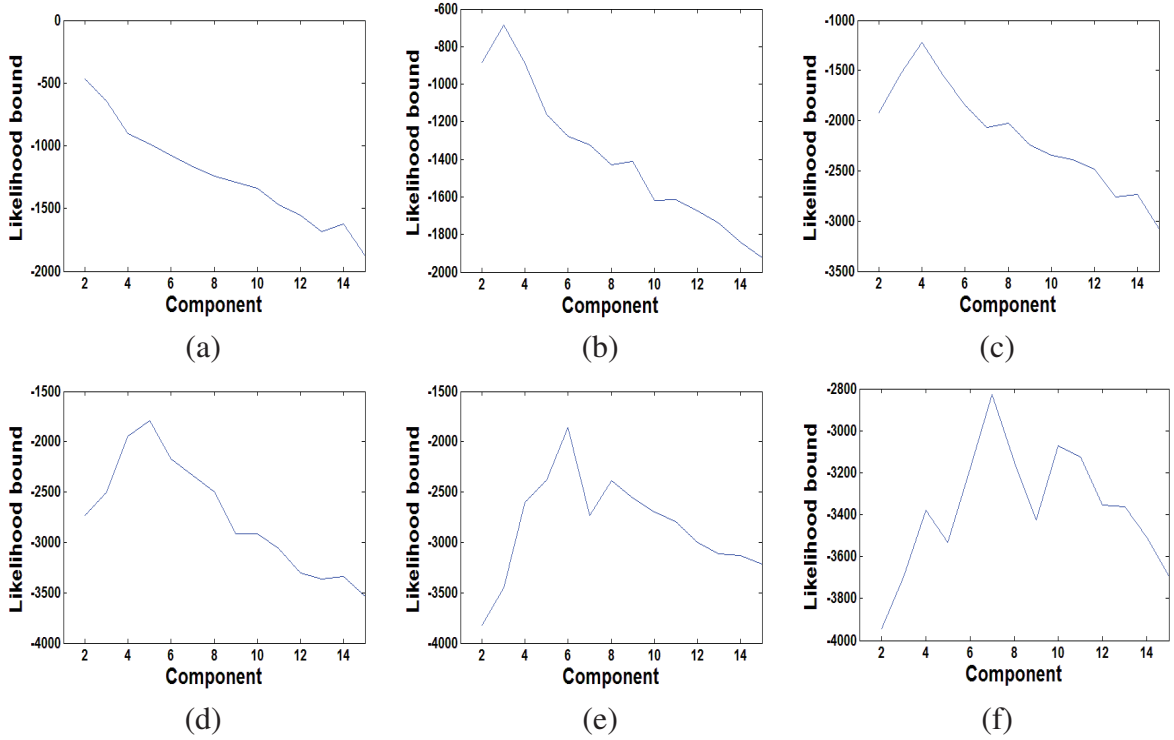


Figure 2.4: Variational likelihood bound as a function of the fixed assumed number of mixture components for the different generated data sets. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4, (e) Data set 5, (f) Data set 6.

iterations before convergence. The corresponding results are shown in Table 2.2. It is obvious that, for each data set, the proposed variational algorithm requires less iterations to converge and has a faster computational time than the deterministic one.

2.3.2 Images Categorization

In this part, we consider the problem of images categorization which is a fundamental problem in vision that has recently drawn considerable interest and seen great progress [82]. Applications include the automatic understanding of images, object recognition, image databases browsing and content-based images suggestion, recommendation and retrieval [83–85]. As the majority of computer vision tasks, an important step for accurate images categorization is the extraction of good descriptors (i.e. discriminative and invariant at the same time) to represent these images. Recently

Table 2.2: Run time (in seconds) and number of iterations required before convergence for varDM and DM.

VarDM			DM	
Data set	Run time	No. iterations	Run time	No. iterations
1	4.81	278	10.62	364
2	4.73	269	10.85	395
3	4.08	191	10.19	282
4	3.95	189	9.83	257
5	3.64	143	9.17	243
6	4.72	265	10.78	386

Table 2.3: Clustering Accuracies with varDM Model and varGM Model. M^* denotes the average number of clusters.

varDM			varGM	
Data set	M^*	Accuracy (%)	M^*	Accuracy (%)
A	4.85 ± 0.19	74.93 ± 1.62	4.56 ± 0.31	65.26 ± 1.38
B	4.03 ± 0.14	78.01 ± 1.56	4.41 ± 0.52	68.34 ± 1.29

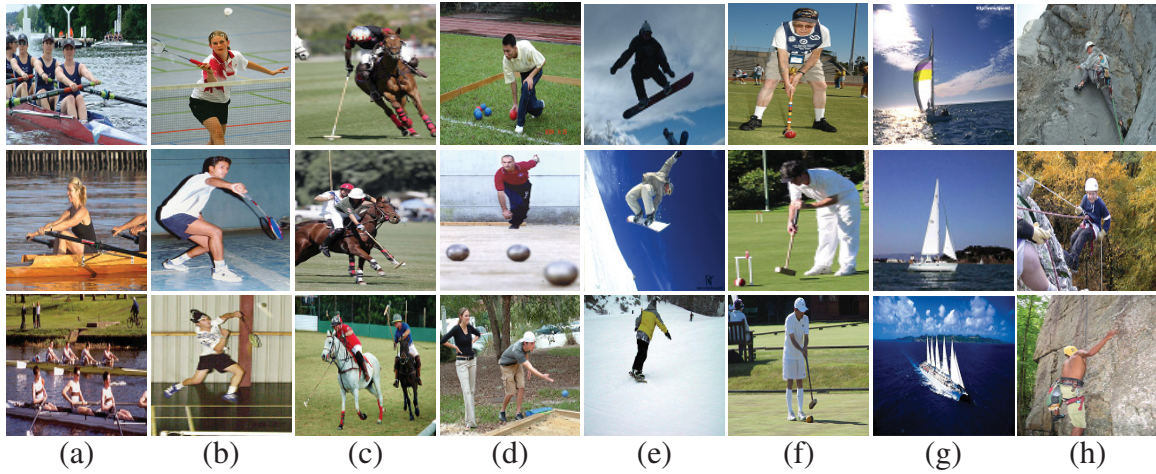


Figure 2.5: Sample images from each group of sports event data set: (a) Rowing. (b) Badminton. (c) polo. (d) Bocce. (e) Snow Boarding. (f) Croquet. (g) Sailing. (h) Rock climbing.

methods based on the bag-of-features approach have shown to give excellent results [86, 87]. In this subsection we therefore follow this class of methods and in particular the one proposed in [87]. First, key points in the images are detected using one of the various detectors and local descriptors which should be invariant to image transformation, occlusions and variations of illumination are extracted. Then, these local descriptors are grouped into \mathcal{W} homogenous clusters, using a clustering or vector quantization algorithm such as K-Means. Therefore, each cluster center is treated as a visual word and a visual vocabulary is build with \mathcal{W} visual words. Applying the paradigm of *bag-of-words*, a \mathcal{W} -dimensional histogram representing the frequency of each visual word is calculated for each image. Finally, the Probabilistic Latent Semantic Analysis (pLSA) model [88] is applied to reduce the dimensionality of the resulting histograms allowing the representation of images as proportional vectors. Thus, our variational Dirichlet mixture modeling framework provides a natural setting to address the categorization task.

In our experiments, we have considered the FeiFei’s sports event data set containing 8 categories of sports scenes: rowing (250 images), badminton (200 images), polo (182 images), bocce (137 images), snow boarding (190 images), croquet (236 images), sailing (190 images), and rock climbing (194 images). Thus, the data set contains 1,579 images in total. We normalize each image into a size of 256×256 pixels. Examples of images from each categories are shown in Figure 2.5.

Table 2.4: Average Rounded Confusion Matrix using the varDM Model to categorize Data Set A.

	Rowing	Badminton	Sailing	Croquet	Rock
Rowing	109	5	28	3	5
Badminton	8	116	0	10	18
Sailing	19	3	122	2	4
Croquet	9	25	1	104	11
Rock	8	18	3	10	111

In our experiments, the key points of each image are detected using the Difference-of-Gaussian (DoG) interest point detector [89] and described using Scale-Invariant Feature Transform (SIFT) descriptor, resulting on 128-dimensional vector for each key point [89]. Then, an accelerated version of the K-Means algorithm [90] is used to cluster all the SIFT vectors into a visual vocabulary of 700 visual words. Note that, the number of visual words is user-specified. Based on our experiments, the best results have been obtained when $\mathcal{W} = [600, 800]$. Then, the new representation for each image is calculated through the pLSA model by considering 35 aspects.

Two data sets are used for testing our algorithm. Data set A consists of 750 images from five categories of the sports event data set: rowing, badminton, sailing, croquet and rock climbing. Data set B consists of 600 images from four different categories of the sports event data set: rowing, polo, snow boarding and bocce. Table 2.3 shows the average number of clusters and the average classification accuracies using both varDM and Gaussian mixture (varGM) models learned by running their respective variational algorithms 20 times. Tables 2.4 and 2.5 show the confusion matrices when applying varDM for data sets A and B, respectively. According to the obtained results we can clearly see that the varDM outperforms the varGM in terms of both categorization accuracy and selection of the optimal number of image categories.

2.3.3 Anomaly Intrusion Detection

Nowadays, intrusion detection systems (IDSs) are becoming more and more important as computer security vulnerabilities and flaws are being discovered everyday [91–94]. The main goal

Table 2.5: Average Rounded Confusion Matrix using the varDM Model to categorize Data Set B.

	Rowing	Polo	Snow	Bocce
Rowing	115	17	8	10
Polo	6	124	13	7
Snow	21	6	109	14
Bocce	5	3	15	127

is to establish approaches which can scan network activities and detect suspicious patterns that may have been derived from intrusion attacks. Intrusion detection is based on the assumption that intrusive activities are noticeably diverse from normal system activities and hence detectable. According to the analysis methods, IDSs can be classified into two main categories: *misuse* detection and *anomaly* detection systems. In misuse detection systems, pre-defined attack patterns and signatures are used for detecting known attacks. Alternatively, anomaly detection systems detect unknown attacks by observing deviations from normal activities of the system. Anomaly detection has the advantage of detecting new types of intrusions. In our work, we first use our mixture model to learn patterns of normal and intrusive activities from training data. Then, we detect and classify intrusive activities which are deviated from the normal activities in a testing data set.

Data Set Description

The well-known KDD Cup 1999 Data ¹ is used to investigate our mixture model. This data set (tcpdump file) was collected at MIT Lincoln laboratory for the 1998 DARPA intrusion detection evaluation program by simulating attacks on a typical U.S. Air Force Lan. Each data instance in the data set is a connection record obtained from the simulated intrusions with 41 features (such as duration, dst.bytes, etc). A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. The training data consists of 494,021 data instances of which 97,277 are normal and 396,744 are attacks. The testing set contains 311,029 data instances

¹<http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>

Table 2.6: Confusion Matrix for Intrusion Detection with Variational Dirichlet Mixture Model.

	Normal	DOS	R2L	U2R	Probe
Normal	49081	1169	9012	1042	289
DOS	38859	181372	562	309	8751
R2L	3617	169	9657	243	95
U2R	185	63	137	2185	66
Probe	401	185	62	149	3369

Table 2.7: Intrusion Detection Results Using different approaches.

Algorithm	varDM	DM	varGM	GM
Accuracy (%)	78.75	75.53	73.34	71.29

of which 60,593 are normal and 250,436 are attacks. All of these attacks fall into one of the following four categories: DOS: denial-of-service (e.g. syn flood); R2L: unauthorized access from a remote machine (e.g. guessing password); U2R: unauthorized access to local superuser (root) privileges (e.g. buffer overflow attack) and Probing: surveillance and other probing (e.g. port scanning).

Results

In our data set, each data instance contains 41 features in which 34 are numeric and 7 are symbolic. In our experiments, only the 34 numeric features are used (i.e. each data is then represented as a 34-dimensional vector). Since the features are on quite different scales in the data set, we need to normalize them such that one feature would not dominant the others in our algorithm. Table 2.6 shows the obtained confusion matrix using our varDM. According to this matrix the detection rate is 78.75%. A summary of the detection results by applying other approaches namely the DM, the varGM, and the Gaussian mixtures (GM) are given in table 2.7. According to these results, we can say that the varDM outperforms significantly, according a student's t-test, the other approaches.

Chapter 3

Unsupervised Feature Selection for High-Dimensional Non-Gaussian Data Clustering with Variational Inference

Clustering has been a subject of extensive research in data mining, pattern recognition and other areas for several decades. The main goal is to assign samples, which are typically non-Gaussian and expressed as points in high-dimensional feature spaces, to one of a number of clusters. It is well-known that in such high-dimensional settings, the existence of irrelevant features generally compromises modeling capabilities. In this chapter, we propose a variational inference framework for unsupervised non-Gaussian feature selection, in the context of finite generalized Dirichlet (GD) mixture-based clustering. Under the proposed principled variational framework, we simultaneously estimate, in a closed-form, all the involved parameters and determine the complexity (i.e. both model and feature selection) of the GD mixture. Extensive simulations using synthetic data along with an analysis of human action videos demonstrate that our variational approach achieves better results than comparable techniques.

3.1 Model specification

The GD distribution is the generalization of the Dirichlet distribution. It has a more general covariance structure (can be positive or negative) than Dirichlet distribution and offers high flexibility and ease of use for the approximation of both symmetric and asymmetric distributions. Compared to the Gaussian distribution, the GD distribution has a smaller number of parameters that makes the estimation and the selection more accurate.

A GD distribution of a D -dimensional random vector \vec{Y} is defined as

$$\text{GD}(\vec{Y}|\vec{\alpha}_j, \vec{\beta}_j) = \prod_{l=1}^D \frac{\Gamma(\alpha_{jl} + \beta_{jl})}{\Gamma(\alpha_{jl})\Gamma(\beta_{jl})} Y_l^{\alpha_{jl}-1} \left(1 - \sum_{k=1}^l Y_k\right)^{\gamma_{jl}} \quad (3.1)$$

where $\sum_{l=1}^D Y_l < 1$ and $0 < Y_l < 1$ for $l = 1, \dots, D$. $\vec{\alpha}_j = (\alpha_{j1}, \dots, \alpha_{jD})$ and $\vec{\beta}_j = (\beta_{j1}, \dots, \beta_{jD})$ are the parameters of the GD distribution, such that, $\alpha_{jl} > 0$, $\beta_{jl} > 0$, $\gamma_{jl} = \beta_{jl} - \alpha_{jl+1} - \beta_{jl+1}$ for $l = 1, \dots, D-1$, and $\gamma_{jD} = \beta_{jD} - 1$. Assume that we have a set of N independent and identically distributed vectors $\mathcal{Y} = (\vec{Y}_1, \dots, \vec{Y}_N)$, where each vector $\vec{Y}_i = (Y_{i1}, \dots, Y_{iD})$ is assumed to be sampled from a finite GD mixture model with M components [17]:

$$p(\vec{Y}_i|\vec{\pi}, \vec{\alpha}, \vec{\beta}) = \sum_{j=1}^M \pi_j \text{GD}(\vec{Y}_i|\vec{\alpha}_j, \vec{\beta}_j), \quad (3.2)$$

where $\vec{\alpha} = (\vec{\alpha}_1, \dots, \vec{\alpha}_M)$ and $\vec{\beta} = (\vec{\beta}_1, \dots, \vec{\beta}_M)$. $\vec{\alpha}_j$ and $\vec{\beta}_j$ are the parameters of the GD distribution representing component j . $\vec{\pi} = (\pi_1, \dots, \pi_M)$ represents the mixing coefficients with the constraints that are positive and sum to one.

According to an interesting mathematical property of the GD thoroughly discussed in [60], the data point \vec{Y}_i can be transformed using a geometric transformation into another D -dimensional data point \vec{X}_i with independent features. Then, the finite GD mixture model is equivalent to the following mixture model

$$p(\vec{X}_i|\vec{\pi}, \vec{\alpha}, \vec{\beta}) = \sum_{j=1}^M \pi_j \prod_{l=1}^D \text{Beta}(X_{il}|\alpha_{jl}, \beta_{jl}) \quad (3.3)$$

where $\vec{X}_i = (X_{i1}, \dots, X_{iD})$, $X_{i1} = Y_{i1}$ and $X_{il} = Y_{il}/(1 - \sum_{k=1}^{l-1} Y_{ik})$ for $l > 1$, and $\text{Beta}(X_{il}|\alpha_{jl}, \beta_{jl})$ is a Beta distribution defined with parameters $(\alpha_{jl}, \beta_{jl})$:

$$\text{Beta}(X_{il}|\alpha_{jl}, \beta_{jl}) = \frac{\Gamma(\alpha_{jl} + \beta_{jl})}{\Gamma(\alpha_{jl})\Gamma(\beta_{jl})} X_{il}^{\alpha_{jl}-1} (1 - X_{il})^{\beta_{jl}-1} \quad (3.4)$$

Consequently, the estimation of a D -dimensional GD is reduced to D estimations of one-dimensional Beta distributions which is interesting for multidimensional data. Moreover, the independence between the features, in the transformed data space, becomes a fact rather than an assumption as

considered in previous unsupervised feature selection Gaussian mixture-based approaches [54, 57, 59, 65].

Next, we assign a binary latent variable $\vec{Z}_i = (Z_{i1}, \dots, Z_{iM})$ to each observation \vec{X}_i , such that $Z_{ij} \in \{0, 1\}$, $\sum_{j=1}^M Z_{ij} = 1$, $Z_{ij} = 1$ if \vec{X}_i belongs to component j and equal to 0, otherwise. The conditional distribution of latent variables $\mathcal{Z} = (\vec{Z}_1, \dots, \vec{Z}_N)$, given the mixing coefficients $\vec{\pi}$, is defined as

$$p(\mathcal{Z}|\vec{\pi}) = \prod_{i=1}^N \prod_{j=1}^M \pi_j^{Z_{ij}}. \quad (3.5)$$

It is noteworthy that the previous model assumes actually that all the features X_{il} are equally important for the clustering task which is not realistic in general, since some of the features might be “noise” and do not contribute to clustering process. In our work, we adopt the unsupervised feature selection scheme that has been proposed in [60] by approximating the feature distribution as

$$p(X_{il}|W_{ikl}, \phi_{il}, \alpha_{jl}, \beta_{jl}, \lambda_{kl}, \tau_{kl}) \simeq \left(\text{Beta}(X_{il}|\alpha_{jl}, \beta_{jl}) \right)^{\phi_{il}} \left(\prod_{k=1}^K \text{Beta}(X_{il}|\lambda_{kl}, \tau_{kl})^{W_{ikl}} \right)^{1-\phi_{il}} \quad (3.6)$$

where ϕ_{il} is a binary latent variable, such that $\phi_{il} = 1$ if feature l is relevant (i.e. supposed to follow a Beta distribution, $\text{Beta}(X_{il}|\alpha_{jl}, \beta_{jl})$, that depends on the class labels), and $\phi_{il} = 0$ if feature l is irrelevant and then supposed to follow a mixture of K Beta distributions, $\text{Beta}(X_{il}|\lambda_{kl}, \tau_{kl})$, independent from the class labels. In addition, W_{ikl} is a binary variable such that $\sum_{k=1}^K W_{ikl} = 1$. When $W_{ikl} = 1$, it indicates that X_{il} comes from the k th component of the irrelevant Beta mixture model. Assuming that W_{ikl} represents the elements of \mathcal{W} , the marginal distribution of \mathcal{W} is defined as

$$p(\mathcal{W}|\vec{\eta}) = \prod_{i=1}^N \prod_{k=1}^K \prod_{l=1}^D \eta_{kl}^{W_{ikl}} \quad (3.7)$$

where η_{kl} represents the prior probability that X_{il} comes from the k th component of the irrelevant Beta distribution, and $\sum_{k=1}^K \eta_{kl} = 1$.

The prior distribution of $\vec{\phi}$ is defined as

$$p(\vec{\phi}|\vec{\epsilon}) = \prod_{i=1}^N \prod_{l=1}^D \epsilon_{l_1}^{\phi_{il}} \epsilon_{l_2}^{1-\phi_{il}} \quad (3.8)$$

where each ϕ_{il} is a Bernoulli variable such that $p(\phi_{il} = 1) = \epsilon_{l_1}$ and $p(\phi_{il} = 0) = \epsilon_{l_2}$. The vector $\vec{\epsilon} = (\vec{\epsilon}_1, \dots, \vec{\epsilon}_D)$ represents the features saliencies (i.e. the probabilities that the features are relevant) such that $\vec{\epsilon}_l = (\epsilon_{l_1}, \epsilon_{l_2})$ and $\epsilon_{l_1} + \epsilon_{l_2} = 1$.

Next, Gamma distributions are adopted to approximate conjugate priors over parameters $\vec{\alpha}, \vec{\beta}, \vec{\lambda}$ and $\vec{\tau}$ as suggested recently in [78], by assuming that the different model's parameters are independent: $p(\vec{\alpha}) = \mathcal{G}(\vec{\alpha}|\vec{u}, \vec{v})$, $p(\vec{\beta}) = \mathcal{G}(\vec{\beta}|\vec{p}, \vec{q})$, $p(\vec{\lambda}) = \mathcal{G}(\vec{\lambda}|\vec{g}, \vec{h})$, $p(\vec{\tau}) = \mathcal{G}(\vec{\tau}|\vec{s}, \vec{t})$, where $\mathcal{G}(\cdot)$ is the Gamma distribution and is defined as

$$\mathcal{G}(x|a, b) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}. \quad (3.9)$$

It is noteworthy that $\vec{\epsilon}, \vec{\pi}$ and $\vec{\eta}$ will be considered as parameters and not as random variables within our framework, thus priors shall not be imposed on them as we will explain further in next section.

3.2 Variational Learning of the Model

In order to estimate the parameters of the finite GD mixture model and to select the number of components correctly, we adopt the variational inference methodology proposed in [47]. We are mainly motivated by the good results obtained recently using variational learning techniques in machine learning applications in general [95, 96] and for the unsupervised feature selection problem in particular [59, 65]. To simplify notation, let us define $\Theta = \{\mathcal{Z}, \mathcal{W}, \vec{\phi}, \vec{\alpha}, \vec{\beta}, \vec{\lambda}, \vec{\tau}\}$ as the set of non-observed random variables and denote $\Lambda = \{\vec{\pi}, \vec{\eta}, \vec{\epsilon}\}$ as the set of parameters. Our goal is to optimize the values of Λ by maximizing the marginal likelihood $p(\mathcal{X}|\Lambda)$. Since this marginalization is intractable, variational inference is then adopted to find a tractable lower bound on $p(\mathcal{X}|\Lambda)$. By applying Jensen's inequality, the lower bound \mathcal{L} of the logarithm of $p(\mathcal{X}|\Lambda)$ can be found as

$$\ln p(\mathcal{X}|\Lambda) \geq \int Q(\Theta) \ln \frac{p(\mathcal{X}, \Theta|\Lambda)}{Q(\Theta)} d\Theta = \mathcal{L}(Q), \quad (3.10)$$

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

Then, we adopt the factorization assumptions for restricting the form of $Q(\Theta)$, such that

$$Q(\Theta) = Q(\mathcal{Z})Q(\vec{\phi})Q(\mathcal{W})Q(\vec{\alpha})Q(\vec{\beta})Q(\vec{\lambda})Q(\vec{\tau}). \quad (3.11)$$

In order to maximize the lower bound $\mathcal{L}(Q)$, we need to make a variational optimization of $\mathcal{L}(Q)$ with respect to each of the factors in turn. For a specific factor $Q_m(\Theta_m)$, the general expression for its optimal solution can be found by

$$Q_m(\Theta_m) = \frac{\exp\langle \ln p(\mathcal{X}, \Theta|\Lambda) \rangle_{\neq m}}{\int \exp\langle \ln p(\mathcal{X}, \Theta|\Lambda) \rangle_{\neq m} d\Theta}, \quad (3.12)$$

where $\langle \cdot \rangle_{\neq m}$ is the expectation with respect to all the factors except for $Q_m(\Theta_m)$. By applying Eq. (3.12) to each variational factor, we obtain the optimal solutions for the factors of the variational posterior as:

$$Q(\mathcal{Z}) = \prod_{i=1}^N \prod_{j=1}^M r_{ij}^{Z_{ij}}, \quad Q(\vec{\phi}) = \prod_{i=1}^N \prod_{l=1}^D f_{il}^{\phi_{il}} (1 - f_{il})^{(1-\phi_{il})}, \quad Q(\mathcal{W}) = \prod_{i=1}^N \prod_{l=1}^D \prod_{k=1}^K m_{ikl}^{W_{ikl}} \quad (3.13)$$

$$Q(\vec{\alpha}) = \prod_{j=1}^M \prod_{l=1}^D \mathcal{G}(\alpha_{jl} | u_{jl}^*, v_{jl}^*), \quad Q(\vec{\beta}) = \prod_{j=1}^M \prod_{l=1}^D \mathcal{G}(\beta_{jl} | p_{jl}^*, q_{jl}^*) \quad (3.14)$$

$$Q(\vec{\lambda}) = \prod_{k=1}^K \prod_{l=1}^D \mathcal{G}(\lambda_{kl} | g_{kl}^*, h_{kl}^*), \quad Q(\vec{\tau}) = \prod_{k=1}^K \prod_{l=1}^D \mathcal{G}(\tau_{kl} | s_{kl}^*, t_{kl}^*) \quad (3.15)$$

where we define

$$r_{ij} = \frac{\rho_{ij}}{\sum_{d=1}^M \rho_{id}}, \quad f_{il} = \frac{\delta_{il}^{(\phi_{il})}}{\delta_{il}^{(\phi_{il})} + \delta_{il}^{(1-\phi_{il})}}, \quad m_{ikl} = \frac{\varphi_{ikl}}{\sum_{d=1}^K \varphi_{ild}}$$

$$\rho_{ij} = \exp \left\{ \ln \pi_j + \sum_{l=1}^D \langle \phi_{il} \rangle [\tilde{\mathcal{R}}_{jl} + (\bar{\alpha}_{jl} - 1) \ln X_{il} + (\bar{\beta}_{jl} - 1) \ln(1 - X_{il})] \right\}$$

$$\delta_{il}^{(\phi_{il})} = \exp \left\{ \sum_{j=1}^M \langle Z_{ij} \rangle [\tilde{\mathcal{R}}_{jl} + (\bar{\alpha}_{jl} - 1) \ln X_{il} + (\bar{\beta}_{jl} - 1) \ln(1 - X_{il})] + \ln \epsilon_l \right\}$$

$$\delta_{il}^{(1-\phi_{il})} = \exp \left\{ \sum_{k=1}^K \langle W_{ikl} \rangle [\tilde{\mathcal{F}}_{kl} + (\bar{\lambda}_{kl} - 1) \ln X_{il} + (\bar{\tau}_{kl} - 1) \ln(1 - X_{il})] + \ln(1 - \epsilon_l) \right\}$$

$$\varphi_{ikl} = \exp \left\{ \langle 1 - \phi_{il} \rangle [\tilde{\mathcal{F}}_{kl} + (\bar{\lambda}_{kl} - 1) \ln X_{il} + (\bar{\tau}_{kl} - 1) \ln(1 - X_{il})] + \ln \eta_{kl} \right\}$$

$$u_{jl}^* = u_{jl} + \sum_{i=1}^N \langle Z_{ij} \rangle \langle \phi_{il} \rangle \bar{\alpha}_{jl} [\psi(\bar{\alpha}_{jl} + \bar{\beta}_{jl}) - \psi(\bar{\alpha}_{jl}) + \bar{\beta}_{jl} \psi'(\bar{\alpha}_{jl} + \bar{\beta}_{jl}) (\langle \ln \beta_{jl} \rangle - \ln \bar{\beta}_{jl})]$$

$$\begin{aligned}
v_{jl}^* &= v_{jl} - \sum_{i=1}^N \langle Z_{ij} \rangle \langle \phi_{il} \rangle \ln X_{il}, & q_{jl}^* &= q_{jl} - \sum_{i=1}^N \langle Z_{ij} \rangle \langle \phi_{il} \rangle \ln(1 - X_{il}) \\
p_{jl}^* &= p_{jl} + \sum_{i=1}^N \langle Z_{ij} \rangle \langle \phi_{il} \rangle \bar{\beta}_{jl} [\psi(\bar{\alpha}_{jl} + \bar{\beta}_{jl}) - \psi(\bar{\beta}_{jl}) + \bar{\alpha}_{jl} \psi'(\bar{\alpha}_{jl} + \bar{\beta}_{jl}) (\langle \ln \alpha_{jl} \rangle - \ln \bar{\alpha}_{jl})] \\
g_{kl}^* &= g_{kl} + \sum_{i=1}^N \langle 1 - \phi_{il} \rangle \langle W_{ikl} \rangle \bar{\lambda}_{kl} [\psi(\bar{\lambda}_{kl} + \bar{\tau}_{kl}) - \psi(\bar{\lambda}_{kl}) + \bar{\tau}_{kl} \psi'(\bar{\lambda}_{kl} + \bar{\tau}_{kl}) (\langle \ln \tau_{kl} \rangle - \ln \bar{\tau}_{kl})] \\
h_{kl}^* &= h_{kl} - \sum_{i=1}^N \langle 1 - \phi_{il} \rangle \langle W_{ikl} \rangle \ln X_{il}, & t_{kl}^* &= t_{kl} - \sum_{i=1}^N \langle 1 - \phi_{il} \rangle \langle W_{ikl} \rangle \ln(1 - X_{il}) \\
s_{kl}^* &= s_{kl} + \sum_{i=1}^N \langle 1 - \phi_{il} \rangle \langle W_{ikl} \rangle \bar{\tau}_{kl} [\psi(\bar{\lambda}_{kl} + \bar{\tau}_{kl}) - \psi(\bar{\tau}_{kl}) + \bar{\lambda}_{kl} \psi'(\bar{\lambda}_{kl} + \bar{\tau}_{kl}) (\langle \ln \lambda_{kl} \rangle - \ln \bar{\lambda}_{kl})] \\
\bar{\alpha}_{jl} &= \langle \alpha_{jl} \rangle = \frac{u_{jl}^*}{v_{jl}^*}, & \bar{\beta}_{jl} &= \langle \beta_{jl} \rangle = \frac{p_{jl}^*}{q_{jl}^*}, & \bar{\lambda}_{kl} &= \langle \lambda_{kl} \rangle = \frac{g_{kl}^*}{h_{kl}^*}, & \bar{\tau}_{kl} &= \langle \tau_{kl} \rangle = \frac{s_{kl}^*}{t_{kl}^*}
\end{aligned}$$

where $\langle \cdot \rangle$ represents an expected value, the $\psi(\cdot)$ is the digamma function and defined as: $\psi(a) = d \ln \Gamma(a) / da$. Notice that, $\tilde{\mathcal{R}}$ and $\tilde{\mathcal{F}}$ are the lower bound approximations of $\mathcal{R} = \langle \ln \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \rangle$ and $\mathcal{F} = \langle \ln \frac{\Gamma(\lambda+\tau)}{\Gamma(\lambda)\Gamma(\tau)} \rangle$, respectively. Since these expectations are intractable, we use the second-order Taylor series expansion to find their lower bounds as proposed in [78]. The expected values in the above formulas are given by

$$\langle Z_{ij} \rangle = r_{ij}, \quad \langle W_{ilk} \rangle = m_{ilk}, \quad \langle \phi_{il} \rangle = f_{il}, \quad \langle 1 - \phi_{il} \rangle = 1 - f_{il}$$

$$\langle \ln \alpha \rangle = \psi(u^*) - \ln v^*, \quad \langle \ln \beta \rangle = \psi(p^*) - \ln q^*, \quad \langle \ln \lambda \rangle = \psi(g^*) - \ln h^*, \quad \langle \ln \tau \rangle = \psi(s^*) - \ln t^*$$

Now, we can obtain a variational lower bound $\mathcal{L}(Q)$ which approximates the true marginal log likelihood $\ln p(\mathcal{X}|\Lambda)$ by using the variational solutions to each factor. The model parameters Λ can be estimated by maximizing $\mathcal{L}(Q)$ with respect to $\vec{\pi}$, $\vec{\eta}$ and $\vec{\epsilon}$. Thus, by setting the derivative of the lower bound with respect to π_j , η_{lk} and ϵ_l to zero, we get

$$\pi_j = \frac{1}{N} \sum_{i=1}^N r_{ij}, \quad \eta_{lk} = \frac{1}{N} \sum_{i=1}^N m_{ilk}, \quad \epsilon_l = \frac{1}{N} \sum_{i=1}^N f_{il} \quad (3.16)$$

Since the solutions for the variational posterior Q and the value of the lower bound depend on the values of $\vec{\pi}$, $\vec{\eta}$ and $\vec{\epsilon}$, the optimization of the model can be solved in a way analogous to the

Algorithm 2 Variational GD mixture with feature selection

Choose the initial number of components M and K .

Initialize the values for hyper-parameters $\vec{u}, \vec{v}, \vec{p}, \vec{q}, \vec{g}, \vec{h}, \vec{s}$ and \vec{t} .

Initialize the values of r_{ij} and m_{ikl} by K -Means algorithm.

repeat

The variational E-step: Update the variational solutions through Eq. (3.13) to Eq. (3.15).

The variational M-step: maximize lower bound $\mathcal{L}(Q)$ with respect to the current values of $\vec{\pi}, \vec{\eta}$ and $\vec{\epsilon}$ using Eq. (3.16).

until Convergence criteria is reached.

Detect the optimal number of components M and K by eliminating the components with small mixing coefficients close to 0.

EM algorithm. The complete algorithm can be summarized in Algorithm 2. It is noteworthy that the proposed algorithm allows implicitly and simultaneously model selection with parameter estimation and feature selection. This is different from classic approaches which perform model selection using model selection rules, derived generally under asymptotical assumption and information theoretic reasoning, such as MML, MDL and AIC [11]. A major drawback of these traditional approaches is that they require the entire learning process to be repeated for different models (i.e. different values of M and K in our case).

3.3 Experimental Results

In this section, we shall illustrate our results with a collection of simulation studies involving both synthetic data and a real-life challenging application namely human action videos categorization. The goal of the synthetic data is to investigate the accuracy of the variational approach. The real application has two main goals. The first goal is to compare our approach which we refer to as varFsGD to the MML-based unsupervised feature selection approach (MMLFsGD) previously proposed in [60]. The second goal is to compare varFsGD with the GD mixture learned in a variational way without feature selection (we refer to this approach as varGD). We have also compared our results with the variational Gaussian mixture-based unsupervised feature selection approach (we shall refer to as varFsGau) proposed in [59]. In all our experiments, we initialize the number of components M and K with large values (15 and 10, respectively) with equal mixing coefficients,

Table 3.1: Parameters of the different generated data sets. N denotes the total number of elements, n_j denotes the number of elements in cluster j for the relevant features. $\alpha_{j1}, \beta_{j1}, \alpha_{j2}, \beta_{j2}, \alpha_{j3}, \beta_{j3}$ and π_j are the real parameters of the mixture models of relevant features. $\hat{\alpha}_{j1}, \hat{\beta}_{j1}, \hat{\alpha}_{j2}, \hat{\beta}_{j2}, \hat{\alpha}_{j3}, \hat{\beta}_{j3}$ and $\hat{\pi}_j$ are the estimated parameters from variational inference.

	n_j	j	α_{j1}	β_{j1}	α_{j2}	β_{j2}	α_{j3}	β_{j3}	π_j	$\hat{\alpha}_{j1}$	$\hat{\beta}_{j1}$	$\hat{\alpha}_{j2}$	$\hat{\beta}_{j2}$	$\hat{\alpha}_{j3}$	$\hat{\beta}_{j3}$	$\hat{\pi}_j$
Data set 1	300	1	30	15	20	40	33	18	0.33	27.94	14.32	18.65	41.27	32.13	17.52	0.32
($N = 900$)	300	2	25	33	30	50	14	62	0.33	23.71	31.15	28.16	48.88	13.57	59.93	0.34
	300	3	40	30	35	26	27	12	0.34	39.54	29.36	36.22	24.51	25.33	11.89	0.34
Data set 2	200	1	30	15	20	20	33	18	0.23	28.68	14.14	19.01	19.55	31.76	17.54	0.24
($N = 900$)	300	2	25	33	30	50	14	62	0.34	25.03	32.72	28.11	48.39	14.58	64.39	0.34
	400	3	40	30	19	21	15	10	0.43	35.57	26.34	18.73	20.58	15.77	9.81	0.42
Data set 3	800	1	45	55	62	47	54	39	0.53	46.01	57.86	60.15	45.29	51.04	41.68	0.54
($N = 1500$)	700	2	59	60	50	65	35	45	0.47	58.10	58.16	48.43	61.89	34.51	47.84	0.46
Data set 4	200	1	15	16	20	15	17	36	0.16	15.31	17.09	19.23	15.21	16.33	38.19	0.16
($N = 1200$)	200	2	18	35	10	25	20	13	0.16	18.95	37.17	10.15	23.94	22.18	12.57	0.15
	400	3	40	28	33	46	18	40	0.33	39.30	27.65	31.17	47.56	19.22	43.83	0.33
	400	4	30	44	25	40	35	22	0.35	30.24	45.79	23.61	38.39	33.37	24.15	0.36

and the feature saliency values are initialized at 0.5. In order to provide broad non-informative prior distributions, the initial value of u, p, g and s for the conjugate priors are set to 1, and v, q, h, t are set to 0.01.

3.3.1 Synthetic Data

First, the performance of the proposed varFsGD algorithm was evaluated in terms of estimation and selection, through quantitative analysis on four 11-dimensional (three relevant features and eight irrelevant features) synthetic data sets. The relevant features were generated in the transformed space from mixtures of Beta distributions with well-separated components, while irrelevant ones were from mixtures of overlapped components. Table 3.1 illustrates the real and estimated parameters of the distributions representing the relevant features for each data set using the proposed variational algorithm. According to this table, the parameters of the model, representing relevant features, and its mixing coefficients are accurately estimated. Although we do not show the estimated values of the parameters of the mixture models representing irrelevant features (the eight remaining features), accurate results were obtained by adopting the proposed algorithm as well.

The feature saliencies of all the 11 features for each generated data set are shown in Figure 3.1.

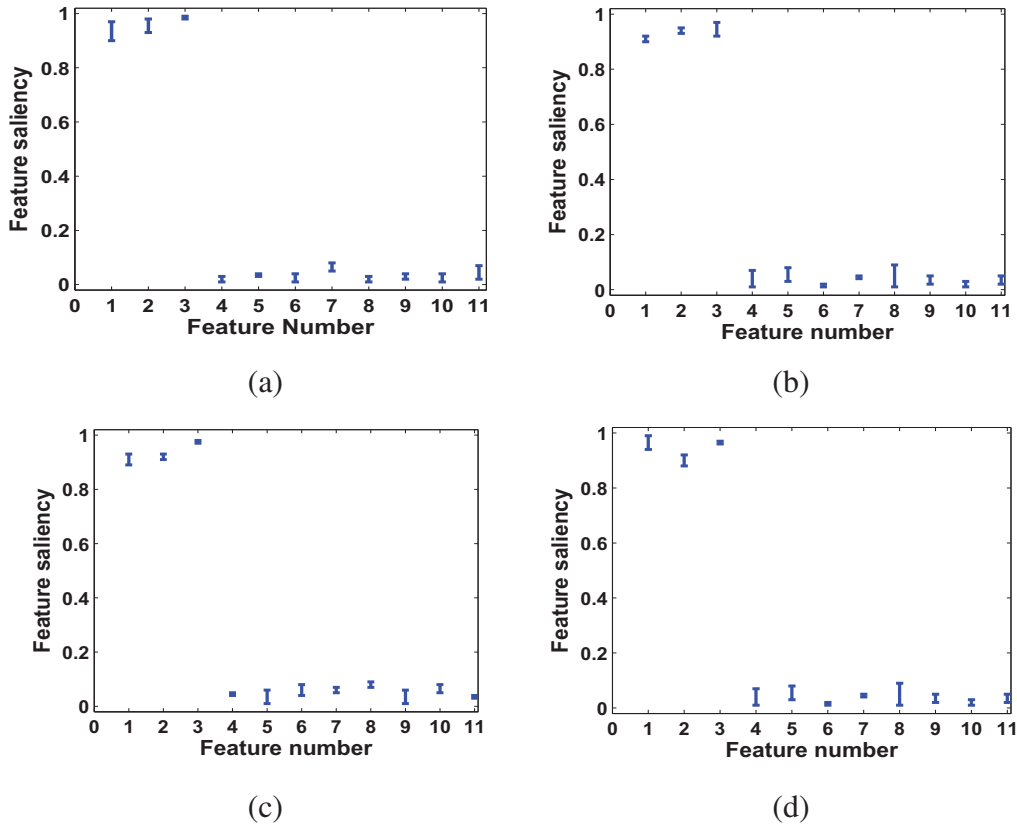


Figure 3.1: Feature saliency for synthetic data sets with one standard deviation over ten runs. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4.

It is clear that features 1, 2 and 3 have been assigned a high degree of relevance which is consistent with the ground-truth. Therefore, we can conclude that, for synthetic data sets, the proposed algorithm successfully detects the true number of components and correctly assigns the importance of features.

3.3.2 Human Action Videos Categorization

With the rapid development of digital technologies, the increase in the availability of multimedia data such as images and videos is tremendous. With thousands of videos on hand, grouping them according to their contents is highly important for a variety of visual tasks such as event analysis [97], video indexing, browsing and retrieval, and digital libraries organization [98]. How

to provide efficient videos categorization approaches has attracted many research efforts and has been addressed by several researchers in the past (see, for instance, [99–101]). Videos categorization remains, however, an extremely challenging task due to several typical scenarios such as unconstrained motions, cluttered scenes, moving backgrounds, object occlusions, non-stationary camera, geometric changes and deformation of objects and variations of illumination conditions and viewpoints. In this section, we present an unsupervised learning method, based on our variational algorithm, for categorizing human action videos. The performance of the proposed method is evaluated on a challenging video data set namely the KTH [102] human action data set.

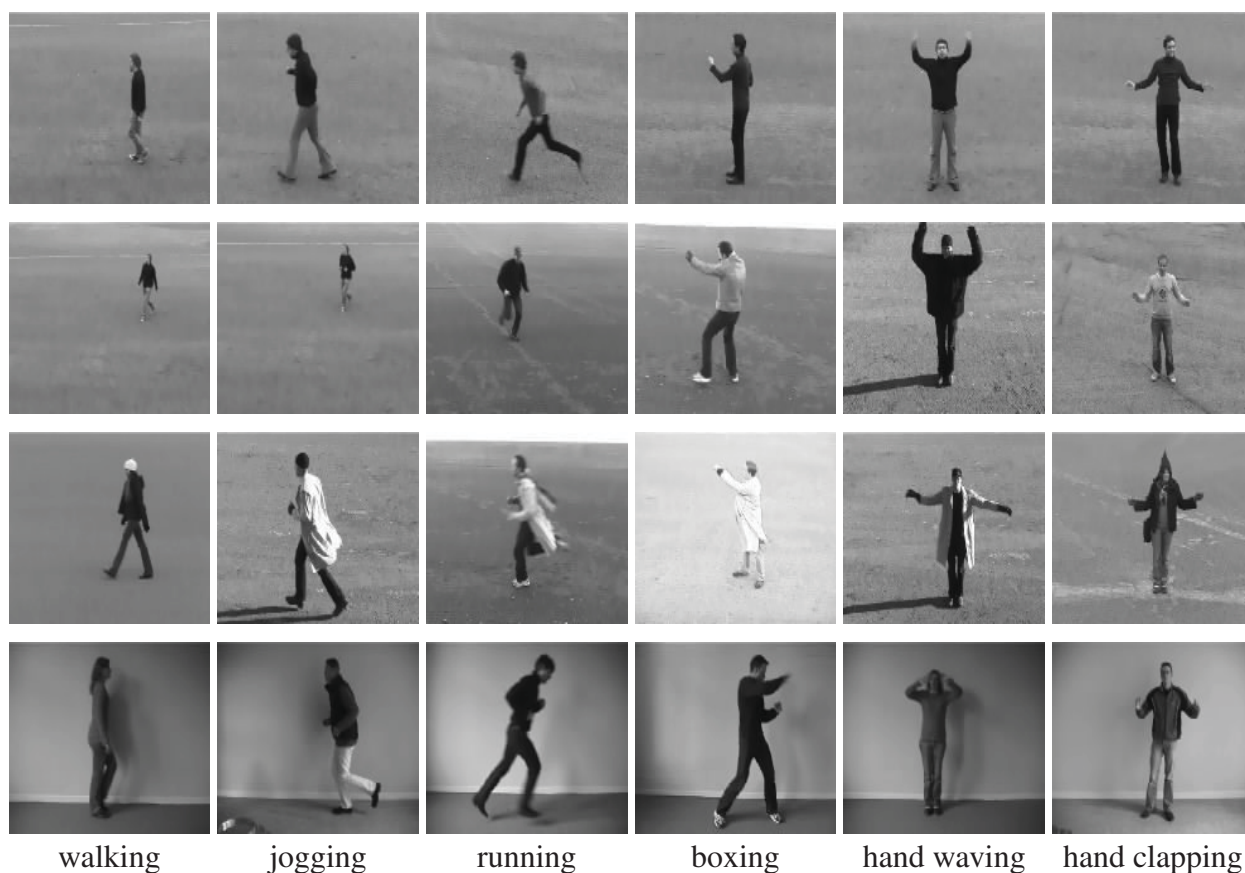


Figure 3.2: Examples of frames, representing different human actions in different scenarios, from video sequences in the KTH data set.

Experimental Methodology

Several studies have been conducted to provide models and visual features in order to consistently (i.e. regardless changes in viewpoint angles, position, distance, size, orientation, or deformation) categorize objects and visual scenes. These studies have shown that a good model is required, and it must be able to select relevant visual features to improve categorization performance [103, 104]. Recently several works have been based on the notion of visual vocabulary constructed via a quantization process, according to a coding rule such as K-Means, of local features (spatio-temporal features in the case of videos) extracted from a set of detected interest points (space-time interest points in the case of videos). This approach allows the representation of images and videos as histograms of visual words and have convincingly proven its effectiveness in several applications (see, for instance, [86]).

Our methodology for unsupervised videos categorization can be summarized as the following. First, local spatio-temporal features from each video sequence are extracted from their detected space-time interest points. Among many of the existing space-time interest points detectors and local spatio-temporal features [99, 105], we employ the space-time interest point detector proposed in [101]¹, which is actually a space-time extension of the well-known Harris operator, and histograms of optic flow (HoF) as proposed in [105]. Next, a visual vocabulary is constructed by quantizing these spatio-temporal features into visual words using K-means algorithm and each video is then represented as a frequency histogram over the visual words. Then, we apply the pLSA model [88] to the obtained histograms as done in [87] in the case of still images. As a result each video is represented now by a D -dimensional proportional vector where D is the number of latent aspects. Finally, we employ our varFsGD model as a classifier to categorize videos by assigning the video sequence to the group which has the highest posterior probability according to Bayes' decision rule.

¹We have also tested another popular feature detector namely the Cuboid detector proposed in [99]. However, we have not noticed a significant improvement according to our experiments.

KTH Human Action Data Set

The KTH human action data set is one of the largest available video sequences data sets of human actions [102]. It contains six types of human action classes including: walking, jogging, running, boxing, hand waving and hand clapping. Each action class is performed several times by 25 subjects in four different scenarios: outdoors (S1), outdoors with scale variation (S2), outdoors with different clothes (S3) and indoors (S4). This data set contains 2391 video sequences and all sequences were taken over homogenous backgrounds with a static camera with 25fps frame rate. All video samples were downsampled to the spatial resolution of 160×120 pixels and have a length of four seconds in average. Examples of frames from video sequences of each category are shown in Figure 3.2. In this experiment, we considered a training set composed of actions related to 16 subjects to construct the visual vocabulary, by setting the number of clusters in the K-Means algorithm (i.e. number of visual words) to 1000, as explained in the previous section. The pLSA model was applied by considering 40 aspects and each video in the database was then represented by a 40-dimensional vector of proportions. Last, the resulting vectors were clustered by our varFsGD model. The entire procedure was repeated 30 times for evaluating the performance of our approach. The optimal number of components was estimated as around 6 while the number of irrelevant Beta components was identified as $K = 2$. The confusion matrix for the KTH data set is shown in Figure 3.3. We note that, most of the confusion takes place between “walking” and “jogging”, “jogging” and “running”, as well as between “hand clapping” and “boxing”. This is due to the fact that similar actions contain similar types of local space-time events.

Table 3.2 shows the average classification accuracy and the average number of relevant components obtained by varFsGD, MMLFsGD, varGD and varFsGau. It clearly shows that our algorithm outperforms the other approaches for clustering KTH human action videos. For instance, the fact that the varFsGD performs better than the varFsGau is actually expected since videos are represented by vectors of proportions for which the GD mixture is one of the best modeling choices unlike the Gaussian mixture which implicitly assumes that the features vectors are Gaussian which is far from the case.

We have also tested the effect of different sizes of visual vocabulary on classification accuracy for varFsGD, MMLFsGD, varGD and varFsGau, as illustrated in Figure 3.4(a). As we can see,

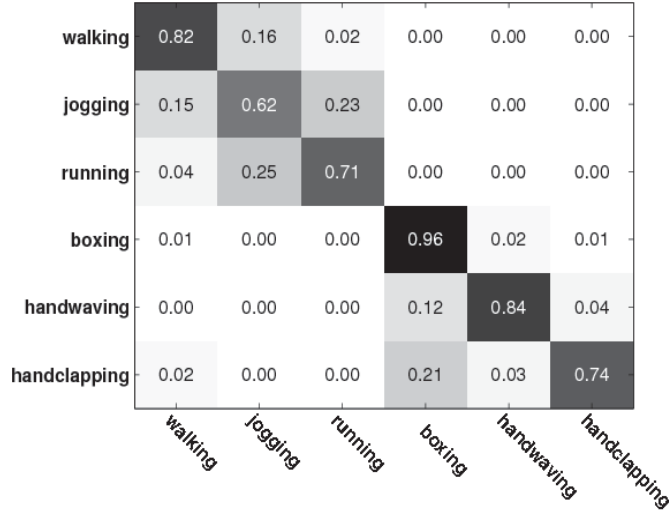
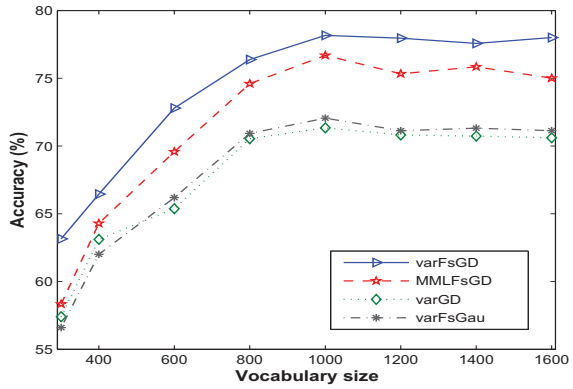


Figure 3.3: Confusion matrix for the KTH data set.

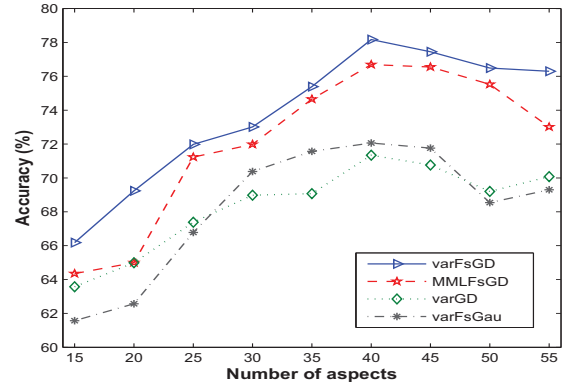
Table 3.2: The average classification accuracy and the number of components (\hat{M}) computed on the KTH data set using varFsGD, MMLFsGD, varGD and varFsGau over 30 random runs.

Algorithm	\hat{M}	Accuracy (%)
varFsGD	5.96	78.17
MMLFsGD	5.87	76.69
varGD	5.53	71.34
varFsGau	5.67	72.06

the classification rate peaks around 1000. The choice of the number of aspects also influences the accuracy of classification. As shown in Figure 3.4(b), the optimal accuracy can be obtained when the number of aspects is set to 40. These aspects may contribute with different degrees in discriminating among image categories. The corresponding feature saliency of the 40-dimensional aspects together with their standard deviations (error bars) can be viewed in Figure 3.5. As illustrated in this figure, the features have different relevance degrees and then contribute differently to clustering. For instance, there are seven features (features number 1, 8, 11, 14, 16, 22, 29) that have saliencies lower than 0.5, and then provide less contribution in clustering. This is because these aspects are associated to all categories and have less discrimination power. By contrast, eight



(a)



(b)

Figure 3.4: (a) Classification accuracy vs. vocabulary size for the KTH data set; (b) Classification accuracy vs. the number of aspects for the KTH data set.

features (features number 2, 10, 13, 25, 28, 33, 36 and 37) have high relevance degrees with feature saliencies greater than 0.9 which can be explained by the fact that these features are mainly associated with specific action categories.

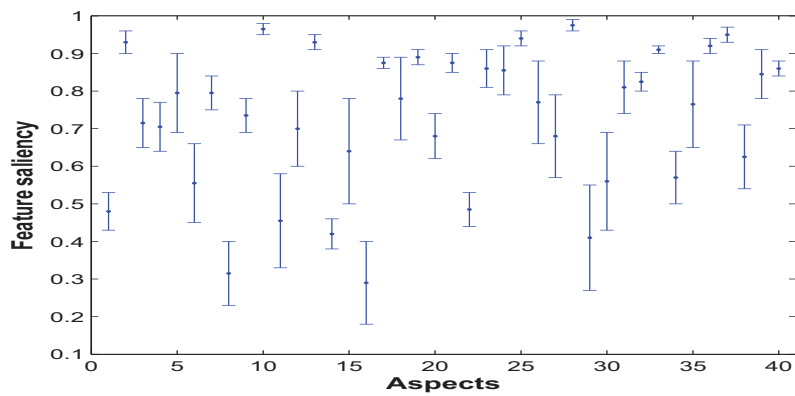


Figure 3.5: Feature saliencies of the different aspect features over 30 runs for the KTH data set.

Chapter 4

Variational Learning of a Dirichlet Process of Generalized Dirichlet Distributions for Simultaneous Clustering and Feature Selection

This chapter introduces a novel enhancement for unsupervised feature selection based on generalized Dirichlet mixture models. Our proposal is based on the extension of the finite mixture model previously developed in [60] to the infinite case, via the consideration of Dirichlet process mixtures, which can be viewed actually as a purely nonparametric model since the number of mixture components can increase as data are introduced. The infinite assumption is used to avoid problems related to model selection (i.e. determination of the number of clusters) and allows simultaneous separation of data into similar clusters and selection of relevant features. Our resulting model is learned within a principled variational Bayesian framework that we have developed. The experimental results reported for both synthetic data and real-world challenging applications involving image categorization, automatic semantic annotation and retrieval show the ability of our approach to provide accurate models by distinguishing between relevant and irrelevant features without over- or under-fitting the data.

4.1 The Infinite GD Mixture Model with Feature Selection

In this section, we describe our main unsupervised infinite feature selection model. We start by a brief overview of the finite GD mixture model. Then, the extension of this model to the infinite case and the integration of feature selection are proposed. Finally, we present the conjugate priors that we will consider for the resulting model learning.

4.1.1 The Finite GD Mixture Model

Consider a random vector $\vec{Y} = (Y_1, \dots, Y_D)$, drawn from a finite mixture of generalized Dirichlet (GD) Distributions with M components [106] as

$$p(\vec{Y}|\vec{\pi}, \vec{\alpha}, \vec{\beta}) = \sum_{j=1}^M \pi_j \text{GD}(\vec{Y}|\vec{\alpha}_j, \vec{\beta}_j) \quad (4.1)$$

where $\vec{\alpha} = \{\vec{\alpha}_1, \dots, \vec{\alpha}_M\}$, $\vec{\beta} = \{\vec{\beta}_1, \dots, \vec{\beta}_M\}$, $\vec{\alpha}_j$ and $\vec{\beta}_j$ are the parameters of the GD distribution representing component j with $\vec{\alpha}_j = \{\alpha_{j1}, \dots, \alpha_{jD}\}$ and $\vec{\beta}_j = \{\beta_{j1}, \dots, \beta_{jD}\}$, and $\vec{\pi} = \{\pi_1, \dots, \pi_M\}$ represents the mixing coefficients which are positive and sum to one. A GD distribution is defined as

$$\text{GD}(\vec{Y}|\vec{\alpha}_j, \vec{\beta}_j) = \prod_{l=1}^D \frac{\Gamma(\alpha_{jl} + \beta_{jl})}{\Gamma(\alpha_{jl})\Gamma(\beta_{jl})} Y_l^{\alpha_{jl}-1} \left(1 - \sum_{k=1}^l Y_k\right)^{\gamma_{jl}} \quad (4.2)$$

where $\sum_{l=1}^D Y_l < 1$ and $0 < Y_l < 1$ for $l = 1, \dots, D$, $\alpha_{jl} > 0$, $\beta_{jl} > 0$, $\gamma_{jl} = \beta_{jl} - \alpha_{j,l+1} - \beta_{j,l+1}$ for $l = 1, \dots, D-1$, and $\gamma_{jD} = \beta_{jD} - 1$.

Now, let us consider a set of N independent identically distributed vectors $\mathcal{Y} = (\vec{Y}_1, \dots, \vec{Y}_N)$ assumed to arise from a finite GD mixture. Following the Bayes' theorem, the probability that vector i is in cluster j conditional on having observed \vec{Y}_i (also known as *responsibilities*) can be written as

$$p(j|\vec{Y}_i) \propto \pi_j \text{GD}(\vec{Y}_i|\vec{\alpha}_j, \vec{\beta}_j) \quad (4.3)$$

In this work, we exploit an interesting mathematical property of the GD distribution previously discussed in [60, 106] to redefine the responsibilities as

$$p(j|\vec{Y}_i) \propto \pi_j \prod_{l=1}^D \text{Beta}(X_{il}|\alpha_{jl}, \beta_{jl}) \quad (4.4)$$

where $X_{i1} = Y_{i1}$ and $X_{il} = Y_{il}/(1 - \sum_{k=1}^{l-1} Y_{ik})$ for $l > 1$ and $\text{Beta}(X_{il}|\alpha_{jl}, \beta_{jl})$ is a Beta distribution defined with parameters $(\alpha_{jl}, \beta_{jl})$. Thus, the clustering structure for a finite GD mixture model underlying data set \mathcal{Y} can be represented by a new data set $\mathcal{X} = (\vec{X}_1, \dots, \vec{X}_N)$ using the following mixture model with conditionally independent features

$$p(\vec{X}_i|\vec{\pi}, \vec{\alpha}, \vec{\beta}) = \sum_{j=1}^M \pi_j \prod_{l=1}^D \text{Beta}(X_{il}|\alpha_{jl}, \beta_{jl}) \quad (4.5)$$

4.1.2 Infinite GD Mixture Model With Feature Selection

A conventional finite mixture model can be extended to have an infinite number of components using the Dirichlet process mixture model with a stick-breaking representation. The Dirichlet process (DP) [107, 108] is a stochastic process whose sample paths are probability measures with probability one. It can be considered as a distribution over distributions. The infinite GD mixture model with feature selection proposed in this chapter is constructed using the DP with a stick-breaking representation. Stick-breaking representation is an intuitive and straightforward constructive definition of the DP [109–111]. It is defined as follows: given a random distribution G , it is distributed according to a DP $G \sim DP(\psi, H)$ if the following conditions are satisfied:

$$\lambda_j \sim \text{Beta}(1, \psi), \quad \Omega_j \sim H, \quad \pi_j = \lambda_j \prod_{s=1}^{j-1} (1 - \lambda_s), \quad G = \sum_{j=1}^{\infty} \pi_j \delta_{\Omega_j} \quad (4.6)$$

where δ_{Ω_j} denotes the Dirac delta measure centered at Ω_j , and ψ is a positive real number. The mixing weights π_j are obtained by recursively breaking an unit length stick into an infinite number of pieces.

Assuming now that the observed data set is generated from a GD mixture model with a countably infinite number of components. Thus, Eq. (4.5) can be rewritten as

$$p(\vec{X}_i | \vec{\pi}, \vec{\alpha}, \vec{\beta}) = \sum_{j=1}^{\infty} \pi_j \prod_{l=1}^D \text{Beta}(X_{il} | \alpha_{jl}, \beta_{jl}). \quad (4.7)$$

Then, for each vector \vec{X}_i , we introduce a binary latent variable $\vec{Z}_i = (Z_{i1}, Z_{i2}, \dots)$, such $Z_{ij} \in \{0, 1\}$ and $Z_{ij} = 1$ if \vec{X}_i belongs to component j and 0, otherwise. Therefore, the likelihood function of the infinite GD mixtures with latent variables, which is actually the conditional distribution of data set \mathcal{X} given the class labels $\mathcal{Z} = (\vec{Z}_1, \dots, \vec{Z}_N)$ can be written as

$$p(\mathcal{X} | \mathcal{Z}, \vec{\alpha}, \vec{\beta}) = \prod_{i=1}^N \prod_{j=1}^{\infty} \left(\prod_{l=1}^D \text{Beta}(X_{il} | \alpha_{jl}, \beta_{jl}) \right)^{Z_{ij}} \quad (4.8)$$

According to Chapter 3, we know that some of the features in a high-dimensional data set may be irrelevant and not contribute to the clustering process. In order to take this fact into account the authors in [54] have supposed 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. This idea has been extended in [60] where the irrelevant features is modeled as a finite mixture of distributions rather than a usual single distribution. In this work, we go a step further by modeling the irrelevant features with an infinite mixture model in order to bypass the difficulty of estimating the appropriate number of components for the mixture model representing irrelevant features. Therefore, each feature X_{il} can be approximated as

$$p(X_{il}) \simeq \left(\text{Beta}(X_{il}|\alpha_{jl}, \beta_{jl}) \right)^{\phi_{il}} \left(\prod_{k=1}^{\infty} \text{Beta}(X_{il}|\sigma_{kl}, \tau_{kl})^{W_{ikl}} \right)^{1-\phi_{il}} \quad (4.9)$$

where W_{ikl} is a binary variable such that $W_{ikl} = 1$ if X_{il} comes from the k th component of the infinite Beta mixture for the irrelevant features. ϕ_{il} is a binary latent variable, such that $\phi_{il} = 1$ indicates that feature l is relevant and follows a Beta distribution $\text{Beta}(X_{il}|\alpha_{jl}, \beta_{jl})$, and $\phi_{il} = 0$ denotes that feature l is irrelevant and supposed to follow an infinite mixture of Beta distributions independent from the class labels:

$$p(X_{il}) = \sum_{k=1}^{\infty} \eta_k \text{Beta}(X_{il}|\sigma_{kl}, \tau_{kl}) \quad (4.10)$$

where η_k denotes the mixing probability and also implies the prior probability that X_{il} is generated from the k th component of the infinite Beta mixture representing irrelevant features.

Thus, we can write the likelihood of the observed data set \mathcal{X} following the infinite GD mixture model with feature selection as

$$p(\mathcal{X}|\mathcal{Z}, \mathcal{W}, \vec{\phi}, \vec{\alpha}, \vec{\beta}, \vec{\sigma}, \vec{\tau}) = \prod_{i=1}^N \prod_{j=1}^{\infty} \left[\prod_{l=1}^D \text{Beta}(X_{il}|\alpha_{jl}, \beta_{jl})^{\phi_{il}} \times \left(\prod_{k=1}^{\infty} \text{Beta}(X_{il}|\sigma_{kl}, \tau_{kl})^{W_{ikl}} \right)^{1-\phi_{il}} \right]^{Z_{ij}} \quad (4.11)$$

where $\mathcal{W} = (\vec{W}_1, \dots, \vec{W}_N)$ with $\vec{W}_i = (\vec{W}_{i1}, \vec{W}_{i2}, \dots)$ and $\vec{W}_{ik} = (W_{ik1}, \dots, W_{ikD})$. $\vec{\phi} = (\vec{\phi}_1, \dots, \vec{\phi}_N)$ contains elements $\vec{\phi}_i = (\phi_{i1}, \dots, \phi_{iD})$. $\vec{\sigma} = (\vec{\sigma}_1, \vec{\sigma}_2, \dots)$ and $\vec{\tau} = (\vec{\tau}_1, \vec{\tau}_2, \dots)$ are the parameters of the Beta mixture representing irrelevant features which comprise elements $\vec{\sigma}_k = (\sigma_{k1}, \dots, \sigma_{kD})$ and $\vec{\tau}_k = (\tau_{k1}, \dots, \tau_{kD})$, respectively.

4.1.3 Prior Distributions of The Proposed Model

We shall follow the variational inference framework for learning our model. Thus, each unknown parameter is given a prior distribution. Since the analysis is considerably simplified if we exploit conjugate prior distributions, conjugate priors are therefore chosen for the unknown random variables $\mathcal{Z}, \mathcal{W}, \vec{\phi}, \vec{\alpha}, \vec{\beta}, \vec{\sigma}$ and $\vec{\tau}$. The prior distributions of \mathcal{Z} and \mathcal{W} given the mixing coefficients $\vec{\pi}$ and $\vec{\eta}$ can be specified as

$$p(\mathcal{Z}|\vec{\pi}) = \prod_{i=1}^N \prod_{j=1}^{\infty} \pi_j^{Z_{ij}} \quad (4.12)$$

$$p(\mathcal{W}|\vec{\eta}) = \prod_{i=1}^N \prod_{k=1}^{\infty} \prod_{l=1}^D \eta_k^{W_{ikl}} \quad (4.13)$$

According to the stick-breaking construction of DP as stated in Eq. (4.6), $\vec{\pi}$ is a function of $\vec{\lambda}$. We rewrite it here for the sake of clarity

$$\pi_j = \lambda_j \prod_{s=1}^{j-1} (1 - \lambda_s) \quad (4.14)$$

Similarly, $\vec{\eta}$ can be defined as a function of $\vec{\gamma}$, such that

$$\eta_k = \gamma_k \prod_{s=1}^{k-1} (1 - \gamma_s) \quad (4.15)$$

Therefore, we can rewrite Eq. (4.12) and Eq. (4.13) as

$$p(\mathcal{Z}|\vec{\lambda}) = \prod_{i=1}^N \prod_{j=1}^{\infty} [\lambda_j \prod_{s=1}^{j-1} (1 - \lambda_s)]^{Z_{ij}} \quad (4.16)$$

$$p(\mathcal{W}|\vec{\gamma}) = \prod_{i=1}^N \prod_{k=1}^{\infty} \prod_{l=1}^D [\gamma_k \prod_{s=1}^{k-1} (1 - \gamma_s)]^{W_{ikl}} \quad (4.17)$$

where $\vec{\lambda} = (\lambda_1, \lambda_2, \dots)$ and $\vec{\gamma} = (\gamma_1, \gamma_2, \dots)$. The prior distributions of $\vec{\lambda}$ and $\vec{\gamma}$ follow the specific Beta distribution given in Eq. (4.6) as

$$p(\vec{\lambda}|\vec{\psi}) = \prod_{j=1}^{\infty} \text{Beta}(1, \psi_j) = \prod_{j=1}^{\infty} \psi_j (1 - \lambda_j)^{\psi_j - 1} \quad (4.18)$$

$$p(\vec{\gamma}|\vec{\varphi}) = \prod_{k=1}^{\infty} \text{Beta}(1, \varphi_k) = \prod_{k=1}^{\infty} \varphi_k (1 - \gamma_k)^{\varphi_k - 1} \quad (4.19)$$

To add more flexibility, another layer is added to the Bayesian hierarchy by introducing prior distributions over the hyperparameters $\vec{\psi} = (\psi_1, \psi_2, \dots)$ and $\vec{\varphi} = (\varphi_1, \varphi_2, \dots)$. Motivated by the fact that the Gamma distribution is conjugate to the stick lengths [112], Gamma priors are placed over $\vec{\psi}$ and $\vec{\varphi}$ as

$$p(\vec{\psi}) = \mathcal{G}(\vec{\psi}|\vec{a}, \vec{b}) = \prod_{j=1}^{\infty} \frac{b_j^{a_j}}{\Gamma(a_j)} \psi_j^{a_j-1} e^{-b_j \psi_j} \quad (4.20)$$

$$p(\vec{\varphi}) = \mathcal{G}(\vec{\varphi}|\vec{c}, \vec{d}) = \prod_{k=1}^{\infty} \frac{d_k^{c_k}}{\Gamma(c_k)} \varphi_k^{c_k-1} e^{-d_k \varphi_k} \quad (4.21)$$

where hyperparameters $\vec{a} = (a_1, a_2, \dots)$, $\vec{b} = (b_1, b_2, \dots)$, $\vec{c} = (c_1, c_2, \dots)$ and $\vec{d} = (d_1, d_2, \dots)$ are subject to the constraints $a_j > 0$, $b_j > 0$, $c_k > 0$ and $d_k > 0$ to ensure that these two prior distributions can be normalized. The prior distribution for the feature relevance indicator variable $\vec{\phi}$ is defined as

$$p(\vec{\phi}|\vec{\epsilon}) = \prod_{i=1}^N \prod_{l=1}^D \epsilon_{l1}^{\phi_{il}} \epsilon_{l2}^{1-\phi_{il}} \quad (4.22)$$

where each ϕ_{il} is a Bernoulli variable such that $p(\phi_{il} = 1) = \epsilon_{l1}$ and $p(\phi_{il} = 0) = \epsilon_{l2}$. The vector $\vec{\epsilon} = (\vec{\epsilon}_1, \dots, \vec{\epsilon}_D)$ represents the features saliencies (i.e. the probabilities that the features are relevant) such that $\vec{\epsilon}_l = (\epsilon_{l1}, \epsilon_{l2})$ and $\epsilon_{l1} + \epsilon_{l2} = 1$. Furthermore, a Dirichlet distribution is chosen over $\vec{\epsilon}$ as [113]

$$p(\vec{\epsilon}) = \prod_{l=1}^D \text{Dir}(\vec{\epsilon}_l|\vec{\xi}) = \prod_{l=1}^D \frac{\Gamma(\xi_1 + \xi_2)}{\Gamma(\xi_1)\Gamma(\xi_2)} \epsilon_{l1}^{\xi_1-1} \epsilon_{l2}^{\xi_2-1} \quad (4.23)$$

where the hyperparameter $\vec{\xi} = (\xi_1, \xi_2)$ is subject to the constraint $(\xi_1, \xi_2) > 0$ in order to ensure that the distribution can be normalized.

Next, we need to define the prior distributions for parameters $\vec{\alpha}$, $\vec{\beta}$, $\vec{\sigma}$ and $\vec{\tau}$ of Beta distributions. Although Beta distribution belongs to the exponential family and has a formal conjugate prior, it is analytically intractable and cannot be used within a variational framework as shown for instance in [78]. Thus, the Gamma distribution is adopted to approximate the conjugate prior, as suggested

in [78], by assuming that parameters of Beta distributions are statistically independent:

$$p(\vec{\alpha}) = \mathcal{G}(\vec{\alpha} | \vec{u}, \vec{v}) = \prod_{j=1}^{\infty} \prod_{l=1}^D \frac{v_{jl}^{u_{jl}}}{\Gamma(u_{jl})} \alpha_{jl}^{u_{jl}-1} e^{-v_{jl} \alpha_{jl}} \quad (4.24)$$

$$p(\vec{\beta}) = \mathcal{G}(\vec{\beta} | \vec{p}, \vec{q}) = \prod_{j=1}^{\infty} \prod_{l=1}^D \frac{q_{jl}^{p_{jl}}}{\Gamma(p_{jl})} \beta_{jl}^{p_{jl}-1} e^{-q_{jl} \beta_{jl}} \quad (4.25)$$

$$p(\vec{\sigma}) = \mathcal{G}(\vec{\sigma} | \vec{g}, \vec{h}) = \prod_{k=1}^{\infty} \prod_{l=1}^D \frac{h_{kl}^{g_{kl}}}{\Gamma(g_{kl})} \sigma_{kl}^{g_{kl}-1} e^{-h_{kl} \sigma_{kl}} \quad (4.26)$$

$$p(\vec{\tau}) = \mathcal{G}(\vec{\tau} | \vec{s}, \vec{t}) = \prod_{k=1}^{\infty} \prod_{l=1}^D \frac{t_{kl}^{s_{kl}}}{\Gamma(s_{kl})} \tau_{kl}^{s_{kl}-1} e^{-t_{kl} \tau_{kl}} \quad (4.27)$$

where all the hyperparameters $\vec{u} = \{u_{jl}\}$, $\vec{v} = \{v_{jl}\}$, $\vec{p} = \{p_{jl}\}$, $\vec{q} = \{q_{jl}\}$, $\vec{g} = \{g_{kl}\}$, $\vec{h} = \{h_{kl}\}$, $\vec{s} = \{s_{kl}\}$ and $\vec{t} = \{t_{kl}\}$ of the above conjugate priors are positive.

A directed graphical representation of the infinite GD mixture model with feature selection is illustrated in Figure 4.1.

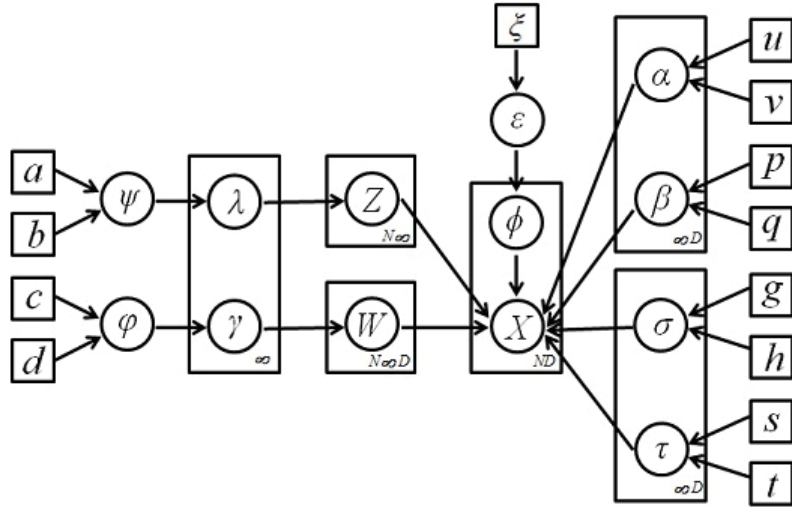


Figure 4.1: Graphical model representation of the infinite GD mixture model with feature selection. Symbols in circles denote random variables; otherwise, they denote model parameters. Plates indicate repetition (with the number of repetitions in the lower right), and arcs describe conditional dependencies between variables.

4.2 Variational Inference

In this section, a variational framework for learning the infinite GD mixture model with feature selection is proposed. In our work, we define $\Theta = \{\mathcal{Z}, \mathcal{W}, \vec{\phi}, \vec{\alpha}, \vec{\beta}, \vec{\sigma}, \vec{\tau}, \vec{\lambda}, \vec{\psi}, \vec{\gamma}, \vec{\varphi}, \vec{\epsilon}\}$ as the set of unknown random variables. The main idea in variational learning is to find an approximation $Q(\Theta)$ for the true posterior distribution $p(\Theta|\mathcal{X})$ [40].

Motivated from the work in [112], we truncate the stick-breaking representation for the infinite GD mixture model at a value of M as

$$\lambda_M = 1, \quad \pi_j = 0 \text{ when } j > M, \quad \sum_{j=1}^M \pi_j = 1 \quad (4.28)$$

Moreover, the infinite Beta mixture model for the irrelevant features is truncated at a value of K such that

$$\gamma_K = 1, \quad \eta_k = 0 \text{ when } k > K, \quad \sum_{k=1}^K \eta_k = 1 \quad (4.29)$$

Please notice that, the truncation levels M and K are variational parameters which can be freely initialized and will be optimized automatically during the learning process.

By employing the factorization assumption and the truncated stick-breaking representation for the proposed model, we then obtain

$$\begin{aligned} Q(\Theta) = & \left[\prod_{i=1}^N \prod_{j=1}^M Q(Z_{ij}) \right] \left[\prod_{j=1}^M Q(\lambda_j) Q(\psi_j) \right] \left[\prod_{i=1}^N \prod_{k=1}^K \prod_{l=1}^D Q(W_{ikl}) \right] \\ & \times \left[\prod_{k=1}^K Q(\gamma_k) Q(\varphi_k) \right] \left[\prod_{i=1}^N \prod_{l=1}^D Q(\phi_{il}) \right] \left[\prod_{l=1}^D Q(\vec{\epsilon}_l) \right] \\ & \times \left[\prod_{j=1}^M \prod_{l=1}^D Q(\alpha_{jl}) Q(\beta_{jl}) \right] \left[\prod_{k=1}^K \prod_{l=1}^D Q(\sigma_{kl}) Q(\tau_{kl}) \right] \end{aligned} \quad (4.30)$$

In this work, the general expression for the optimal solution of each variational factor is given by

$$Q_s(\Theta_s) = \frac{\exp\langle \ln p(\mathcal{X}, \Theta) \rangle_{i \neq s}}{\int \exp\langle \ln p(\mathcal{X}, \Theta) \rangle_{i \neq s} d\Theta} \quad (4.31)$$

where $\langle \cdot \rangle_{i \neq s}$ denotes an expectation with respect to all the distributions $Q_i(\Theta_i)$ except for $i = s$.

By applying Eq. (4.31) to each factor of the variational posterior, we then acquire the following optimal solutions

$$Q(\mathcal{Z}) = \prod_{i=1}^N \prod_{j=1}^M r_{ij}^{Z_{ij}}, \quad Q(\vec{\lambda}) = \prod_{j=1}^M \text{Beta}(\lambda_j | \theta_j, \vartheta_j) \quad (4.32)$$

$$Q(\vec{\psi}) = \prod_{j=1}^M \mathcal{G}(\psi_j | a_j^*, b_j^*), \quad Q(\mathcal{W}) = \prod_{i=1}^N \prod_{k=1}^K \prod_{l=1}^D (m_{ikl}^{W_{ikl}}) \quad (4.33)$$

$$Q(\vec{\gamma}) = \prod_{k=1}^K \text{Beta}(\gamma_k | \rho_k, \varpi_k), \quad Q(\vec{\varphi}) = \prod_{k=1}^K \mathcal{G}(\varphi_k | c_k^*, d_k^*) \quad (4.34)$$

$$Q(\vec{\phi}) = \prod_{i=1}^N \prod_{l=1}^D f_{il}^{\phi_{il}} (1 - f_{il})^{(1-\phi_{il})}, \quad Q(\vec{\epsilon}) = \prod_{l=1}^D \text{Dir}(\vec{\epsilon}_l | \xi^*) \quad (4.35)$$

$$Q(\vec{\alpha}) = \prod_{j=1}^M \prod_{l=1}^D \mathcal{G}(\alpha_{jl} | u_{jl}^*, v_{jl}^*), \quad Q(\vec{\beta}) = \prod_{j=1}^M \prod_{l=1}^D \mathcal{G}(\beta_{jl} | p_{jl}^*, q_{jl}^*) \quad (4.36)$$

$$Q(\vec{\sigma}) = \prod_{k=1}^K \prod_{l=1}^D \mathcal{G}(\sigma_{kl} | g_{kl}^*, h_{kl}^*), \quad Q(\vec{\tau}) = \prod_{k=1}^K \prod_{l=1}^D \mathcal{G}(\tau_{kl} | s_{kl}^*, t_{kl}^*) \quad (4.37)$$

where we have defined

$$r_{ij} = \frac{\tilde{r}_{ij}}{\sum_{j=1}^M \tilde{r}_{ij}}, \quad f_{il} = \frac{f_{il}^{(\phi_{il})}}{f_{il}^{(\phi_{il})} + f_{il}^{(1-\phi_{il})}}, \quad m_{ikl} = \frac{\tilde{m}_{ikl}}{\sum_{k=1}^K \tilde{m}_{ikl}} \quad (4.38)$$

$$\tilde{r}_{ij} = \exp\left\{ \sum_{l=1}^D \langle \phi_{il} \rangle [\tilde{\mathcal{R}}_{jl} + (\bar{\alpha}_{jl} - 1) \ln X_{il} + (\bar{\beta}_{jl} - 1) \ln(1 - X_{il})] + \langle \ln \lambda_j \rangle + \sum_{s=1}^{j-1} \langle \ln(1 - \lambda_s) \rangle \right\} \quad (4.39)$$

$$\tilde{m}_{ikl} = \exp\left\{ (1 - \phi_{il}) [\tilde{\mathcal{F}}_{kl} + (\bar{\sigma}_{kl} - 1) \ln X_{il} + (\bar{\tau}_{kl} - 1) \ln(1 - X_{il})] + \langle \ln \gamma_k \rangle + \sum_{s=1}^{k-1} \langle \ln(1 - \gamma_s) \rangle \right\} \quad (4.40)$$

$$f_{il}^{(\phi_{il})} = \exp\left\{ \sum_{j=1}^M \langle Z_{ij} \rangle [\tilde{\mathcal{R}}_{jl} + (\bar{\alpha}_{jl} - 1) \ln X_{il} + (\bar{\beta}_{jl} - 1) \ln(1 - X_{il})] + \langle \ln \epsilon_{l_1} \rangle \right\} \quad (4.41)$$

$$f_{il}^{(1-\phi_{il})} = \exp\left\{ \sum_{k=1}^K \langle W_{ikl} \rangle [\tilde{\mathcal{F}}_{kl} + (\bar{\sigma}_{kl} - 1) \ln X_{il} + (\bar{\tau}_{kl} - 1) \ln(1 - X_{il})] + \langle \ln \epsilon_{l_2} \rangle \right\} \quad (4.42)$$

$$u_{jl}^* = u_{jl} + \sum_{i=1}^N \langle Z_{ij} \rangle \langle \phi_{il} \rangle \bar{\alpha}_{jl} [\psi(\bar{\alpha}_{jl} + \bar{\beta}_{jl}) - \psi(\bar{\alpha}_{jl}) + \bar{\beta}_{jl} \psi'(\bar{\alpha}_{jl} + \bar{\beta}_{jl}) (\langle \ln \beta_{jl} \rangle - \ln \bar{\beta}_{jl})] \quad (4.43)$$

$$p_{jl}^* = p_{jl} + \sum_{i=1}^N \langle Z_{ij} \rangle \langle \phi_{il} \rangle \bar{\beta}_{jl} [\psi(\bar{\alpha}_{jl} + \bar{\beta}_{jl}) - \psi(\bar{\beta}_{jl}) + \bar{\alpha}_{jl} \psi'(\bar{\alpha}_{jl} + \bar{\beta}_{jl}) (\langle \ln \alpha_{jl} \rangle - \ln \bar{\alpha}_{jl})] \quad (4.44)$$

$$g_{kl}^* = g_{kl} + \sum_{i=1}^N \langle 1 - \phi_{il} \rangle \langle W_{ikl} \rangle \bar{\sigma}_{kl} [\psi(\bar{\sigma}_{kl} + \bar{\tau}_{kl}) - \psi(\bar{\sigma}_{kl}) + \bar{\tau}_{kl} \psi'(\bar{\sigma}_{kl} + \bar{\tau}_{kl}) (\langle \ln \tau_{kl} \rangle - \ln \bar{\tau}_{kl})] \quad (4.45)$$

$$s_{kl}^* = s_{kl} + \sum_{i=1}^N \langle 1 - \phi_{il} \rangle \langle W_{ikl} \rangle \bar{\tau}_{kl} [\psi(\bar{\sigma}_{kl} + \bar{\tau}_{kl}) - \psi(\bar{\tau}_{kl}) + \bar{\sigma}_{kl} \psi'(\bar{\sigma}_{kl} + \bar{\tau}_{kl}) (\langle \ln \sigma_{kl} \rangle - \ln \bar{\sigma}_{kl})] \quad (4.46)$$

$$v_{jl}^* = v_{jl} - \sum_{i=1}^N \langle Z_{ij} \rangle \langle \phi_{il} \rangle \ln X_{il}, \quad q_{jl}^* = q_{jl} - \sum_{i=1}^N \langle Z_{ij} \rangle \langle \phi_{il} \rangle \ln(1 - X_{il}) \quad (4.47)$$

$$h_{kl}^* = h_{kl} - \sum_{i=1}^N \langle 1 - \phi_{il} \rangle \langle W_{ikl} \rangle \ln X_{il}, \quad t_{kl}^* = t_{kl} - \sum_{i=1}^N \langle 1 - \phi_{il} \rangle \langle W_{ikl} \rangle \ln(1 - X_{il}) \quad (4.48)$$

$$\theta_j = 1 + \sum_{i=1}^N \langle Z_{ij} \rangle, \quad \vartheta_j = \langle \psi_j \rangle + \sum_{i=1}^N \sum_{s=j+1}^M \langle Z_{is} \rangle, \quad a_j^* = a_j + 1 \quad (4.49)$$

$$b_j^* = b_j - \langle \ln(1 - \lambda_j) \rangle, \quad \rho_k = 1 + \sum_{i=1}^N \sum_{l=1}^D \langle W_{ikl} \rangle, \quad c_k^* = c_k + 1 \quad (4.50)$$

$$\varpi_k = \langle \varphi_k \rangle + \sum_{i=1}^N \sum_{s=k+1}^K \sum_{l=1}^D \langle W_{isl} \rangle, \quad d_k^* = d_k - \langle \ln(1 - \gamma_k) \rangle \quad (4.51)$$

$$\xi_1^* = \xi_1 + \sum_{i=1}^N \langle \phi_{il} \rangle, \quad \xi_2^* = \xi_2 + \sum_{i=1}^N \langle 1 - \phi_{il} \rangle \quad (4.52)$$

where $\psi(\cdot)$ is the digamma function and defined as: $\psi(a) = d \ln \Gamma(a) / da$. $\tilde{\mathcal{R}}$ and $\tilde{\mathcal{F}}$ are the lower bound approximations of $\mathcal{R} = \langle \ln \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \rangle$ and $\mathcal{F} = \langle \ln \frac{\Gamma(\lambda+\tau)}{\Gamma(\lambda)\Gamma(\tau)} \rangle$, respectively. The expected values in the above formulas are given by

$$\bar{\alpha}_{jl} = \frac{u_{jl}^*}{v_{jl}^*}, \quad \bar{\beta}_{jl} = \frac{p_{jl}^*}{q_{jl}^*}, \quad \bar{\sigma}_{kl} = \frac{g_{kl}^*}{h_{kl}^*}, \quad \bar{\tau}_{kl} = \frac{s_{kl}^*}{t_{kl}^*} \quad (4.53)$$

$$\langle \psi_j \rangle = \frac{a_j^*}{b_j^*}, \quad \langle \varphi_k \rangle = \frac{c_k^*}{d_k^*}, \quad \langle Z_{ij} \rangle = r_{ij}, \quad \langle W_{ikl} \rangle = m_{ikl} \quad (4.54)$$

$$\langle \phi_{il} \rangle = f_{il}, \quad \langle 1 - \phi_{il} \rangle = 1 - f_{il}, \quad \langle \ln \alpha \rangle = \psi(u^*) - \ln v^* \quad (4.55)$$

$$\langle \ln \beta \rangle = \psi(p^*) - \ln q^*, \quad \langle \ln \sigma \rangle = \psi(g^*) - \ln h^*, \quad \langle \ln \tau \rangle = \psi(s^*) - \ln t^* \quad (4.56)$$

$$\langle \ln \lambda_j \rangle = \psi(\theta_j) - \psi(\theta_j + \vartheta_j), \quad \langle \ln(1 - \lambda_j) \rangle = \psi(\vartheta_j) - \psi(\theta_j + \vartheta_j) \quad (4.57)$$

$$\langle \ln \gamma_k \rangle = \psi(\rho_k) - \psi(\rho_k + \varpi_k), \quad \langle \ln(1 - \gamma_k) \rangle = \psi(\varpi_k) - \psi(\rho_k + \varpi_k) \quad (4.58)$$

$$\langle \ln \epsilon_{l_1} \rangle = \psi(\xi_1^*) - \psi(\xi_1^* + \xi_2^*), \quad \langle \ln \epsilon_{l_2} \rangle = \psi(\xi_2^*) - \psi(\xi_1^* + \xi_2^*) \quad (4.59)$$

The complete algorithm can be summarized in Algorithm 3.

Algorithm 3 Variational learning of infinite GD mixtures with feature selection

Choose the initial truncation levels M and K .

Initialize the values for hyper-parameters $u_{jl}, v_{jl}, p_{jl}, q_{jl}, g_{kl}, h_{kl}, s_{kl}, t_{kl}, a_j, b_j, c_k, d_k, \xi_1$ and ξ_2 .

Initialize the values of r_{ij} and m_{ikl} by K -Means algorithm.

repeat

The variational E-step: Estimate the expected values in Eqs. (4.53)~(4.59), use the current distributions over the model parameters.

The variational M-step: Update the variational solutions for each factor by Eqs. (4.32)~(4.37) using the current values of the moments.

until Convergence criteria is reached.

Compute the expected value of λ_j as $\langle \lambda_j \rangle = \theta_j / (\theta_j + \vartheta_j)$ and substitute it into Eq. (4.14) to obtain the estimated values of the mixing coefficients π_j .

Compute the expected value of γ_k as $\langle \gamma_k \rangle = \rho_k / (\rho_k + \varpi_k)$ and substitute it into Eq. (4.15) to obtain the estimated values of the mixing coefficients η_k .

Calculate the expected values of the features saliencies by $\langle \epsilon_l \rangle = \xi_1^* / (\xi_1^* + \xi_2^*) = (\xi_1 + \sum_{i=1}^N \langle \phi_{il} \rangle) / (\xi_1 + \xi_2 + N)$.

Detect the optimal number of components M and K by eliminating the components with small mixing coefficients close to 0.

4.3 Experimental Results

In this section, we evaluate the effectiveness of the proposed variational infinite GD mixture model with feature selection (*InFsGD*) through synthetic data and two challenging applications namely unsupervised image categorization and image annotation and retrieval. In all our experiments, we initialize the truncation levels M and K as 15 and 10, respectively. The initial values of hyperparameters u, p, g and s of the Gamma priors are set to 1, and v, q, h, t are set to 0.01. The hyperparameters a, b, c and d are set to 1, while ξ_1 and ξ_2 are set to 0.1. Our simulations have supported these specific choices.

Table 4.1: Parameters of the generated data sets. N denotes the total number of elements, N_j denotes the number of elements in cluster j . α_{j1} , α_{j2} , β_{j1} , β_{j2} and π_j are the real parameters. $\hat{\alpha}_{j1}$, $\hat{\alpha}_{j2}$, $\hat{\beta}_{j1}$, $\hat{\beta}_{j2}$ and $\hat{\pi}_j$ are the estimated parameters by the proposed algorithm.

	N_j	j	α_{j1}	β_{j1}	α_{j2}	β_{j2}	π_j	$\hat{\alpha}_{j1}$	$\hat{\beta}_{j1}$	$\hat{\alpha}_{j2}$	$\hat{\beta}_{j2}$	$\hat{\pi}_j$
Data set 1	200	1	10	15	21	12	0.50	10.12	14.59	20.38	11.73	0.501
($N = 400$)	200	2	25	18	35	40	0.50	23.67	18.65	36.18	41.26	0.499
Data set 2	200	1	10	15	21	12	0.25	9.81	15.89	20.51	12.10	0.253
($N = 800$)	200	2	25	18	35	40	0.25	25.77	18.32	36.03	41.68	0.249
	400	3	18	35	10	25	0.50	17.35	34.29	10.72	26.65	0.498
Data set 3	200	1	10	15	21	12	0.25	10.09	15.57	21.33	11.54	0.247
($N = 800$)	200	2	25	18	35	40	0.25	24.13	17.28	35.15	38.66	0.251
	200	3	18	35	10	25	0.25	18.61	34.19	9.71	25.08	0.248
	200	4	33	27	45	13	0.25	31.95	26.83	43.89	12.27	0.254
Data set 4	200	1	10	15	21	12	0.20	9.34	14.50	20.18	12.35	0.197
($N = 1000$)	200	2	25	18	35	40	0.20	26.07	18.16	34.49	39.12	0.199
	200	3	18	35	10	25	0.20	17.31	36.53	10.76	24.22	0.203
	200	4	33	27	45	13	0.20	31.52	26.35	47.03	13.98	0.204
	200	5	20	10	42	38	0.20	19.88	10.94	41.14	36.67	0.197

4.3.1 Synthetic data

The purpose of the synthetic data is to investigate the accuracy of the proposed algorithm in terms of parameters estimation and model selection. The performance of the *InFsGD* was evaluated through quantitative analysis on four ten-dimensional (two relevant features and eight irrelevant features) synthetic data. The relevant features were generated in the transformed space from mixtures of Beta distributions with well-separated components, while irrelevant ones were from mixtures of overlapped components. Table 4.1 illustrates the real and estimated parameters of the distributions representing the relevant features for each data set using the proposed algorithm. According to this table, the parameters of the model, representing relevant features, and its mixing coefficients are accurately estimated by the *InFsGD*. Similarly, the values of the parameters of the mixture models representing irrelevant features (the eight remaining features) were also correctly obtained (in terms of both parameters estimation and model selection) by adopting the proposed algorithm.

Figure 4.2 shows the estimated mixing coefficients of the mixture components, in each data set, after convergence. By removing the components with very small mixing coefficients (close to 0) in each data set, we obtain the correct number of components for the mixtures representing relevant features. Furthermore, we present the results of the features saliencies of all the 10 features for each data set over ten runs in Figure 4.3. It obviously shows that features 1 and 2 have been assigned a high degree of relevance, which matches the ground-truth.

4.3.2 Visual Scenes Categorization

In this experiment, a challenging problem namely image categorization is highlighted. It is a fundamental task in vision and has recently drawn considerable interest and has been successfully applied in various applications such as the automatic understanding of images, object recognition, image databases browsing and content-based images suggestion and retrieval [114]. As the majority of computer vision tasks, a central step for accurate images categorization is the extraction of good descriptors (i.e. discriminative and invariant at the same time) to represent these images. Recently local descriptors have been widely and successfully used [115, 116] mainly via the

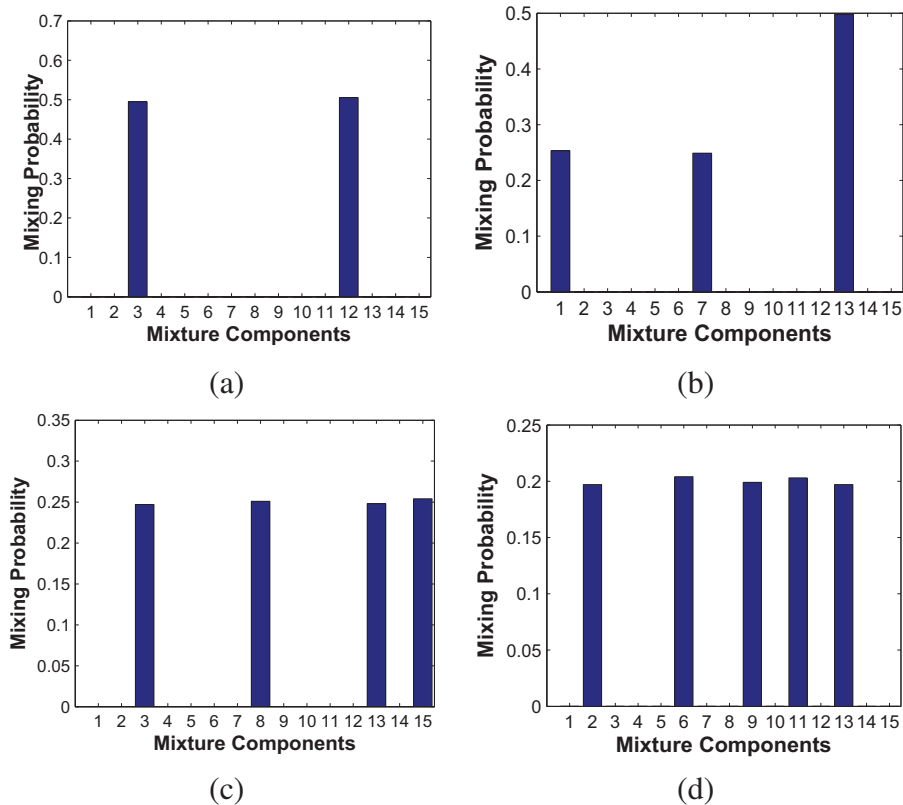


Figure 4.2: Mixing probabilities of components, π_j , found for each synthetic data set after convergence. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4.

bag-of-visual words approach [86, 87, 117] which has allowed the development of many models inspired from text analysis such as the pLSA model [88]. Recently, it has been shown that the performance of visual words-based approaches to images categorization can be significantly improved by adopting multiple image segmentations instead of considering the entire image as a way to utilize visual grouping cues to generate groups of related visual words [117, 118].

The methodology that we have adopted for categorizing images can be summarized as follows: First, we compute multiple candidate segmentations for each image in the collection using Normalized Cuts [119]¹. Following that, Gradient location-orientation histogram (GLOH) descriptors [120] are extracted from each image using the Hessian-Laplace region detector [121]².

¹Source code: <http://www.seas.upenn.edu/~timothee/software/ncut/ncut.html>

²Source code: <http://www.robots.ox.ac.uk/~vgg/research/affine/>

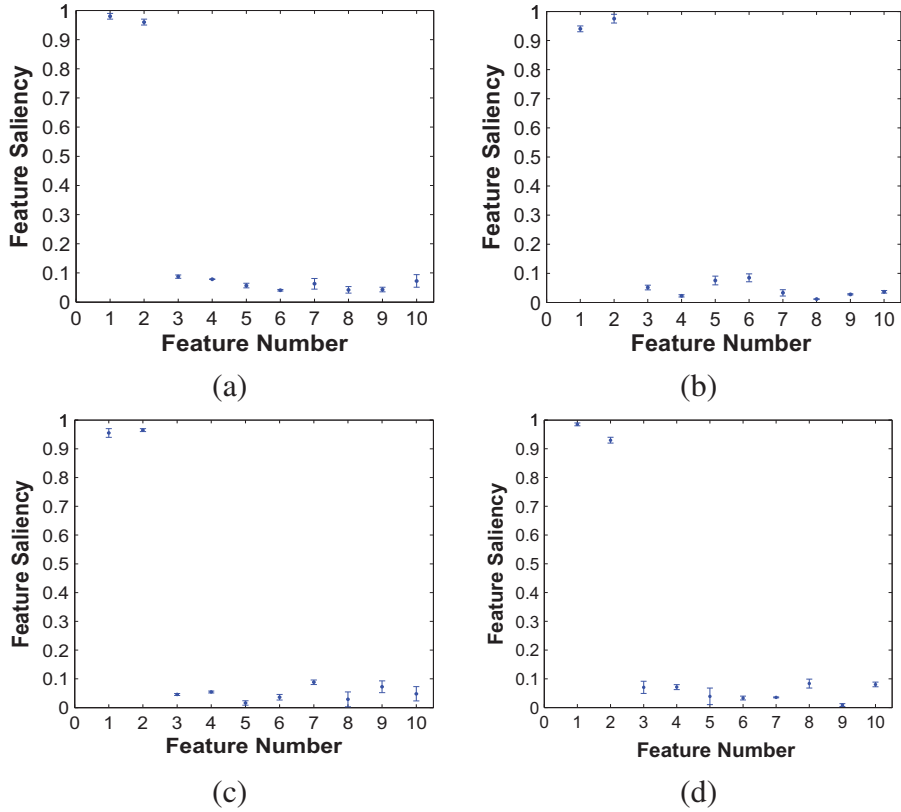


Figure 4.3: Features saliencies for synthetic data sets with one standard deviation over ten runs. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4.

Note that, the GLOH descriptor is an extension of the SIFT descriptor, and is shown to outperform SIFT [120]. Principal component analysis (PCA) is then used to reduce the dimensionality to 128. Next, a visual vocabulary \mathcal{V} is constructed by quantizing these feature vectors into visual words using K -means algorithm and each image is then represented as a frequency histogram over the visual words. Based on our experiments, the optimal performance can be obtained when $\mathcal{V} = 800$. Then, we apply the pLSA model to the bag-of-visual words representation which allows the description of each image as a D -dimensional vector of proportions where D is the number of aspects (or learned topics). Finally, we employ the proposed *InFsGD* as a classifier to categorize images by assigning each test image to the class which has the highest posterior probability according to Bayes' decision rule.

In our experiment, we adopted a subset of the challenging Caltech data set [122] to evaluate the

effectiveness of the proposed approach. Specifically, we considered four object classes from the Caltech data set [122] which include: “airplane”, “face”, “car”, and “motorbike”. Sample images from this data set is displayed in Figure 4.4. This data set is randomly divided into two halves: one for training (constructing the visual words) and the other for testing. We evaluated the performance of the proposed algorithm by running it 20 times.

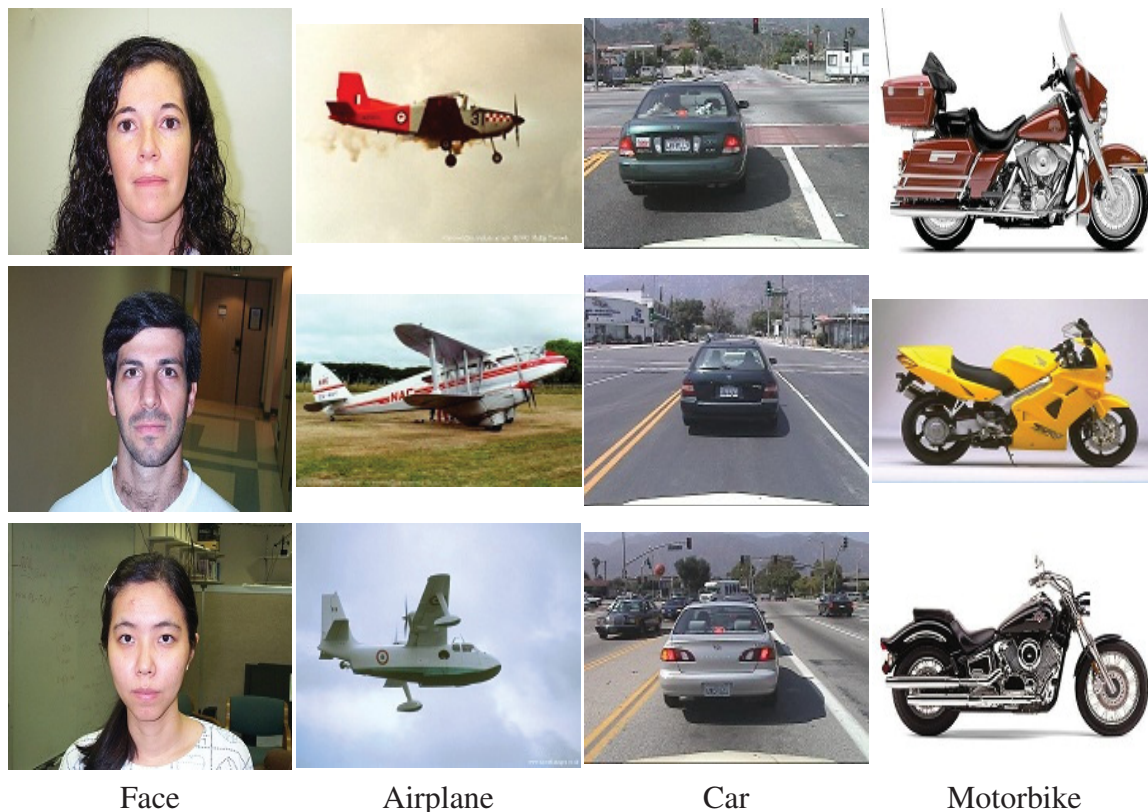


Figure 4.4: Sample images from the four categories of the Caltech data set.

For comparison, we have also applied four other models with the same experimental setting: the finite GD mixture model with feature selection ($FsGD$), the infinite GD mixture model without feature selection ($InGD$), the infinite Gaussian mixture model ($InGau$) proposed in [112] and the Gaussian mixture model with feature selection ($FsGau$) as learned in [59]. To make a fair comparison, all of these models are learned in a variational way. In our experiment, first, multiple segmentations for each image is computed. Some sample segments for images from each category

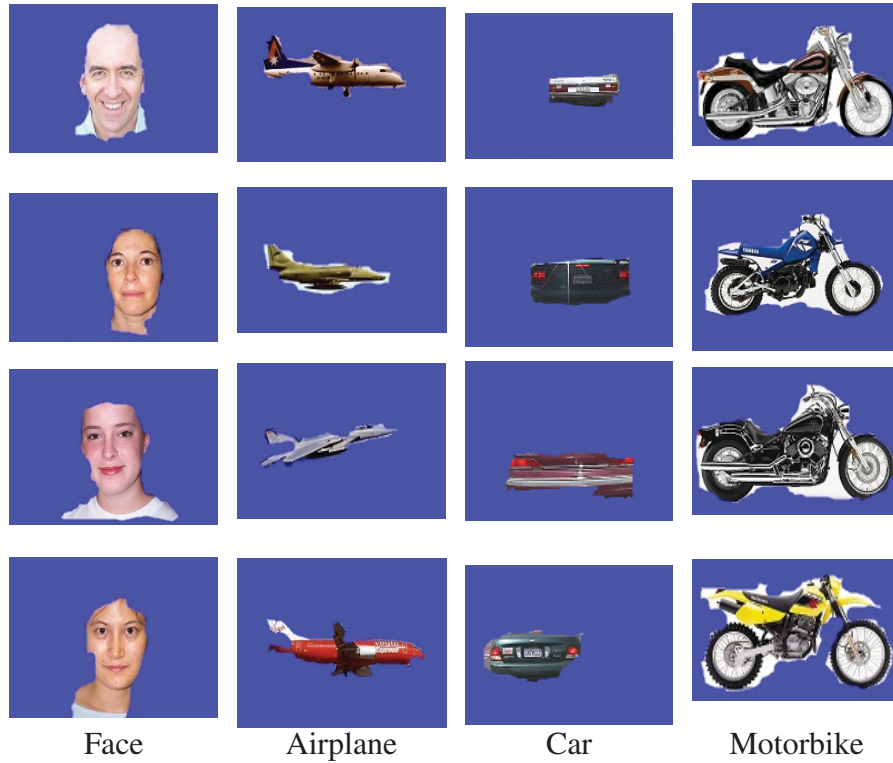


Figure 4.5: Sample segmentation results from the four categories of the Caltech data sets

in this data set are shown in Figure 4.5. The categorizing accuracy using the different tested approaches are presented in Table 4.2. According to the results in this table, the proposed *InFsGD* provides the best performance among the tested algorithms in terms of the highest classification rate and the most accurately estimation of the number of categories. Additionally, the number of components for the mixture model representing irrelevant features was estimated as 2. Furthermore, we have tested the evolution of the classification accuracy with different number of aspects as shown in Figure. 4.6 (a). Based on this figure, the highest classification accuracy can be obtained when we set the number of aspects to 40. The corresponding feature saliencies of the 40 aspects obtained by *InFsGD* are illustrated in Figure. 4.6 (b). As shown in this figure, it is clear that the features have different relevance degrees and then contribute differently to images categorization.

Table 4.2: The average classification accuracy and the number of categories (\hat{M}) computed by different algorithms for the Caltech data set.

	<i>InFsGD</i>	<i>FsGD</i>	<i>InGD</i>	<i>InGau</i>	<i>FsGau</i>
\hat{M}	3.9	3.75	3.85	3.8	3.7
Accuracy (%)	90.21	88.64	88.03	84.19	81.75

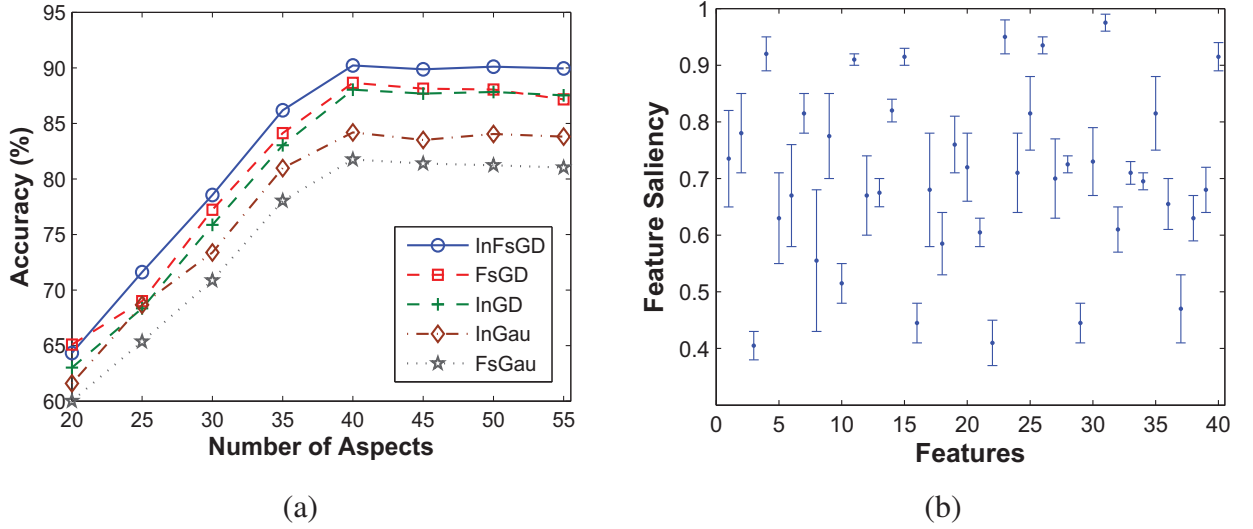


Figure 4.6: (a) Classification accuracy vs. the number of aspects; (b) Feature saliency for each aspect.

4.3.3 Image Auto-Annotation

Methodology

Many images carrying extremely rich information are now archived in large databases. A challenging problem is then to automatically analyze, organize, index, browse and retrieve these images. A lot of approaches have been proposed to address this problem. In particular, semantic image understanding and auto-annotation have been the topic of extensive research in the past [123–128]. The main goal is to extract high-level semantic features in addition to low level features to bridge the gap between them and to enhance visual scenes interpretation abilities [129–131]. Automatic annotation approaches can be divided into two main groups of approaches [132, 133]. The first group deals directly with the annotation problem by providing labels to the complete image or its

different regions (see, for instance, [129, 130]). The second group tackles this problem via two independent steps where the first step categorizes the images and the second one attaches labels to them using the top ranked categories (see, for instance, [125, 133]). Approaches in this second group have shown promising results recently. Thus, the goal of this subsection is to develop an annotation-driven image retrieval approach, based on the work in [133], via categorization results obtained with the proposed *InFsGD* in a bag-of-visual key words representation. Our aim is to build an efficient annotation-retrieval approach to handle the problem of image search under three challenging scenarios as stated in [133]: 1) use a tagged image or a set of keywords as query to search images on the untagged portion of a partially tagged image database; 2) use an untagged image as query to search images on the tagged portion of a partially tagged image database; 3) use an untagged image as query to search images on an untagged image database. The methodology that we have adopted for this experiment can be divided into three sequential steps namely: images categorization, annotation, and retrieval.

In the categorization stage, the proposed *InFsGD* is integrated with the pLSA model to categorize images through a bag-of-key visual words representation. First, interest points are detected using the Difference-of-Gaussian (DoG) detector [121]. Then, we use PCA-SIFT descriptor¹ [134], computed on detected keypoints of all images and resulting on 36-dimensional vector for each keypoint. Subsequently, the *K*-Means algorithm is used to construct a visual vocabulary by quantizing these PCA-SIFT vectors into visual words. In our experiments, we set the vocabulary size to 1000. Each image is then represented as a frequency histogram over the visual words. Then, the pLSA model is applied to the obtained histograms to represent each image by a 50-dimensional proportional vector where 50 is the number of latent aspects. Finally, our *InFsGD* is deployed to cluster the images.

The categorization results in the previous stage are exploited to perform image annotation. Here, we follow an approach proposed in [133] which considers the problem of image annotation from three phases: 1) the frequency of occurrence of potential tags based on the categorization results; 2) saliency of the given tags; 3) the congruity of a word among all the candidate tags. Assume that we have a training image data set that contains several categories. Each category is

¹Source code of PCA-SIFT: <http://www.cs.cmu.edu/~yke/pcasift>

annotated by 4 to 5 tags where common tags may appear in different categories. At the beginning, we collect all the tags from each category. The total number of categories in the data set is denoted as C and the number of categories that have each unique tag t is represented as $F(t)$. Then, tag saliency can be evaluated similarly as for inverse document frequency in the field of document retrieval. For a test image, a ranked list of predicted categories is generated according to the Bayes' decision rule in the classification. Then, the top 5 predicted categories are chosen and the union of all involved unique tags denoted as $U(I)$ forms the set of candidate tags. Thus, we define $f(t|I)$ as the frequency of the occurrence of each unique tag t among the top 5 predicted categories. We follow the idea proposed in [133] to determine the word congruity using WordNet [135] with the Leacock and Chowdrow measure [136]. WordNet is a large lexical database of English which groups English words into sets of cognitive synonyms called synsets. Hence, the congruity for a candidate tag t can be calculated by [133]:

$$G(t|I) = \frac{d_{tot}(I)}{d_{tot}(I) + |U(I)| \sum_{x \in U(I)} d_{LCH}(x, t)} \quad (4.60)$$

We adopt the same settings for d_{LCH} and r_{LCH} as in [133], such that the distance between two tags t_1 and t_2 is: $d_{LCH}(t_1, t_2) = \exp(-r_{LCH}(t_1, t_2) + 3.584) - 1$. In addition, $d_{tot}(I)$ evaluates the pairwise semantic distance among all candidate tags and is defined as:

$$d_{tot}(I) = \sum_{x \in U(I)} \sum_{y \in U(I)} d_{LCH}(x, y) \quad (4.61)$$

By having all the three annotation factors on hand, we can compute the overall score for a candidate tag as

$$A(t|I) = a_1 f(t|I) + \frac{a_2}{\ln C} \ln\left(\frac{C}{1 + F(t)}\right) + a_3 G(t|I) \quad (4.62)$$

where $a_1 + a_2 + a_3 = 1$ represents the degree of importance of the three factors. Then, a tag t is chosen for annotation only if its score is within the top ε percentile among the candidate tags. According to our experimental results, we set $a_1 = 0.5$, $a_2 = 0.2$, $a_3 = 0.3$, and $\varepsilon = 0.7$. For retrieving images, we use automatic annotation and the WordNet-based bag-of-words distances as introduced in [133]. The core idea is that if tags were missing in the query image or in our database, automatic annotation is then performed and the bag-of-words distances between

Table 4.3: The average classification accuracy computed by different algorithms.

Method	Accuracy (%)
<i>InFsGD</i>	75.1
<i>InGD</i>	74.7
<i>InGau</i>	73.6
<i>SC-GM</i>	71.8
<i>FsGau</i>	70.2

query image tags and the database tags are calculated. This distance is used to rank the degree of relevance of the images in the database and then to perform images search accordingly (more details and discussions can be found in [133]).

Results

We test out approach using a subset of LabelMe data set [137] which contains both class labels and annotations. First, we use the LabelMe Matlab toolbox² to obtain images online from 8 outdoor scene classes: “highway”, “inside city”, “tall building”, “street”, “forest”, “coast”, “mountain” and “open country”. We randomly choose 200 images from each category. Thus, we have 1600 images in total. Each category is associated with 4-5 tags. We randomly divide the data set into two partitions: one for training, the other for testing. First, we have performed categorization using the proposed *InFsGD* with bag-of-visual key words representation as described previously. We compare our approach with other four well-defined approaches: the infinite GD mixture model without feature selection (*InGD*), the variational infinite Gaussian mixture model (*InGau*), the combination of a structure-composition model and a Gaussian mixture model (we denote it as *SC-GM*) as proposed in [133] and the Gaussian mixture model with feature selection (*FsGau*). The categorization result of the 8 outdoor scene images is illustrated in Table 4.3. According to this table, we can observe that the proposed *InFsGD* outperforms other four approaches in terms of the highest classification accuracy rate (75.1%).

The obtained result from the categorization is then exploited by the annotation stage. The performance of annotation is evaluated by precision and recall which are defined in the standard

²<http://labelme.csail.mit.edu/>

Table 4.4: Performance evaluation on the automatic annotation system based on different categorization methods.

Method	Mean Precision (%)	Mean Recall (%)
<i>InFsGD</i>	31.5	43.6
<i>InGD</i>	30.4	42.3
<i>InGau</i>	29.8	40.2
<i>SC-GM</i>	27.1	38.7
<i>FsGau</i>	26.3	36.8

way: the annotation precision for a keyword is defined as the number of tags correctly predicted divided by the total number of predicted tags. The annotation recall is defined as the number of tags correctly predicted, divided by the number of tags in the ground-truth annotation. In our experiments, the average number of tags generated for each test image is 4.05. Table 4.4 shows the performance evaluation of the automatic annotation approach according to the categorization result obtained by using different methods. It is clear that, annotation with the categorization result obtained by *InFsGD* provides the best performance. Table 4.5 presents some examples of the annotations produced by using *InFsGD* categorization method.

In the last step, we perform image retrieval under the three scenarios as described in the previous subsection. For the first scenario in which the database is not tagged and query may either be keywords or tagged image, the retrieval is performed by first automatically annotating the database through categorization and annotation steps. Then, image retrieval is performed according to the bag-of-words distances between query tags and our annotation. In this experiment, we use 40 pairs of query words that are randomly chosen from all the candidate tags. In the second scenario, the database is tagged and the query is an untagged image. Thus, the first step to automatically annotate the query image. Then, the database is ranked according to the bag-of-words distances. In the third scenario, neither the image database nor the query is tagged. Therefore, both the image database and the query images have to be annotated automatically first. Subsequently, image retrieval is applied once again using the bag-of-words distance evaluation. We choose 100 images randomly as the set of query images in this experiment. The performance of semantic retrieval was

Table 4.5: Sample annotation results by using *InFsGD* classification method.






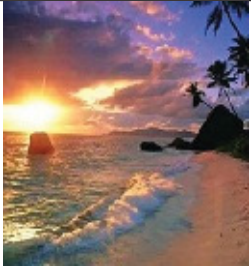
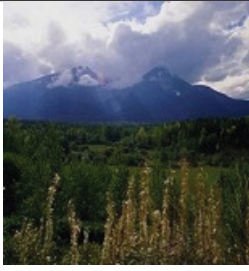

				
Our labels	car, road, mountain	car, sidewalk, window	sky, building, tree	human, car, tree
LabelMe labels	truck, car, sky, road, mountain	building, car, window, sidewalk, human	building, tree, car, sky	person, car, sidewalk, building, tree
				
Our labels	sea water, tree, sky	sand, tree, sea water	forest, sky, cloud	cloud, field, mountain, tree
LabelMe labels	tree, forest, mountain, cloud, sky	sea water, sand, sky, cloud	mountain, sky, field, tree	sky, sand, field, mountain, car

Table 4.6: The comparison of image retrieval performance.

Method	Scenario 1		Scenario 2		Scenario 3	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)	Precision (%)	Recall (%)
<i>InFsGD</i>	51.5	58.9	45.3	50.2	47.5	56.6
<i>InGD</i>	49.7	56.6	42.5	49.3	46.6	54.1
<i>InGau</i>	48.6	56.3	41.4	48.7	45.9	52.8
<i>SC-GM</i>	46.2	55.7	38.6	45.6	41.7	53.5
<i>FsGau</i>	43.8	52.1	37.1	43.4	38.3	51.0

evaluated by measuring precision and recall. In this case, precision is defined as the proportion of retrieved images that are relevant, and recall denotes the proportion of relevant images that are retrieved. An image is considered relevant if there is an overlap between the original tags of the query image or query word and the original tags of the retrieved image. Since categorization is the baseline of our annotation-driven image retrieval approach. We have also tested the impact of using different categorization algorithms on annotation-driven image retrieval performance and illustrates the corresponding result in Table 4.6 on retrieving the top 10 relevant images. As we can observe from this table, using *InFsGD* as the categorization method provides the best performance for all three scenarios which indicates that the categorization algorithm is a significant influence factor for the annotation-driven image retrieval scheme that we have applied.

Chapter 5

Online Learning of a Dirichlet Process Mixture of Beta-Liouville Distributions via Variational Inference

A large class of problems can be formulated in terms of clustering process. Mixture models are an increasingly important tool in statistical pattern recognition and for analyzing and clustering complex data. Two challenging aspects that should be addressed when considering mixture models are: how to choose between a set of plausible models and how to estimate the model's parameters. In this chapter, we address both problems simultaneously within a unified online nonparametric Bayesian framework that we develop to learn a Dirichlet process mixture of Beta-Liouville distributions (i.e. an infinite Beta-Liouville mixture model). The proposed infinite model is used for the online modeling and clustering of proportional data for which the Beta-Liouville mixture has been shown to be effective. We propose a principled approach for approximating the intractable model's posterior distribution by a tractable one, such that all the involved mixture's parameters can be estimated simultaneously and effectively in a closed form. This is done through variational inference that enjoys important advantages, such as handling of unobserved attributes and preventing under- or over-fitting, and that we explain in details. The effectiveness of the proposed work is evaluated on three challenging real applications namely facial expression recognition, behavior modeling and recognition, and dynamic textures clustering.

5.1 Beta-Liouville Mixture Model

Recently, Beta-Liouville mixture models have drawn considerable attention and have been successfully applied in many applications [24]. The Beta-Liouville distribution contains the Dirichlet distribution as a special case and has a smaller number of parameters than the generalized Dirichlet. Furthermore, Beta-Liouville mixture models have shown better performance than both the Dirichlet and the generalized Dirichlet mixtures as detailed in [24]. More properties and discussions about the Beta-Liouville can be viewed in [138, 139]. In this section, first we introduce the finite Beta-Liouville mixture model. Then, we present its extension to the infinite case via a stick-breaking construction of Dirichlet process framework.

5.1.1 Finite Beta-Liouville Mixture Model

Given a D -dimensional vector $\vec{X} = (X_1, \dots, X_D)$ which follows the Beta-Liouville distribution with positive parameters $\theta = (\alpha_1, \dots, \alpha_D, \alpha, \beta)$, then the probability density function of \vec{X} is given by [138]

$$\text{BL}(\vec{X}|\theta) = \frac{\Gamma(\sum_{d=1}^D \alpha_d) \Gamma(\alpha + \beta)}{\Gamma(\alpha) \Gamma(\beta)} \prod_{d=1}^D \frac{X_d^{\alpha_d - 1}}{\Gamma(\alpha_d)} \left(\sum_{d=1}^D X_d \right)^{\alpha - \sum_{d=1}^D \alpha_d} \left(1 - \sum_{d=1}^D X_d \right)^{\beta - 1} \quad (5.1)$$

Assume that we have observed a set of N vectors $\mathcal{X} = \{\vec{X}_1, \dots, \vec{X}_N\}$, where each vector $\vec{X}_i = (X_{i1}, \dots, X_{iD})$ is represented in a D -dimensional space and assumed to be generated from a finite Beta-Liouville mixture model with M components, then [24]

$$p(\vec{X}_i | \vec{\pi}, \vec{\theta}) = \sum_{j=1}^M \pi_j \text{BL}(\vec{X}_i | \theta_j) \quad (5.2)$$

where $\text{BL}(\vec{X}_i | \theta_j)$ is a Beta-Liouville distribution corresponding to component j with parameters $\theta_j = (\alpha_{j1}, \dots, \alpha_{jD}, \alpha_j, \beta_j)$. In addition, $\vec{\theta} = (\theta_1, \dots, \theta_M)$, and $\vec{\pi} = (\pi_1, \dots, \pi_M)$ denotes the vector of mixing coefficients which are positive and sum to one.

5.1.2 Infinite Beta-Liouville Mixture Model

Stick-breaking Construction

In this subsection, we extend the finite Beta-Liouville mixture model to the infinite case by exploiting a Dirichlet process formulation. In our work, the Dirichlet process is constructed by adopting a stick-breaking framework, which is defined as follows [109]: given a random distribution G , it is Dirichlet process distributed with a base distribution H and concentration parameter ψ (denoted as $G \sim \text{DP}(\psi, H)$), if the following conditions are satisfied:

$$\lambda_j \sim \text{Beta}(1, \psi), \quad \Omega_j \sim H, \quad \pi_j = \lambda_j \prod_{s=1}^{j-1} (1 - \lambda_s), \quad G = \sum_{j=1}^{\infty} \pi_j \delta_{\Omega_j} \quad (5.3)$$

where δ_{Ω_j} denotes the Dirac delta measure centered at Ω_j , and π_j is the mixing proportion in terms of mixture modeling terminology and is defined by recursively breaking a unit length stick into an infinite number of pieces. The Dirichlet process can be translated to a mixture model with a countably infinite number of components by its nonparametric nature [140]. In the case of Dirichlet process mixture model, the actual number of components is not fixed, and can be automatically inferred from the data using Bayesian posterior inference framework.

The Infinite Model

Assume now that we have observed \mathcal{X} which is generated from a Beta-Liouville mixture model with a countably infinite number of components. Then, the infinite Beta-Liouville mixture model can be written as

$$p(\vec{X}_i | \vec{\pi}, \vec{\theta}) = \sum_{j=1}^{\infty} \pi_j \text{BL}(\vec{X}_i | \theta_j) \quad (5.4)$$

In mixture modeling, we generally use auxiliary variables to allocate each vector to a specific cluster. Thus, we introduce a M -dimensional binary random vector $\vec{Z}_i = \{Z_{i1}, \dots, Z_{iM}\}$ for each observed vector \vec{X}_i , 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. $\mathcal{Z} = \{\vec{Z}_1, \dots, \vec{Z}_N\}$ is known as the set of “membership vectors” of the mixture

model and its distribution is specified in terms of the mixing coefficients $\vec{\pi}$, such that

$$p(\mathcal{Z}|\vec{\pi}) = \prod_{i=1}^N \prod_{j=1}^{\infty} \pi_j^{Z_{ij}} \quad (5.5)$$

Notice that, $\vec{\pi}$ is a function of $\vec{\lambda}$ according to the stick-breaking construction of Dirichlet process as shown in Eq. (5.3). Then, we can write

$$p(\mathcal{Z}|\vec{\lambda}) = \prod_{i=1}^N \prod_{j=1}^{\infty} [\lambda_j \prod_{s=1}^{j-1} (1 - \lambda_s)]^{Z_{ij}} \quad (5.6)$$

The prior distribution of $\vec{\lambda}$ is the specific Beta distribution as shown in Eq. (5.3):

$$p(\vec{\lambda}|\vec{\psi}) = \prod_{j=1}^{\infty} \text{Beta}(1, \psi_j) = \prod_{j=1}^{\infty} \psi_j (1 - \lambda_j)^{\psi_j - 1} \quad (5.7)$$

The primary difficulty when adopting variational learning approach lies with the choice of conjugate priors. In our case, since α_d , α and β are positive, Gamma distributions $\mathcal{G}(\cdot)$ are adopted to approximate conjugate priors for these parameters: $p(\alpha_d) = \mathcal{G}(\alpha_d|u_d, v_d)$, $p(\alpha) = \mathcal{G}(\alpha|g, h)$ and $p(\beta) = \mathcal{G}(\beta|s, k)$.

5.2 Online Variational Model Learning

In this section, we first develop a batch variational inference framework for learning infinite Beta-Liouville mixture models. Subsequently, an online extension is proposed. To summarize, the main goal is to develop a variational approach that learns an infinite Beta-Liouville mixture model by simultaneously optimizing both its parameters and its structure (i.e. complexity or number of mixture components) in both batch and online settings. To simplify the notation, in the following sections we define $\Theta = \{\mathcal{Z}, \Lambda\}$ as the set of latent and unknown random variables where $\Lambda = \{\vec{\lambda}, \vec{\theta}\}$.

5.2.1 Batch Variational Learning

The main idea of variational inference to find an approximation $Q(\Theta)$ for the true posterior distribution $p(\Theta|\mathcal{X})$. This is done by maximizing the lower bound on the model evidence $\ln p(\mathcal{X})$,

which is defined by

$$\mathcal{L}(Q) = \int Q(\Theta) \ln[p(\mathcal{X}, \Theta)/Q(\Theta)] d\Theta \quad (5.8)$$

In this work, we adopt a truncation technique proposed in [112] to truncate the variational distributions at a value M , such that $\lambda_M = 1$, $\sum_{j=1}^M \pi_j = 1$, and $\pi_j = 0$ when $j > M$. Notice that the truncation level M is a variational parameter which can be freely initialized and will be optimized automatically during the learning process. By adopting the truncated stick-breaking representation and the factorization assumption, we obtain

$$Q(\Theta) = \left[\prod_{i=1}^N Q(Z_i) \right] \left[\prod_{j=1}^M \prod_{d=1}^D Q(\alpha_{jd}) \right] \left[\prod_{j=1}^M Q(\lambda_j) Q(\alpha_j) Q(\beta_j) \right] \quad (5.9)$$

Two alternative approaches with equivalent results can be applied for variational inference. In the first approach, a general solution for optimizing each variational factor exists and is given by [66, chapter 10]

$$Q_s(\Theta_s) = \frac{\exp\langle \ln p(\mathcal{X}, \Theta) \rangle_{\neq s}}{\int \exp\langle \ln p(\mathcal{X}, \Theta) \rangle_{\neq s} d\Theta} \quad (5.10)$$

where $\langle \cdot \rangle_{\neq s}$ denotes the expectation with respect to the Q distributions over all variables except for Θ_s . We have adopted this approach in previous Chapters to learn finite Dirichlet, GD and infinite GD mixture models. The second approach for deriving optimization solutions in variational inference is based on a gradient method [141]. Since this gradient-based approach can be easily adapted to online learning, it is adopted here to learn infinite Beta-Liouville mixtures in a batch manner and then will be extended into an online version in the next subsection. The major idea of the gradient-based variational learning approach is that, since the model has conjugate priors, the functional form of the factors in the variational posterior distribution is known. Thus, by taking general parametric forms for these distributions, the lower bound can be considered as a function of the parameters of these distributions. The optimization of variational factors is then achieved by maximizing the lower bound with respect to these parameters. In our case, the functional form for each variational factor is the same as its conjugate prior distribution, namely Discrete for \mathcal{Z} , Beta for $\vec{\lambda}$, and Gamma for $\vec{\alpha}_d$, $\vec{\alpha}$ and $\vec{\beta}$. Therefore, we can define the parametric forms for these

variational posterior distributions as the following

$$Q(\mathcal{Z}) = \prod_{i=1}^N \prod_{j=1}^M r_{ij}^{Z_{ij}}, \quad Q(\vec{\lambda}) = \prod_{j=1}^M \text{Beta}(\lambda_j | c_j, d_j) \quad (5.11)$$

$$Q(\vec{\alpha}_d) = \prod_{j=1}^M \prod_{d=1}^D \mathcal{G}(\alpha_{jd} | u_{jd}^*, v_{jd}^*) \quad (5.12)$$

$$Q(\vec{\alpha}) = \prod_{j=1}^M \mathcal{G}(\alpha_j | g_j^*, h_j^*), \quad Q(\vec{\beta}) = \prod_{j=1}^M \mathcal{G}(\beta_j | s_j^*, k_j^*) \quad (5.13)$$

Consequently, the parameterized lower bound $\mathcal{L}(Q)$ can be obtained by substituting Eqs. (5.11), (5.12) and (5.13) into Eq. (5.8) (See Appendix C.1). Maximizing this bound with respect to these parameters then gives the required re-estimation equations (Details of the variational inference procedure are given in Appendices C.2 to C.4). Thus, we can obtain

$$r_{ij} = \frac{\tilde{r}_{ij}}{\sum_{j=1}^M \tilde{r}_{ij}}, \quad c_j = 1 + \sum_{i=1}^N \langle Z_{ij} \rangle, \quad d_j = \psi_j + \sum_{i=1}^N \sum_{s=j+1}^N \langle Z_{is} \rangle \quad (5.14)$$

$$\begin{aligned} \tilde{r}_{ij} = \exp & \left[\mathcal{S}_j + \mathcal{H}_j + (\bar{\alpha}_j - \sum_{d=1}^D \bar{\alpha}_{jd}) \ln \left(\sum_{d=1}^D X_{id} \right) + (\bar{\beta}_j - 1) \ln \left(1 - \sum_{d=1}^D X_{id} \right) + \sum_{d=1}^D (\bar{\alpha}_{jd} - 1) \ln X_{id} \right. \\ & \left. + \langle \ln \lambda_j \rangle + \sum_{s=1}^{j-1} \langle \ln(1 - \lambda_s) \rangle \right] \end{aligned} \quad (5.15)$$

where \mathcal{S}_j and \mathcal{H}_j are given by Eq. (C.4) and Eq. (C.3), respectively.

$$u_{jd}^* = u_{jd} + \sum_{i=1}^N \langle Z_{ij} \rangle \bar{\alpha}_{jd} \left[\Psi \left(\sum_{d=1}^D \bar{\alpha}_{jd} \right) + \Psi' \left(\sum_{d=1}^D \bar{\alpha}_{jd} \right) \sum_{l \neq d}^D (\langle \ln \alpha_{jl} \rangle - \ln \bar{\alpha}_{jl}) \bar{\alpha}_{jl} - \Psi(\bar{\alpha}_{jd}) \right] \quad (5.16)$$

$$v_{jd}^* = v_{jd} - \sum_{i=1}^N \langle Z_{ij} \rangle \left[\ln X_{id} - \ln \left(\sum_{d=1}^D X_{id} \right) \right] \quad (5.17)$$

$$g_j^* = g_j + \sum_{i=1}^N \langle Z_{ij} \rangle \left[\bar{\beta}_j \Psi'(\bar{\alpha}_j + \bar{\beta}_j) (\langle \ln \beta_j \rangle - \ln \bar{\beta}_j) - \Psi(\bar{\alpha}_j) + \Psi(\bar{\alpha}_j + \bar{\beta}_j) \right] \bar{\alpha}_j \quad (5.18)$$

$$h_j^* = h_j - \sum_{i=1}^N \langle Z_{ij} \rangle \ln \left(\sum_{d=1}^D X_{id} \right), \quad k_j^* = k_j - \sum_{i=1}^N \langle Z_{ij} \rangle \ln \left(1 - \sum_{d=1}^D X_{id} \right) \quad (5.19)$$

$$s_j^* = s_j + \sum_{i=1}^N \langle Z_{ij} \rangle [\bar{\alpha}_j \Psi'(\bar{\alpha}_j + \bar{\beta}_j) (\langle \ln \alpha_j \rangle - \ln \bar{\alpha}_j) + \Psi(\bar{\alpha}_j + \bar{\beta}_j) - \Psi(\bar{\beta}_j)] \bar{\beta}_j \quad (5.20)$$

where $\Psi(\cdot)$ is the digamma function. The expected values in the above formulas are defined as

$$\bar{\alpha}_{jd} = \frac{u_{jd}^*}{v_{jd}^*}, \quad \bar{\alpha}_j = \frac{g_j^*}{h_j^*}, \quad \bar{\beta}_j = \frac{s_j^*}{k_j^*}, \quad \langle Z_{ij} \rangle = r_{ij} \quad (5.21)$$

$$\langle \ln \alpha_{jd} \rangle = \Psi(u_{jd}^*) - \ln v_{jd}^* \quad (5.22)$$

$$\langle \ln \alpha_j \rangle = \Psi(g_j^*) - \ln h_j^*, \quad \langle \ln \beta_j \rangle = \Psi(s_j^*) - \ln k_j^* \quad (5.23)$$

$$\langle \ln \lambda_j \rangle = \Psi(c_j) - \Psi(c_j + d_j), \quad \langle \ln(1 - \lambda_j) \rangle = \Psi(d_j) - \Psi(c_j + d_j) \quad (5.24)$$

The batch variational inference for infinite Beta-Liouville mixture model can be approached via an EM-like framework and is summarized in Algorithm 4. The convergence of this batch learning algorithm can be monitored through inspection of the variational bound. After convergence, we may notice that the expected values of the mixing coefficients of some components are numerically distinguishable from their prior values while others are close 0. This effect can be explained qualitatively in terms of the automatic trade-off in a Bayesian model between fitting the data and the complexity of the model, in which the complexity penalty stems from components whose parameters are pushed away from their prior values [66].

5.2.2 Online Variational Inference

In this subsection, we extend the batch variational inference approach for learning infinite Beta-Liouville mixture model to online settings by adopting the framework proposed in [141]. Since in many real-world applications data points are continuously arriving over time in an online manner, it is desirable to estimate the variational lower bound corresponding to a fixed amount of data. In our case, let t denotes the actual amount of observed data. Then, the current lower bound for the observed data is given by

$$\mathcal{L}^{(t)}(Q) = \frac{N}{t} \sum_{i=1}^t \int Q(\Lambda) d\Lambda \sum_{\vec{Z}_i} Q(\vec{Z}_i) \ln \left[\frac{p(\vec{X}_i, \vec{Z}_i | \Lambda)}{Q(\vec{Z}_i)} \right] + \int Q(\Lambda) \ln \left[\frac{p(\Lambda)}{Q(\Lambda)} \right] d\Lambda \quad (5.25)$$

Algorithm 4 Batch variational learning of infinite Beta-Liouville mixture.

- 1: Choose the initial truncation level M .
 - 2: Initialize the values for hyper-parameters $\psi_j, u_{jd}, v_{jd}, g_j, h_j, s_j$ and k_j .
 - 3: Initialize the values of r_{ij} by K -Means algorithm.
 - 4: **repeat**
 - 5: *The variational E-step:*
 - 6: Estimate the expected values in Eqs. (5.21)~(5.24), use the current distributions over the model parameters.
 - 7: *The variational M-step:*
 - 8: Update the variational solutions for each factor using Eqs. (5.11), (5.12) and (5.13) and the current values of the moments.
 - 9: **until** Convergence criterion is reached.
 - 10: Compute the expected value of λ_j as $\langle \lambda_j \rangle = c_j / (c_j + d_j)$ and substitute it into Eq. (5.3) to obtain the estimated values of the mixing coefficients π_j .
 - 11: Detect the optimal number of components M by eliminating the components with small mixing coefficients close to 0 (less than 10^{-5}).
-

where $\Lambda = \{\vec{\lambda}, \vec{\theta}\}$. The key idea of the online variational learning algorithm is to successively maximize the current variational lower bound Eq. (5.25). Assume that we have already observed a data set $\{X_1, \dots, X_{(t-1)}\}$. For a new observation X_t , we can maximize the current lower bound $\mathcal{L}^{(t)}(Q)$ with respect to $Q(\vec{Z}_t)$, while other variational factors are fixed to $Q^{(t-1)}(\vec{\lambda})$, $Q^{(t-1)}(\vec{\alpha}_d)$, $Q^{(t-1)}(\vec{\alpha})$ and $Q^{(t-1)}(\vec{\beta})$. Thus, the variational solution to $Q(\vec{Z}_t)$ is given by

$$Q(\vec{Z}_t) = \prod_{j=1}^M r_{tj}^{Z_{tj}} \quad (5.26)$$

where

$$r_{tj} = \frac{\tilde{r}_{tj}}{\sum_{j=1}^M \tilde{r}_{tj}} \quad (5.27)$$

and

$$\begin{aligned} \tilde{r}_{tj} = \exp \left\{ \mathcal{S}_j^{(t-1)} + \mathcal{H}_j^{(t-1)} + (\bar{\alpha}_j^{(t-1)} - \sum_{d=1}^D \bar{\alpha}_{jd}^{(t-1)}) \ln \left(\sum_{d=1}^D X_{td} \right) + (\bar{\beta}_j^{(t-1)} - 1) \ln \left(1 - \sum_{d=1}^D X_{td} \right) \right. \\ \left. + \sum_{d=1}^D (\bar{\alpha}_{jd}^{(t-1)} - 1) \ln X_{td} + \langle \ln \lambda_j^{(t-1)} \rangle + \sum_{s=1}^{j-1} \langle \ln(1 - \lambda_s^{(t-1)}) \rangle \right\} \quad (5.28) \end{aligned}$$

Next, the current lower bound $\mathcal{L}^{(t)}(Q)$ is maximized with respect to $Q^{(t)}(\vec{\lambda})$, while $Q(\vec{Z}_t)$ is fixed and other variational factors remain at their $(t-1)$ th values. Therefore, we can obtain the variational solution to $Q^{(t)}(\vec{\lambda})$:

$$Q^{(t)}(\vec{\lambda}) = \prod_{j=1}^M \text{Beta}(\lambda_j^{(t)} | c_j^{(t)}, d_j^{(t)}) \quad (5.29)$$

where the hyperparameters are defined by

$$c_j^{(t)} = c_j^{(t-1)} + \rho_t \Delta c_j^{(t)}, \quad d_j^{(t)} = d_j^{(t-1)} + \rho_t \Delta d_j^{(t)} \quad (5.30)$$

where ρ_t is the learning rate which is used to reduce the earlier inaccurate estimation effects that contributed to the lower bound and accelerate the convergence of the learning process. In this work, we adopt a learning rate function introduced in [142], such that $\rho_t = (\eta_0 + t)^{-a}$, subject to the constraints $a \in (0.5, 1]$ and $\eta_0 \geq 0$. In Eq. (5.30), $\Delta c_j^{(t)}$ and $\Delta d_j^{(t)}$ are the natural gradients of the corresponding hyperparameters. The natural gradient of a parameter is obtained by multiplying the gradient by the inverse of Riemannian metric, which cancels the coefficient matrix for the posterior parameter distribution. Thus, we can obtain the following natural gradients as

$$\Delta c_j^{(t)} = c_j^{(t)} - c_j^{(t-1)} = 1 + Nr_{tj} - c_j^{(t-1)} \quad (5.31)$$

$$\Delta d_j^{(t)} = d_j^{(t)} - d_j^{(t-1)} = \psi_j + N \sum_{s=j+1}^M r_{ts} - d_j^{(t-1)} \quad (5.32)$$

Subsequently, the current lower bound $\mathcal{L}^{(t)}(Q)$ is maximized with respect to $Q^{(t)}(\vec{\alpha}_d)$ and the corresponding variational solution is given by

$$Q^{(t)}(\vec{\alpha}_d) = \prod_{j=1}^M \prod_{d=1}^D \mathcal{G}(\alpha_{jd}^{(t)} | u_{jd}^{*(t)}, v_{jd}^{*(t)}) \quad (5.33)$$

where

$$u_{jd}^{*(t)} = u_{jd}^{*(t-1)} + \rho_t \Delta u_{jd}^{*(t)}, \quad v_{jd}^{*(t)} = v_{jd}^{*(t-1)} + \rho_t \Delta v_{jd}^{*(t)} \quad (5.34)$$

The corresponding natural gradients are defined by

$$\Delta u_{jd}^{*(t)} = u_{jd} + Nr_{tj} \bar{\alpha}_{jd} \left[\Psi \left(\sum_{d=1}^D \bar{\alpha}_{jd} \right) + \Psi' \left(\sum_{d=1}^D \bar{\alpha}_{jd} \right) \sum_{l \neq d}^D (\langle \ln \alpha_{jl} \rangle - \ln \bar{\alpha}_{jl}) \bar{\alpha}_{jl} - \Psi(\bar{\alpha}_{jd}) \right] - u_{jd}^{*(t-1)} \quad (5.35)$$

$$\Delta v_{jd}^{*(t)} = v_{jd} - Nr_{tj} \left[\ln X_{td} - \ln \left(\sum_{d=1}^D X_{td} \right) \right] - v_{jd}^{*(t-1)} \quad (5.36)$$

The solutions to the hyperparameters of $Q^{(t)}(\vec{\alpha})$ and $Q^{(t)}(\vec{\beta})$ can be computed similarly. This online variational inference procedure is repeated until all the variational factors are updated with respect to the new observation. The computational complexity for the proposed online variational infinite Beta-Liouville mixture is $\mathcal{O}(MD)$ in contrast to $\mathcal{O}(NMD)$ for its batch version in each iteration. This is because the batch algorithm updates the variational solutions by using the whole data set in each iteration. Thus, the proposed online algorithm is much more computationally efficient since the estimation quality of the batch algorithm is improved more slowly than in the case of the online one. The total computational time depends on the number of iterations required for convergence. The online variational inference for infinite Beta-Liouville mixture model is summarized in Algorithm 5.

Algorithm 5 Online variational learning of infinite Beta-Liouville mixture.

Choose the initial truncation level M .

Initialize the values for hyper-parameters $\psi_j, u_{jd}, v_{jd}, g_j, h_j, s_j$ and k_j .

for $t = 1 \rightarrow N$ **do**

The variational E-step:

Update the variational solution to $Q(\vec{Z}_t)$ using Eq. (5.26).

The variational M-step:

Compute learning rate $\rho_t = (\eta_0 + t)^{-\alpha}$.

Calculate the following natural gradients: $\Delta c_j^{(t)}, \Delta d_j^{(t)}, \Delta u_{jd}^{*(t)}, \Delta v_{jd}^{*(t)}, \Delta g_j^{*(t)}, \Delta h_j^{*(t)}, \Delta s_j^{*(t)}$ and $\Delta k_j^{*(t)}$.

Update the variational solutions to $Q^{(t)}(\vec{\lambda}), Q^{(t)}(\vec{\alpha}_d), Q^{(t)}(\vec{\alpha})$ and $Q^{(t)}(\vec{\beta})$.

Repeat the variational *E-step* and *M-step* until new data is observed.

end for

5.3 Experimental Results

5.3.1 Design of Experiments

In this section, the effectiveness of the proposed online infinite Beta-Liouville mixture model (referred to as *OIBLM*) is evaluated through three challenging applications involving facial expression recognition, behavior modeling and recognition, and dynamic textures clustering. The first goal of these applications is to evaluate the performance of *OIBLM* in terms of estimation (estimating the model’s parameters) and selection (selecting the number of components of the mixture model). The second goal is to show that our algorithm works well on diverse types of digital data. Three types of digital media namely images, videos and dynamic textures are considered in our experiments where each kind of media is used in one application. The third goal is to demonstrate the merits of Beta-Liouville mixtures by comparing the performance of the proposed *OIBLM* to three other online infinite mixture models including the infinite generalized Dirichlet (*OIGDM*), infinite Dirichlet (*OIDM*) and infinite Gaussian (*OIGM*) mixtures. To make a fair comparison, all these models are learned using online variational inference. It is also noteworthy that in all our real applications, the testing data are supposed to arrive sequentially in an online fashion. In our experiments, we initialize the truncation level M and the hyperparameter ψ to 15 and 0.1, respectively. The initial values of hyperparameters u , g and s of the Gamma priors are set to 1, and v , h , k are set to 0.01. The parameters a and η_0 of the learning rate are set to 0.75 and 64, respectively. Our simulations have supported these specific choices. It is worth mentioning that we have evaluated the sensitivity of our model to the initialization specification by repeating our algorithm several times with different initial values of hyperparameters. However, no significant improvement or influence on the learning process has been observed according to our experiments.

5.3.2 Facial Expression Recognition

Problem statement

Facial expression recognition is a crucial step to understand human emotion and paralinguistic communication. It provides clues about affective state, cognitive activity and psychopathology

and then may have important applications in human-computer interaction [143]. This problem is challenging and far from straightforward especially under variable illumination conditions and head motion [144, 145]. The majority of the research efforts on vision-based facial expression analysis and recognition rely on the well-known Ekman’s emotional categorization referred to as the basic emotions [146] including happiness, sadness, surprise, fear, anger, and disgust, that are widely discussed in a series of interesting books [147–149]. The development of methodologies to tackle this problem is still an active area of research with several promising approaches proposed in the literature [150–156]. Although different, these approaches have been mainly based on solving two sub-problems namely feature extraction and facial expression categorization. In this experiment, we follow these approaches by applying our *OIBLM* for categorization in conjunction with Local Binary Pattern (LBP) [157] features-based representation. The choice of LBP features is motivated by the fact that they have shown recently promising results in facial image analysis [157, 158]. In contrast to other proposed facial expression features, LBP features are more robust against illumination changes and are more computationally efficient [158]. It is noteworthy that we shall focus on static face images, without regard to temporal information, in this subsection. Temporal behaviors of facial expression in image sequences will be considered in the set of experiments in subsection 5.3.3.

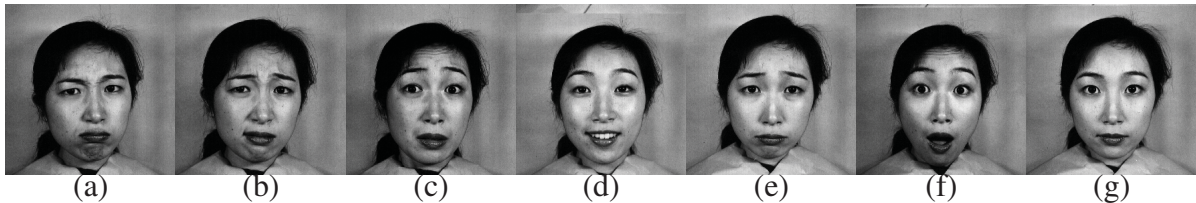


Figure 5.1: Sample images from the JAFFE data set: (a) Anger, (b) Disgust, (c) Fear, (d) Happiness, (e) Sadness, (f) Surprise, (g) Neutral.

Methodology and Results

We use the same preprocessing step suggested in [159] by cropping original images into 110×150 pixels to reduce the influence of background. As a result, the cropped images remain the central

Table 5.1: The average recognition accuracy (%) and the number of categories (\widehat{M}) computed by different algorithms for the JAFFE data set. The numbers in parenthesis are the standard deviations of the corresponding quantities.

	<i>OIBLM</i>	<i>OIGDM</i>	<i>OIDM</i>	<i>OIGM</i>
\widehat{M}	6.71 (0.25)	6.64 (0.29)	6.52 (0.35)	6.43 (0.38)
Accuracy	88.28 (1.09)	86.17 (1.33)	84.52 (1.18)	81.37 (1.41)

part of facial expression. Next, we extract LBP features from face images. More specifically, each cropped face image is first divided into small regions from which LBP histograms are then extracted and concatenated into a single feature histogram representing the face image [158]. We use the same experimental settings for extracting LBP features as in [158]: we adopt a 59-bin LBP operator in the (8,2) neighborhood (which means 8 sampling points on a circle of radius of 2) and divide each image (110×150) into 18×21 pixels regions. Therefore, face images are divided into 42 (6×7) regions and are then represented by LBP histograms with length of 2478 (59×42). Then, we apply the pLSA model [88] as a dimensionality reduction technique to the LBP feature vectors. Each image is then represented as a 40-dimensional vector of proportions. Finally, we employ the proposed *OIBLM* to cluster the sequentially arriving images.

In our experiment, we have adopted the Japanese Female Facial Expression (JAFFE) data set¹ which is a benchmark in the field of facial expression recognition. It contains 213 images of 7 facial expressions (neutral plus six basic facial expressions: anger, disgust, fear, happiness, sadness and surprise) posed by 10 Japanese female models aged from 20 to 40. Each image size is of 256×256 pixels and each expresser has 2~4 samples for each expression. Sample images from this data set with different facial expressions are shown in Figure 5.1.

We evaluated the performance of the proposed algorithm by running it 30 times. The confusion matrix for the JAFFE data set provided by *OIBLM* is shown in Figure 5.2. Furthermore, we have tested three other algorithms (*OIGDM*, *OIDM* and *OIGM*) for comparison. The average recognition accuracy and the average estimated number of categories obtained by each algorithm are shown in Table 5.1. According to this table, it is obvious that the proposed *OIBLM* outperforms

¹This data set is available at: <http://www.kasrl.org/jaffe.html>

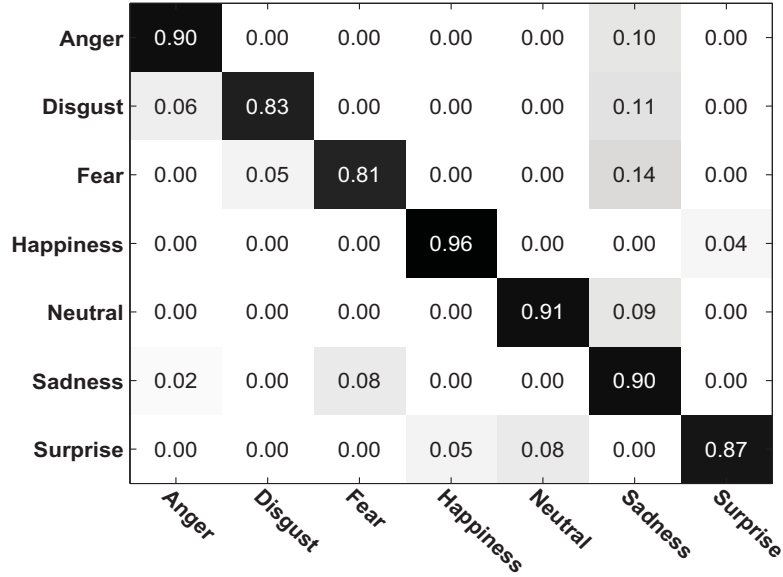


Figure 5.2: Confusion matrix obtained by *OIBLM* for the JAFFE data set.

Table 5.2: The average recognition accuracy rate (Acc) and the average estimated number of categories (\widehat{M}) computed using different algorithms on the three data sets: facial expression (face), mouse behavior (mouse) and human activity (UCF11).

Data set	<i>OIBLM</i>		<i>OIGDM</i>		<i>OIDM</i>		<i>OIGM</i>	
	Acc (%)	\widehat{M}	Acc (%)	\widehat{M}	Acc (%)	\widehat{M}	Acc (%)	\widehat{M}
Face	87.18 (1.19)	5.72 (0.23)	85.94 (1.26)	5.63 (0.28)	82.71 (1.43)	5.56 (0.31)	80.25 (1.71)	5.52 (0.37)
Mouse	75.68 (0.98)	4.67 (0.29)	74.09 (1.02)	4.61 (0.32)	71.33 (1.57)	4.55 (0.31)	69.54 (1.49)	4.49 (0.35)
UCF11	81.27 (1.34)	10.46 (0.45)	79.13 (1.67)	10.35 (0.52)	77.45 (1.82)	10.29 (0.61)	74.39 (1.75)	10.25 (0.58)

the other three algorithms by providing the highest recognition accuracy rate (88.28%) and the most accurate estimated number of categories (6.71).

5.3.3 Behavior Modeling and Recognition

Learning object, event and behavior classes is an important problem in computer vision which has several applications [160–162]. Recent popular methods have been based on the representation of images and videos as collections of local visual descriptors extracted from patches or interest points. Various interest points (space-time interest points in the case of videos) detectors and local

visual descriptors exist. The usual way to use the resulting visual descriptors is to quantize them, using a certain clustering process such as K-Means or randomized forests [163, 164], to produce the so-called visual words. In this experiment, we present an unsupervised learning method, based on our online variational algorithm with the bag-of-visual words representation, for recognizing various kinds of behaviors in video sequences. Among many of the existing space-time interest points detectors and local spatio-temporal features, we adopt the so-called cuboid detector [99] which has shown its effectiveness in behavior modeling. The Cuboid detector is based on temporal Gabor filters and a histogram of the cuboid types and shall be used here as our behavior descriptor.

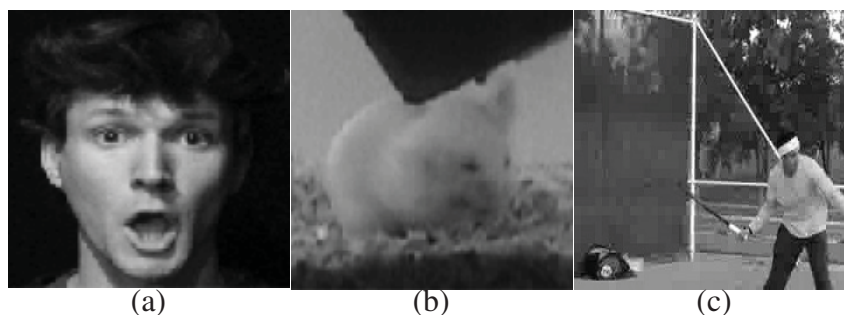


Figure 5.3: Sample frames from the each data set. (a): facial expression; (b): mouse behavior; (c): human action.

The methodology of our unsupervised behavior recognition approach is summarized as follows. First, we extract local spatio-temporal features known as cuboids using the cuboid detector as proposed in [99] from the already observed video sequences. In our work, we use the same settings as in [99] for extracting cuboids and constructing the behavior descriptors. Next, a visual vocabulary is constructed by quantizing these spatio-temporal features into visual words using K-means algorithm and each video is then represented as a frequency histogram over the visual words. Then, we apply the pLSA model as a dimensionality reduction technique to represent each video as a D -dimensional proportional vector where D is the number of latent aspects. In this experiment, according to our experimental results, the optimal number of aspects was around 45. Lastly, the testing videos are clustered using the proposed *OIBLM* algorithm.

We conducted our experiments on three representative domains: temporal behaviors of facial

expressions [165], mouse behavior and human action. we use the same facial expression and mouse behavior data sets provided by [99]. The facial expression video data set contains about 192 video clips which are collected from 2 individuals under 2 lighting conditions. Each individual was asked to repeatedly perform 6 expressions (anger, disgust, fear, joy, sadness and surprise) 8 times. The mouse data includes 406 clips with 5 behaviors performed by the same mouse: drinking, eating, exploring, grooming and sleeping. The human action video data that we adopted in this experiment is the UCF11 data sets [166]². It contains 1168 video sequences in total with 11 action categories: cycling, diving, golf swinging, soccer juggling, trampoline jumping, horse-back riding, basketball shooting, volleyball spiking, swinging, tennis swinging, and walking with a dog. Sample frames from each data set are shown in Figure 5.3.

Each data set is randomly divided into two halves: one for constructing the visual vocabulary, the other for testing. The results are obtained over 30 runs. Table 5.2 shows the average number of clusters and the average recognition accuracies using *OIBLM*, *OIGDM*, *OIDM* and *OIGM* algorithms. The average performance of these different algorithms is also illustrated in Figure 5.4. According to these results, we can clearly see that the *OIBLM* outweighs the other algorithms by providing the best performance on all testing data sets. Given the difficulty of the considered data sets, these results are rather encouraging.

5.3.4 Dynamic Textures Clustering

Dynamic texture, which is an extension of texture to the temporal domain, can be defined as a video sequence of moving scenes that exhibit some stationarity characteristics in time (e.g., fire, sea waves, smoke, swinging flag in the wind, foliage, etc.) [167]. Dynamic textures have attracted growing attention during the last decade since they can be used in various applications such as facial expressions recognition, video surveillance, development of screen savers, personalized web pages, and video games [168–170].

In this experiment, we address the problem of clustering dynamic textures using the proposed *OIBLM* algorithm. Given a video sequence of a single dynamic texture, our goal is to recognize

²This data set is available at: <http://vision.eecs.ucf.edu/datasetsActions.html>

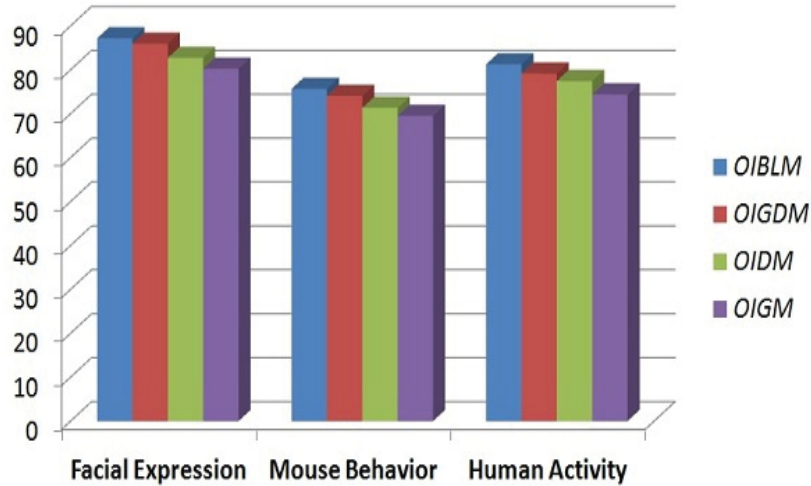


Figure 5.4: Performance comparison on the three data sets: facial expression, mouse behavior and human activity using different algorithms.

which class the video sequence belongs to. We adopt a dynamic texture modeling framework previously proposed in [170]. This framework is based on modeling a video sequence by a collection of linear dynamical systems (LDSs) where each one describes a small spatio-temporal patch extracted from the video. In particular, we use the so-called bag-of-systems (BoS) representation which is able to explicitly capture the dynamics of dynamic textures. The first step of this approach consists of extracting LDS descriptors from the available video sequences using the dense sampling approach [170]. More specifically, given a video sequence, first we divide it into non-overlapping spatio-temporal volumes with size $a \times b \times c$, where a and b denote the spatial size while c is the temporal size. In this experiment, we used a patch-size of $20 \times 20 \times 25$ which has provided us the optimal performance according to our results. Then, each spatio-temporal volume is modeled using a LDS of order 3 to form a feature descriptor. After extracting all the features from the video sequences, we build a visual vocabulary using the K-Medoid approach [2] to quantize these features into visual words. Next, we reduce the dimensionality of these feature vectors via the pLSA model by considering 35 topics. Then, each dynamic texture is represented as a 35-dimensional proportional vector. Finally, we apply the proposed *OIBLM* to cluster our dynamic textures.

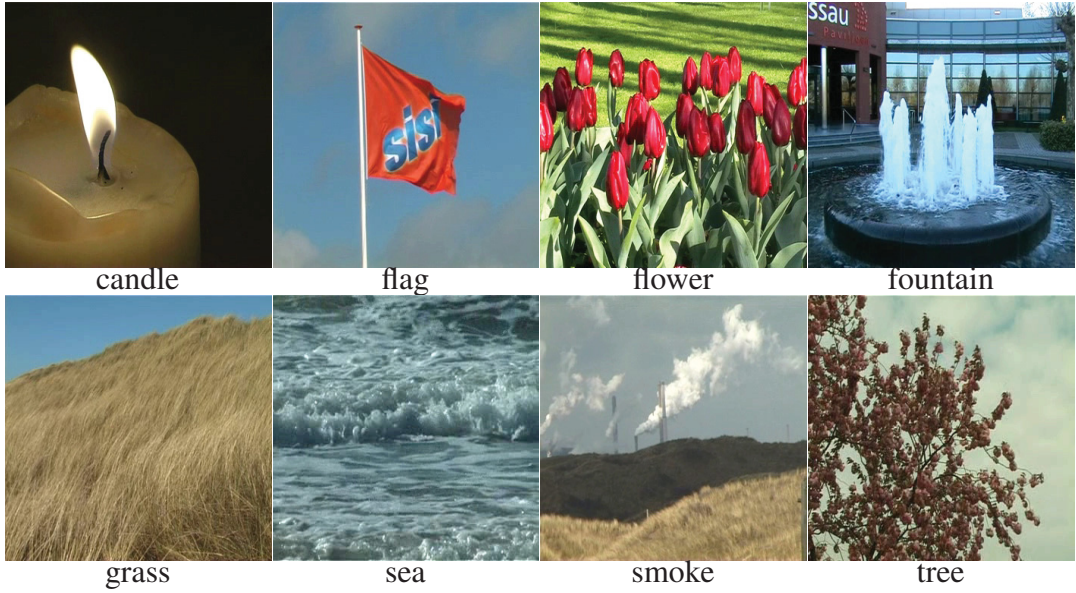


Figure 5.5: Sample frames from the DynTex data set.

Table 5.3: The average accuracy and the number of categories (\widehat{M}) computed by different algorithms when clustering the DynTex data set.

	<i>OIBLM</i>	<i>OIGDM</i>	<i>OIDM</i>	<i>OIGM</i>
\widehat{M}	6.75 (0.41)	6.69 (0.38)	6.46 (0.49)	6.37 (0.52)
Accuracy	83.37 (1.72)	80.62 (1.96)	77.75 (2.34)	74.87(2.28)

A challenging dynamic textures data set, which is known as the DynTex database [171]³, is considered in this experiment. This data set contains around 650 dynamic texture video sequences from various categories. In our case, we use a subset of this data set which contains 8 categories of dynamic textures: candle, flag, flower, fountain, grass, sea, smoke and tree. Each category has 20 video sequences with a size of 352×288 . As a preprocessing step, we re-sampled all the video sequences into a size of 360×300 to avoid extracting overlapping patches and in order to not disregard any region. We have used half the data to construct the visual vocabulary and the rest for testing. Sample frames from each category are shown in Figure 5.5. We run the proposed *OIBLM* 30 times for evaluating its performance. For comparison, we have also tested *OIGDM*, *OIDM*

³This data set is available at: <http://projects.cwi.nl/dyntex/index.html>

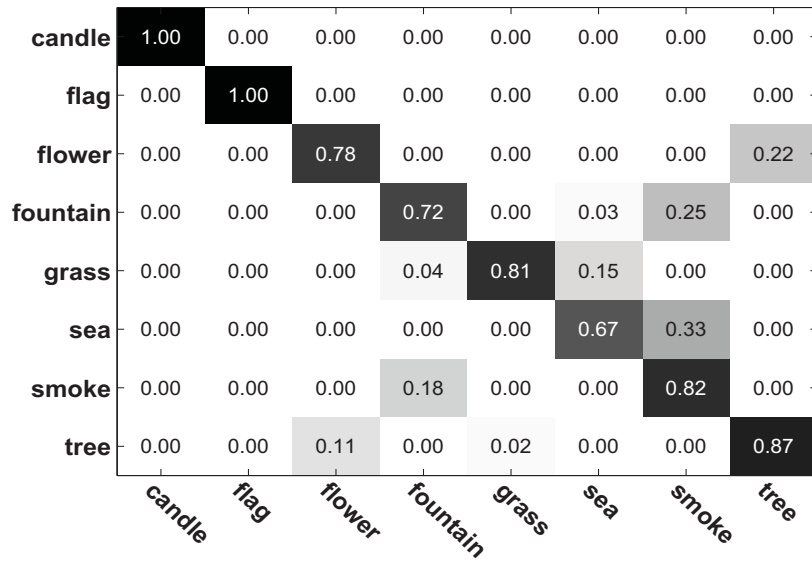


Figure 5.6: Confusion matrix obtained by *OIBLM* for the DynTex data set.

and *OIGM* algorithms using the same experimental methodology. Figure 5.6 shows the confusion matrix for the DynTex data set using *OIBLM*. The average results of the clustering accuracy and the estimated number of categories are illustrated in Table 5.3. Although the number of categories is underestimated (6.75) by our algorithm, it is clear that it outperforms the rest of the algorithms in terms of the highest categorization accuracy rate (83.37%) as shown in Figure 5.7.

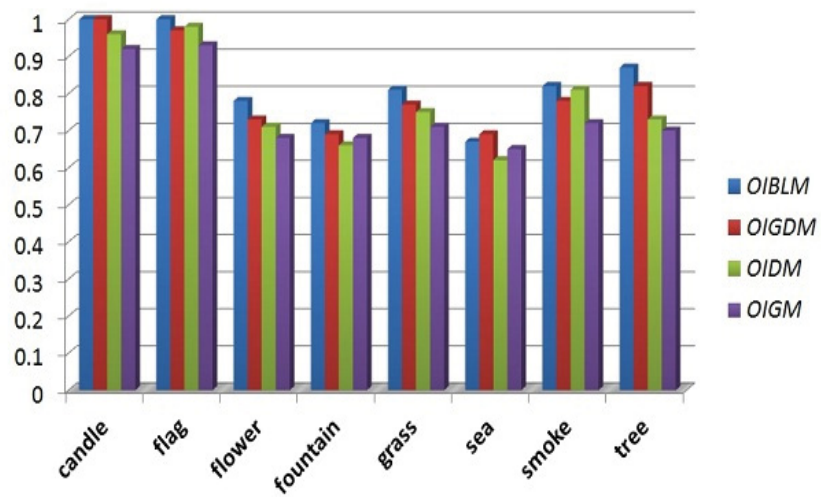


Figure 5.7: Performance comparison in terms of classification accuracy provided different algorithms for the DynTex data set.

Chapter 6

Conclusions

Clustering is an important problem in several fields, such as signal and image processing. In this thesis, we have developed several approaches for high-dimensional non-Gaussian data clustering. Our approaches are based on variational learning of various mixture models such as Dirichlet, generalized Dirichlet and Beta-Liouville. We are mainly motivated by the promising results obtained by using these mixtures to model non-Gaussian data, especially those involving normalized count data (i.e., proportional vectors) which naturally appear in many applications such as text, image and video modeling.

In Chapter 2, we have presented an efficient attractive procedure for the variational learning of finite Dirichlet mixture models. Our procedure is based on the construction and the optimization of a lower bound on the model's likelihood by choosing completely factorized conditional distributions over the model's variables. The proposed framework can be viewed as a compromise between ML estimation which prefers complex models and then causes over-fitting and pure Bayesian techniques which penalizes complex models, but unfortunately require intensive computations and are generally intractable. Indeed, unlike pure Bayesian methods which require sampling, the proposed variational approach approximates posterior distributions over model parameters analytically thanks to the accurate choice of specific conjugate priors. Through extensive experiments we have shown that proposed variational framework allows the automatic and simultaneous adjusting of the mixture parameters and the number of components. It is noteworthy that the ability of our variational approach to lead to a model with the correct number of components has been based solely on empirical evidence via our experiments. These experiments have involved both synthetic and real challenging problems such as image databases categorization and intrusion detection.

Most of the feature selection algorithms based on mixture models assume that the data in

each component follow Gaussian distribution, which is seldom the case in real-life applications. Unlike these approaches, we have proposed in Chapter 3 a principled variational framework for unsupervised feature selection in the case of non-Gaussian data which naturally appear in many applications from different domains and disciplines. Variational frameworks offer a deterministic alternative for Bayesian approximate inference by maximizing a lower bound on the marginal likelihood which main advantage is computational efficiency and guaranteed convergence that can be easily assessed as compared to MCMC-based approaches which make posterior approximation in a stochastic sense. We have shown that the variational approach can be used to obtain a closed form parameters posteriors for our model. The proposed approach has been applied to both synthetic data and to a challenging application which concerns human action videos categorization, with encouraging results. It is noteworthy that the proposed selection model is also applicable to many other challenging problems involving non-Gaussian proportional data such as text mining and compression, and protein sequences modeling in biology.

Until recently, feature selection approaches based on mixture models were almost exclusively considered in the finite case. The work proposed in Chapter 4 is motivated by an attempt to overcome this limitation via the extension of the simultaneous clustering and feature selection approach based on finite generalized Dirichlet mixture models, to the infinite case via Dirichlet processes with a stick-breaking representation. The proposed technique drives much of its power from the flexibility of the generalized Dirichlet mixture, the high generalization accuracy of Dirichlet processes, and the advantages of the variational Bayesian framework that we have developed to learn our model. Our method has been successfully tested in several scenarios and our experimental results using synthetic data and real-world applications namely visual scenes categorization, image annotation and retrieval have shown advantages derived from its adoption. The model developed in this chapter is also applicable to many other problems which involve high-dimensional data clustering such as gene microarray data sets analysis, text clustering and retrieval, and object recognition.

In Chapter 5, we have presented a coherent statistical framework based on the newly introduced Beta-Liouville mixture which has been shown to outperform both the Dirichlet and the generalized Dirichlet mixtures for proportional data clustering. The proposed framework uses Dirichlet

process formalism with a truncated stick-breaking representation which results in an infinite Beta-Liouville mixture model. The learning of this infinite model has been tackled via an efficient attractive procedure, based on online variational inference, that we have developed. Within this learning framework, we have developed a variational lower bound on the likelihood of the proposed infinite model which optimization results in a deterministic EM-like algorithm. Extensive empirical results have shown the merits and effectiveness of the proposed approach. These experiments have involved real challenging problems namely facial expression recognition, behavior modeling and recognition, and dynamic textures categorization.

In conclusion, variational frameworks offer a deterministic alternative for Bayesian approximate inference by maximizing a lower bound on the marginal likelihood which main advantage is computational efficiency and guaranteed convergence that can be easily assessed as compared to MCMC-based approaches which make posterior approximation in a stochastic sense. Like pure Bayesian learning, variational learning provides good generalization capabilities, but at a significant lower computational cost since it does not need calculations of high-dimensional integrals using MCMC methods. The variational approach allows analytical calculations of posterior distributions over the mixture hidden variables, parameters and structure. In other words it allows simultaneous inference on both model and parameter space. It is our hope that the proposed approaches will serve to inspire more interesting applications and learning techniques since proportional data arise in many other problems such as protein sequence modeling in molecular biology, text mining, images annotation, user profiling, collaborative filtering and recommendation.

There are a number of potential future directions that we are going to pursue. These directions are towards extending the approaches we have currently proposed to more general domains. For instance, we can integrate hierarchies into our approaches through hierarchical Bayesian nonparametric frameworks such as hierarchical Dirichlet process (HDP) [172] and hierarchical Pitman-Yor process (HPYP) [173]. Indeed, both HDP and HPYP are extensions to the conventional Dirichlet process where hierarchical model structures are employed. Specifically, HDP possesses a Bayesian hierarchy where the base measure for a set of Dirichlet processes is itself distributed according to a Dirichlet process, while HPYP is a hierarchical Bayesian model based on a two-parameters generalization of the Dirichlet process. We are mainly motivated by the fact that hierarchies can help to

unify statistics, providing a Bayesian interpretation of frequentist concepts such as shrinkage and random effects [174]. Thus, by taking the building blocks provided by simple stochastic processes such as the Dirichlet process, it is possible to construct models that exhibit richer kinds of probabilistic structure. In addition, we may go a step further by extending these hierarchical Bayesian nonparametric frameworks to online settings to make them more efficient and more easily applicable to massive and streaming data.

List of References

- [1] K. C. Gowda and G. Krishna. Dissaggregative clustering using the concept of mutual nearest neighborhood. *IEEE Transactions on Systems, Man and Cybernetics*, 8(12):888–895, 1978.
- [2] L. Kaufman and P. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York, 1990.
- [3] J. M. Pena, J. A. Lozano and P. Larranaga. Learning recursive Bayesian multinets for data clustering by means of constructive induction. *Machine Learning*, 47:63–89, 2002.
- [4] H. Zha, X. He, C. H. Q. Ding, M. Gu and H. D. Simon. Spectral relaxation for k-means clustering. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 1057–1064, 2001.
- [5] A. K. Jain, M. N. Murty and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.
- [6] G. J. McLachlan and D. Peel. *Finite Mixture Models*. New York: Wiley, 2000.
- [7] H. Thomas and T. P. Hettmansperger. Modelling change in cognitive understanding with finite mixtures. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 50(4):435–448, 2001.
- [8] S. Richardson, L. Leblond, I. Jaussent and P. J. Green. Mixture models in measurement error problems, with reference to epidemiological studies. *Journal of the Royal Statistical Society. Series A*, 165(3):549–566, 2002.

- [9] J-S. Hu, C-C. Cheng and W-H. Liu. Robust speaker's location detection in a vehicle environment using GMM models. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 36(2):403–412, 2006.
- [10] S. Marcel and J. del R. Millán. Person authentication using brainwaves (eeg) and maximum a posteriori model adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):743–748, 2007.
- [11] N. Bouguila and D. Ziou. Unsupervised selection of a finite Dirichlet mixture model: An MML-based approach. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):993–1009, 2006.
- [12] W. Fan, N. Bouguila, and D. Ziou. A variational statistical framework for object detection. In Bao-Liang Lu, Liqing Zhang, and James T. Kwok, editors, *ICONIP (2)*, volume 7063 of *Lecture Notes in Computer Science*, pages 276–283. Springer, 2011.
- [13] W. Fan and N. Bouguila. Infinite Dirichlet mixture model and its application via variational Bayes. In *Proc. of International Conference on Machine Learning and Applications and Workshops (ICMLA)*, pages 129–132, 2011.
- [14] W. Fan and N. Bouguila. A variational statistical framework for clustering human action videos. In *Proc. of International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pages 1–4, 2012.
- [15] W. Fan and N. Bouguila. Online variational finite Dirichlet mixture model and its applications. In *Proc. of International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, pages 448–453, 2012.
- [16] W. Fan and N. Bouguila. Variational learning for Dirichlet process mixtures of Dirichlet distributions and applications. *Multimedia Tools and Applications*, pages 1–18, 2012, In press.

- [17] N. Bouguila and D. Ziou. High-dimensional unsupervised selection and estimation of a finite generalized Dirichlet mixture model based on minimum message length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1716–1731, 2007.
- [18] W. Fan, N. Bouguila, and D. Ziou. Unsupervised anomaly intrusion detection via localized Bayesian feature selection. In *Proc. of the IEEE International Conference on Data Mining (ICDM)*, pages 1032–1037, 2011.
- [19] W. Fan and N. Bouguila. Online learning of a Dirichlet process mixture of generalized Dirichlet distributions for simultaneous clustering and localized feature selection. *Journal of Machine Learning Research - Proceedings Track*, 25:113–128, 2012.
- [20] W. Fan and N. Bouguila. Nonparametric localized feature selection via a Dirichlet process mixture of generalized Dirichlet distributions. In Tingwen Huang, Zhigang Zeng, Chuan-dong Li, and Chi-Sing Leung, editors, *ICONIP (3)*, volume 7665 of *Lecture Notes in Computer Science*, pages 25–33. Springer, 2012.
- [21] W. Fan and N. Bouguila. Variational learning of dirichlet process mixtures of generalized Dirichlet distributions and its applications. In Shuigeng Zhou, Songmao Zhang, and George Karypis, editors, *Advanced Data Mining and Applications (ADMA)*, volume 7713 of *Lecture Notes in Computer Science*, pages 199–213. Springer Berlin Heidelberg, 2012.
- [22] W. Fan and N. Bouguila. A variational component splitting approach for finite generalized Dirichlet mixture models. In *Proc. of International Conference on Communications and Information Technology (ICCIT)*, pages 53–57, 2012.
- [23] W. Fan and N. Bouguila. Online variational learning of generalized Dirichlet mixture models with feature selection. *Neurocomputing*, 2013, In press.
- [24] N. Bouguila. Hybrid generative/discriminative approaches for proportional data modeling and classification. *IEEE Transactions on Knowledge and Data Engineering*, 24(12):2184–2202, 2012.

- [25] W. Fan and N. Bouguila. Learning finite Beta-liouville mixture models via variational Bayes for proportional data clustering. In *Proc. of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1323–1329, 2013.
- [26] W. Fan and N. Bouguila. Online facial expression recognition based on finite Beta-Liouville mixture models. In *Proc. of International Conference on Computer and Robot Vision (CRV)*, pages 37–44, 2013.
- [27] W. Fan and N. Bouguila. Variational learning of finite beta-liouville mixture models using component splitting. In *Proc. of International Joint Conference on Neural Networks (IJCNN)*, 2013.
- [28] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
- [29] J.G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 1997.
- [30] M-W. Mak and S-Y. Kung. Estimation of elliptical basis function parameters by the EM algorithm with application to speaker verification. *IEEE Transactions on Neural Networks*, 11(4):961–969, 2000.
- [31] S. Waterhouse, D. MacKay and T. Robinson. Bayesian methods for mixtures of experts. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 351–357, 1995.
- [32] C.P. Robert. *The Bayesian choice*. Springer-Verlag, 2001.
- [33] C. E. Rasmussen. A Practical Monte Carlo Implementation of Bayesian Learning. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 598–604, 1995.
- [34] D. Barber and B. Schottky. Radial basis functions: a Bayesian treatment. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 1997.
- [35] D. Husmeier. The Bayesian evidence scheme for regularizing probability-density estimating neural networks. *Neural Computation*, 11(12):2685–2717, 2000.

- [36] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, 1999.
- [37] D. Husmeier, W. D. Penny, and S. J. Roberts. An empirical evaluation of Bayesian sampling with hybrid Monte Carlo for training neural network classifiers. *Neural Networks*, 12:677–705, 1999.
- [38] S. M. Lewis and A. E. Raftery. Estimating Bayes factors via posterior simulation with the Laplace-Metropolis estimator. *Journal of the American Statistical Association*, 92:648–655, 1997.
- [39] H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Proc. of the 15th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 21–30, 1999.
- [40] H. Attias. A variational Bayes framework for graphical models. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 209–215, 1999.
- [41] G. E. Hinton and D. van Camp. Keeping neural network simple by minimizing the description length of the weights. In *Proc. of the Sixth Annual Conference on Computational Learning Theory (COLT)*, pages 5–13, 1993.
- [42] D. J. C. MacKay. Developments in probabilistic modelling with neural networks - ensemble learning. In *Proc. of the 3rd Annual Symposium on Neural Networks*, pages 191–198, 1995.
- [43] D. Barber and C. M. Bishop. Ensemble learning for multi-layer networks. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 1997.
- [44] M. N. Gibbs and D. J. MacKay. Variational Gaussian process classifiers. *IEEE Transactions on Neural Networks*, 11(6):1458–1464, 2000.
- [45] C. M. Bishop. Variational learning in graphical models and neural networks. In *Proc. of the International Conference on Artificial Neural Networks (ICANN)*, pages 13–22. Springer, 1998.

- [46] A. E. Teschendorff, Y. Wang, N. L. Barbosa-Morais, J. D. Brenton, and C. Caldas. A variational Bayesian mixture modelling framework for cluster analysis of gene-expression data. *Bioinformatics*, 21(13):3025–3033, 2005.
- [47] A. Corduneanu and C. M. Bishop. Variational Bayesian model selection for mixture distributions. In *Proc. of the 8th International Conference on Artificial Intelligence and Statistics (AISTAT)*, pages 27–34, 2001.
- [48] B. Wang and D. M. Titterton. Convergence properties of a general algorithm for calculating variational Bayesian estimates for a normal mixture model. *Bayesian Analysis*, 1(3):625–650, 2006.
- [49] P. J. Green and D. J. Murdoch. Exact sampling for Bayesian inference: Towards general purpose algorithms (with discussions). In J. M. Bernardo et al., editor, *Bayesian Statistics 6*, pages 301–321, 1998.
- [50] D. J. Lunn, A. Thomas, N. Best and D. Spiegelhalter. WinBUGS - a Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing*, 10:325–337, 2000.
- [51] D. A. van Dyk and T. Park. Partially collapsed Gibbs samplers: Theory and methods. *Journal of the American Statistical Association*, 103(482):790–796, 2008.
- [52] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, 23(9):881–890, 1974.
- [53] P. E. Green, F. J. Carmone and J. Kim. A preliminary study of optimal variable weighting in k-means clustering. *Journal of Classification*, 7(2):271–285, 1990.
- [54] 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*, pages 1154–1166, 2004.

- [55] W. S. DeSarbo, J. D. Carroll, L. A. Clark and P. E. Green. Synthesized clustering: A method for amalgamating alternative clustering based with differential weighting of variables. *Psychometrika*, 49(1):57–78, 1984.
- [56] Y. Kim, W. N. Street and F. Menczer. Feature selection in unsupervised learning via evolutionary search. In *Proc. of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 365–369, 2000.
- [57] M. W. Graham and D. J. Miller. Unsupervised learning of parsimonious mixtures on large spaces with integrated feature and component selection. *IEEE Transactions on Signal Processing*, 54(4):1289–1303, 2006.
- [58] A. E. Raftery and N. Dean. Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473):168–178, 2006.
- [59] C. Constantinopoulos, M. K. Titsias and A. Likas. Bayesian feature and model selection for Gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):1013–1018, 2006.
- [60] S. Boutemedjet, N. Bouguila and D. Ziou. A hybrid feature extraction selection approach for high-dimensional non-Gaussian data clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1429–1443, 2009.
- [61] H. Lian. Sparse Bayesian hierarchical modeling of high-dimensional clustering problems. *Journal of Multivariate Analysis*, 101(7):1728–1737, 2010.
- [62] D. A. Bell and H. Wang. A formalism for relevance and its application in feature subset selection. *Machine Learning*, 41(2):175–195, 2000.
- [63] Z. Lu and H. H. S. Ip. Generalized competitive learning of Gaussian mixture models. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 39(4):901–909. 2009.
- [64] A. L. Yuille, P. Stolorz and J. Utans. Statistical physics, mixtures of distributions, and the EM algorithm. *Neural Computation*, 6(2):334–340, 1994.

- [65] Y. Li, M. Dong and J. Hua. Simultaneous localized feature selection and model detection for Gaussian mixtures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):953–960, 2009.
- [66] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [67] C. M. Bishop, N. Lawrence, T. Jaakola and M. I. Jordan. Approximating posterior distributions in belief networks using mixtures. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 1997.
- [68] N. D. Lawrence, C. M. Bishop, and M. I. Jordan. Mixture representations for inference and learning in boltzmann machines. In *Proc. of the 15th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 320–327, 1998.
- [69] G. Parisi. *Statistical Field Theory*. Addison-Wesley, 1988.
- [70] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [71] W. Fan, 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.
- [72] W. Fan, N. Bouguila, and D. Ziou. Unsupervised hybrid feature extraction selection for high-dimensional non-Gaussian data clustering with variational inference. *IEEE Transactions on Knowledge and Data Engineering*, 25(7):1670–1685, 2013.
- [73] W. Fan and N. Bouguila. Variational learning of a dirichlet process of generalized Dirichlet distributions for simultaneous clustering and feature selection. *Pattern Recognition*, 46(10):2754–2769, 2013.
- [74] W. Fan and N. Bouguila. Online learning of a Dirichlet process mixture of Beta-liouville distributions via variational inference. *IEEE Transactions on Neural Networks and Learning Systems*, 24(11):1850–1862, 2013.

- [75] N. Bouguila, D. Ziou, and J. Vaillancourt. Unsupervised learning of a finite mixture model based on the Dirichlet distribution and its application. *IEEE Transactions on Image Processing*, 13(11):1533–1543, 2004.
- [76] P. Diaconis and D. Ylvisaker. Conjugate priors for exponential families. *The Annals of Statistics*, 7:269–281, 1979.
- [77] E. Castillo, A.S. Hadi, and C. Solares. Learning and updating of uncertainty in Dirichlet models. *Machine Learning*, 26(1):43–63, 1997.
- [78] Z. Ma and A. Leijon. Bayesian estimation of Beta mixture models with variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:2160–2173, 2011.
- [79] D. J. C. Mackay. Probable networks and plausible predictions – a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3):469–505, 1995.
- [80] B. Wang and D. M. Titterton. Convergence and asymptotic normality of variational Bayesian approximations for exponential family models with missing values. In *Proc. of the 20th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 577–584, 2004.
- [81] N. Nasios and A. G. Bors. Variational learning for Gaussian mixture models. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 36(4):849–862, 2006.
- [82] V. Viitaniemi and J. Laaksonen. Techniques for still image scene classification and object detection. In *Proc. of the International Conference on Artificial Neural Networks (ICANN)*, pages 35–44. Springer, 2006.
- [83] J. S. Bonet and P. Viola. Structure driven image database retrieval. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 1997.
- [84] A. Quattoni, M. Collins and T. Darrell. Conditional random fields for object recognition. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 2004.

- [85] S. Boutemedjet, D. Ziou and N. Bouguila. Unsupervised feature selection for accurate recommendation of high-dimensional image data. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 177–184, 2007.
- [86] G. Csurka, C. R. Dance, L. Fan, J. Willamowski and C. Bray. Visual categorization with bags of keypoints. In *Proc. of 8th European Conference on Computer Vision (ECCV), workshop on Statistical Learning in Computer Vision*, 2004.
- [87] A. Bosch, A. Zisserman and X. Munoz. Scene classification via pLSA. In *Proc. of 9th European Conference on Computer Vision (ECCV)*, pages 517–530, 2006.
- [88] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196, 2001.
- [89] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [90] C. Elkan. Using the triangle inequality to accelerate k-means. In *Proc. of the 20th International Conference on Machine Learning (ICML)*, pages 147–153, 2003.
- [91] D. E. Denning. An intrusion-detection model. *IEEE Transactions on Software Engineering*, SE-13(2):222–232, 2006.
- [92] M. Bahrololum and M. Khaleghi. Anomaly intrusion detection system using hierarchical Gaussian mixture model. *International Journal of Computer Science and network Security*, 8(8):264–271, 2008.
- [93] S. Northcutt and J. Novak. *Network Intrusion Detection: An Analyst's Handbook*. New Riders Publishing, 2002.
- [94] A. Abraham S. Chebrolu and J. P. Thomas. Feature deduction and ensemble design of intrusion detection systems. *Computers and Security*, 24(4):295–307, 2005.
- [95] T. S. Jaakkola and M. I. Jordan. Variational probabilistic inference and the QMR-DT network. *Journal of Artificial Intelligence Research*, 10:291–322. 1999.

- [96] S. J. Roberts and W. D. Penny. Variational bayes for generalized autoregressive models. *IEEE Transactions on Signal Processing*, 50(9):2245–2257, 2002.
- [97] X. Zhou, X. Zhuang, S. Yan, S-F. Chang, M. Hasegawa-Johnson and T. S. Huang. SIFT-based kernel for video event analysis. In *Proc. of the ACM International Conference on Multimedia (MM)*, pages 229–238, 2008.
- [98] D. Zhong, H. Zhang and S-F. Chang. Clustering methods for video browsing and annotation. In *Proc. of the SPIE Conference on Storage and Retrieval for Video and Image Databases*, pages 239–246, 1997.
- [99] P. Dollár, V. Rabaud, G. Cottrell and S. Belongie. Behavior recognition via sparse spatio-temporal feature. In *Proc. of the IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pages 65–72, 2005.
- [100] J. C. Niebles, H. Wang and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *Proc. of the British Machine Vision Conference (BMVC)*, pages 1249–1258, 2006.
- [101] I. Laptev, M. Marszalek, C. Schmid and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.
- [102] C. Schödl, I. Laptev and B. Caputo. Recognizing human actions: A local SVM approach. In *Proc. of the 17th International Conference on Pattern Recognition (ICPR)*, pages 32–36, 2004.
- [103] S. Becker. Learning to categorize objects using temporal coherence. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 361–368, 1992.
- [104] A. Frome, Y. Singer and J. Malik. Image retrieval and classification using local distance functions. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 417–424, 2006.

- [105] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2/3):107–123, 2005.
- [106] N. Bouguila and D. Ziou. A hybrid SEM algorithm for high-dimensional unsupervised learning using a finite generalized Dirichlet mixture. *IEEE Transactions on Image Processing*, 15(9):2657–2668, sept. 2006.
- [107] R. M. Korwar and M. Hollander. Contributions to the theory of Dirichlet processes. *The Annals of Probability*, 1:705–711, 1973.
- [108] T. S. Ferguson. Bayesian density estimation by mixtures of normal distributions. *Recent Advances in Statistics*, 24:287–302, 1983.
- [109] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.
- [110] H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96:161–173, 2001.
- [111] H. Ishwaran and L. F. James. Some further developments for stick-breaking priors: Finite and infinite clustering and classification. *Shankhya: The Indian Journal of Statistics*, 65(3):577–592, 2003.
- [112] D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1:121–144, 2005.
- [113] J. M. Dickey. Multiple hypergeometric functions: Probabilistic interpretations and statistical uses. *Journal of the American Statistical Association*, 78(383):628–637, 1983.
- [114] Z. Su, H. Zhang, S. Li and S. Ma. Relevance feedback in content-based image retrieval: Bayesian framework, features subspaces, and progressive learning. *IEEE Transactions on Image Processing*, 12:924–937, 2003.
- [115] J. Matas, J. Buriánek, and J. Kittler. Object recognition using the invariant pixel-set signature. In *Proc. of the British Machine Vision Conference (BMVC)*, 2000.

- [116] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1465–1479, 2006.
- [117] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1605–1614. IEEE Computer Society, 2006.
- [118] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pages 1 –8. IEEE Computer Society, 2007.
- [119] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888 –905, 2000.
- [120] K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pages 1792 –1799 Vol. 2. IEEE Computer Society, 2005.
- [121] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60:63–86, 2004.
- [122] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages II–264 – II–271 vol.2. IEEE Computer Society, 2003.
- [123] R. Zhao and W. I. Grosky. From features to semantics: Some preliminary results. In *Proc. of the IEEE International Conference on Multimedia and Expo (ICME)*, pages II.679–II.682. IEEE Computer Society, 2000.
- [124] M. R. Naphade and T. S. Huang. A probabilistic framework for semantic video indexing, filtering, and retrieval. *IEEE Transactions on Multimedia*, 3:141–151, 2001.

- [125] E. Chang. CBSA: Content-based soft annotation for multinomial image retrieval using Bayes point machines. *IEEE Transactions on Circuit and Systems for Video Technology*, 13(1):26–38, 2003.
- [126] J. Luo, A. E. Savakis and A. Singhal. A Bayesian network-based framework for semantic image understanding. *Pattern Recognition*, 38:919–934, 2005.
- [127] J. Fan, Y. Gao, H. Luo and G. Xu. Statistical modeling and conceptualization of natural images. *Pattern Recognition*, 38:865–885, 2005.
- [128] P. H. Gosselin and M. Cord. Feature-based approach to semi-supervised similarity learning. *Pattern Recognition*, 39:1839–1851, 2006.
- [129] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pages II.408–II.415. IEEE Computer Society, 2001.
- [130] F. Monay and D. Gatica-Perez. On image auto-annotation with latent space models. In *Proc. of the eleventh ACM international conference on Multimedia (MM)*, pages 275–278. ACM, 2003.
- [131] E. P. Xing, R. Yan, and A. G. Hauptmann. Mining associated text and images with dual-wing harmoniums. In *Proc. of the Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 633–641. AUAI Press, 2005.
- [132] J. Li. A mutual semantic endorsement approach to image retrieval and context provision. In *Proc. of the 7th ACM SIGMM international workshop on Multimedia information retrieval (MIR)*, pages 173–182. ACM, 2005.
- [133] R. Datta, W. Ge, J. Li, and J. Z. Wang. Toward bridging the annotation-retrieval gap in image search by a generative modeling approach. In *Proc. of the 14th annual ACM international conference on Multimedia (MM)*, pages 977–986. ACM, 2006.

- [134] Y. Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 506–513, 2004.
- [135] G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41, 1995.
- [136] C. Leacock and M. Chodorow. *Combining local context and WordNet similarity for word sense identification*, pages 305–332. In C. Fellbaum (Ed.), MIT Press, 1998.
- [137] B. Russell, A. Torralba, K. Murphy, and W. Freeman. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77:157–173, 2008.
- [138] K. T. Fang, S. Kotz and K. W. Ng. *Symmetric Multivariate and Related Distributions*. Chapman and Hall, New York, 1990.
- [139] N. Bouguila. Count data modeling and classification using finite mixtures of distributions. *IEEE Transactions on Neural Networks*, 22(2):186–198, 2011.
- [140] C. E. Rasmussen. The infinite Gaussian mixture model. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 554–560, 2000.
- [141] M. Sato. Online model selection based on the variational Bayes. *Neural Computation*, 13:1649–1681, 2001.
- [142] M. D. Hoffman, D. M. Blei, and F. Bach. Online learning for latent Dirichlet allocation. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 856–864, 2010.
- [143] G. Littlewort, M. S. Bartlett, I. R. Fasel, J. Chenu, T. Kanda, H. Ishiguro, and J. R. Movellan. Towards social robots: Automatic evaluation of human-robot interaction by face detection and expression classification. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 2003.

- [144] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1424–1445, 2000.
- [145] M. Stewart Bartlett, G. Littlewort, B. Braathen, T. J. Sejnowski, and J. R. Movellan. A prototype for automatic recognition of spontaneous facial actions. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 1271–1278, 2002.
- [146] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman and T. J. Sejnowski. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):974–989, 1999.
- [147] P. Ekman and W. V. Friesen. *Unmasking the Face*. New Jersey: Prentice Hall, 1975.
- [148] P. Ekman and W. V. Friesen. *Facial Action Coding System (FACS): Manual*. Palo Alto: Consulting Psychologists Press, 1978.
- [149] P. Ekman. *Emotion in the Human Face*. Cambridge University Press, 1982.
- [150] M. Rosenblum, Y. Yacoob and L. S. Davis. Human expression recognition from motion using a radial basis function network architecture. *IEEE Transactions on Neural Networks*, 7(5):1121–1138, 1996.
- [151] Y. Yacoob and L. S. Davis. Recognizing human facial expressions from long image sequences using optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):636–642, 1996.
- [152] C. Padgett and G. W. Cottrell. Representing face images for emotion classification. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 894–900, 1996.
- [153] I. A. Essa and A. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):757–763, 1997.

- [154] M. J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.
- [155] Y-L. Tian, T. Kanade, and J. F. Cohn. Recognizing action units for facial expression analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(2):97–115, 2001.
- [156] I. Kotsia, I. Pitas, and S. Zafeiriou. Novel multiclass classifiers based on the minimization of the within-class variance. *IEEE Transactions on Neural Networks*, 20(1):14–34, 2009.
- [157] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.
- [158] C. Shan, S. Gong, and P. W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, 2009.
- [159] F. Cheng, J. Yu, and H. Xiong. Facial expression recognition in JAFFE dataset based on gaussian process classification. *IEEE Transactions on Neural Networks*, 21(10):1685–1690, 2010.
- [160] B. W. Mel. SEEMORE: Combining color, shape, and texture histogramming in a neurally-inspired approach to visual object recognition. *Neural Computation*, 9:777–804, 1997.
- [161] M. P. Kumar, P. H. S. Ton, and A. Zisserman. Obj cut. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18–25, 2005.
- [162] J. Winn and N. Jojic. LOCUS: Learning object classes with unsupervised segmentation. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 756–763, 2005.
- [163] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 985–992, 2006.

- [164] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, 2008.
- [165] Y. Zhang and Q. Ji. Active and dynamic information fusion for facial expression understanding from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):699–714, 2005.
- [166] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos “in the wild”. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1996–2003, 2009.
- [167] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.
- [168] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics, SIGGRAPH 2003*, 22(3):277–286, July 2003.
- [169] G. Zhao and M. Pietikäinen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, 2007.
- [170] A. Ravichandran, R. Chaudhry, and R. Vidal. Categorizing dynamic textures using a bag of dynamical systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):342–353, 2013.
- [171] R. Péteri, S. Fazekas, and M. J. Huiskes. DynTex : a Comprehensive Database of Dynamic Textures. *Pattern Recognition Letters*, 31(12):1627–1632, 2010.
- [172] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

- [173] Y. W. Teh and M. I. Jordan. Hierarchical Bayesian nonparametric models with applications. In N. Hjort, C. Holmes, P. Müller, and S. Walker, editors, *Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press, 2010.
- [174] Y. W. Teh and M. I. Jordan. *Hierarchical Bayesian Nonparametric Models with Applications*. Cambridge University Press, 2010.
- [175] M. W. Woolrich and T. E. Behrens. Variational Bayes inference of spatial mixture models for segmentation. *IEEE Transactions on Medical Imaging*, 25(10):1380–1391, 2006.

Appendix A

Proof of Equations (2.14) and (2.15)

According to Eq. (1.8), the general expression for the variational solution $Q_s(\Theta_s)$ can be written as

$$\ln Q_s(\Theta_s) = \langle \ln p(\mathcal{X}, \Theta) \rangle_{j \neq s} + \text{const.} \quad (\text{A.1})$$

where any terms that are independent of $Q_s(\Theta_s)$ are absorbed into the additive constant. Using the previous equation and the logarithm of joint distribution in Eq. (2.11), we develop the following variational solutions for $Q(\mathcal{Z})$ and $Q(\vec{\alpha})$.

A.1 Proof of Equation (2.14): Variational Solution to $Q(\mathcal{Z})$

$$\ln Q(Z_{ij}) = Z_{ij} [\ln \pi_j + \mathcal{R}_j + \sum_{l=1}^D (\bar{\alpha}_{jl} - 1) \ln X_{il}] + \text{const.} \quad (\text{A.2})$$

where

$$\mathcal{R}_j = \left\langle \ln \frac{\Gamma(\sum_{l=1}^D \alpha_{jl})}{\prod_{l=1}^D \Gamma(\alpha_{jl})} \right\rangle_{\alpha_{j1}, \dots, \alpha_{jD}}, \quad \bar{\alpha}_{jl} = \langle \alpha_{jl} \rangle = \frac{u_{jl}}{v_{jl}} \quad (\text{A.3})$$

Unfortunately, a closed-form expression cannot be found for \mathcal{R}_j , so the standard variational inference can not be applied directly. Therefore, we need to propose a lower bound approximation to obtain a closed-form expression. The second-order Taylor series expansion has been successfully applied in variational inference for providing tractable approximations [78, 175] and we shall use it here. Indeed, we approximate the function \mathcal{R}_j using a second-order Taylor expansion about the expected values of the parameters $\vec{\alpha}_j$. Let us define $\tilde{\mathcal{R}}_j$ to denote the approximation of \mathcal{R}_j , and $(\bar{\alpha}_{j1}, \dots, \bar{\alpha}_{jD})$ to represent the expected values of $\vec{\alpha}_j$. This lower bound approximation is given

by Eq. (2.18) and is proved in Appendix B. Then, the optimization in Eq. (A.2) becomes tractable after replacing \mathcal{R}_j by $\tilde{\mathcal{R}}_j$.

From Eq. (A.2), it is straightforward to see that the optimal solution to \mathcal{Z} has the logarithmic form of Eq. (2.4) except for the normalization constant. Thus, $\ln Q(\mathcal{Z})$ can be written as

$$\ln Q(\mathcal{Z}) = \sum_{i=1}^N \sum_{j=1}^M z_{ij} \ln \rho_{ij} + \text{const.} \quad (\text{A.4})$$

$$\ln \rho_{ij} = \ln \pi_j + \tilde{\mathcal{R}}_j + \sum_{l=1}^D (\tilde{\alpha}_{jl} - 1) \ln X_{il} \quad (\text{A.5})$$

Note that, any terms that do not depend on Z_{ij} can be absorbed into the constant part. If we take the exponential of both sides in Eq. (A.4), we obtain

$$Q(\mathcal{Z}) \propto \prod_{i=1}^N \prod_{j=1}^M \rho_{ij}^{Z_{ij}} \quad (\text{A.6})$$

This distribution needs to be normalized which can be performed as following

$$Q(\mathcal{Z}) = \prod_{i=1}^N \prod_{j=1}^M r_{ij}^{Z_{ij}}, \quad r_{ij} = \frac{\rho_{ij}}{\sum_{j=1}^M \rho_{ij}} \quad (\text{A.7})$$

Note that the $\{r_{ij}\}$ are nonnegative and sum to one. Therefore, we can obtain the standard result for $Q(\mathcal{Z})$ as

$$\langle Z_{ij} \rangle = r_{ij} \quad (\text{A.8})$$

where $\{r_{ij}\}$ are playing the role of responsibilities as in the conventional EM algorithm.

A.2 Proof of Equation (2.15): Variational Solution to $Q(\vec{\alpha})$

Since there are M components in the mixture model by considering the assumption that the parameters α_{jl} are independent, $Q(\vec{\alpha})$ can be factorized as

$$Q(\vec{\alpha}) = \prod_{j=1}^M \prod_{l=1}^D Q(\alpha_{jl}) \quad (\text{A.9})$$

Let us consider the variational optimization regarding the specific factor $Q(\alpha_{js})$. The logarithm of the optimized factor is given by

$$\ln Q(\alpha_{js}) = \sum_{i=1}^N r_{ij} \mathcal{J}(\alpha_{js}) + \alpha_{js} \sum_{i=1}^N r_{ij} \ln X_{is} + (u_{js} - 1) \ln \alpha_{js} - v_{js} \alpha_{js} + \text{const.} \quad (\text{A.10})$$

where

$$\mathcal{J}(\alpha_{js}) = \left\langle \ln \frac{\Gamma(\alpha_s + \sum_{l \neq s}^D \alpha_{jl})}{\Gamma(\alpha_s) \prod_{l \neq s}^D \Gamma(\alpha_{jl})} \right\rangle_{\Theta \neq \alpha_{js}} \quad (\text{A.11})$$

where $\mathcal{J}(\alpha_{js})$ is defined as a function of α_{js} and is unfortunately analytically intractable. Therefore, similar to \mathcal{R}_j in the previous subsection, we need to find a lower bound to approximate $\mathcal{J}(\alpha_{js})$ which we obtain via a first-order Taylor expansion [78] [66, chapter 10] about $\bar{\alpha}_{js}$ (the expected value of α_{js}) (see Appendix B):

$$\mathcal{J}(\alpha_{js}) \geq \bar{\alpha}_{js} \ln \alpha_{js} \left\{ \Psi\left(\sum_{l=1}^D \bar{\alpha}_{jl}\right) - \Psi(\bar{\alpha}_{js}) + \sum_{l \neq s}^D \bar{\alpha}_{jl} \Psi'\left(\sum_{l=1}^D \bar{\alpha}_{jl}\right) (\langle \ln \alpha_{jl} \rangle - \ln \bar{\alpha}_{jl}) \right\} + \text{const.} \quad (\text{A.12})$$

If we substitute this lower bound back into Eq. (A.10), we obtain a new optimal solution to α_{js} as

$$\begin{aligned} \ln Q(\alpha_{js}) &= \sum_{i=1}^N r_{ij} \bar{\alpha}_{js} \ln \alpha_{js} \left[\Psi\left(\sum_{l=1}^D \bar{\alpha}_{jl}\right) - \Psi(\bar{\alpha}_{js}) + \sum_{l \neq s}^D \Psi'\left(\sum_{l=1}^D \bar{\alpha}_{jl}\right) \bar{\alpha}_{jl} (\langle \ln \alpha_{jl} \rangle - \ln \bar{\alpha}_{jl}) \right] \\ &\quad + \alpha_{js} \sum_{i=1}^N r_{ij} \ln X_{is} + (u_{js} - 1) \ln \alpha_{js} - v_{js} \alpha_{js} + \text{const.} \\ &= \ln \alpha_{js} (u_{js} + \varphi_{js} - 1) - \alpha_{js} (v_{js} - \vartheta_{js}) + \text{const.} \end{aligned} \quad (\text{A.13})$$

where

$$\varphi_{js} = \sum_{i=1}^N r_{ij} \bar{\alpha}_{js} \left[\Psi\left(\sum_{l=1}^D \bar{\alpha}_{jl}\right) - \Psi(\bar{\alpha}_{js}) + \sum_{l \neq s}^D \Psi'\left(\sum_{l=1}^D \bar{\alpha}_{jl}\right) \bar{\alpha}_{jl} (\langle \ln \alpha_{jl} \rangle - \ln \bar{\alpha}_{jl}) \right] \quad (\text{A.14})$$

$$\vartheta_{js} = \sum_{i=1}^N r_{ij} \ln X_{is} \quad (\text{A.15})$$

We can see that Eq. (A.13) has the logarithmic form of a Gamma distribution. Taking the exponential of its both sides, we obtain

$$Q(\alpha_{js}) \propto \alpha_{js}^{u_{js} + \varphi_{js} - 1} e^{-(v_{js} - \vartheta_{js})\alpha_{js}} \quad (\text{A.16})$$

Therefore, we can obtain the optimal solutions to the hyper-parameters u_{js} and v_{js} as

$$u_{js}^* = u_{js} + \varphi_{js} , \quad v_{js}^* = v_{js} - \vartheta_{js} \quad (\text{A.17})$$

where φ_{js} and ϑ_{js} are given by Eqs. (2.20) and (2.21), respectively.

Appendix B

Proof of Equations (2.18) and (A.12)

B.1 Lower Bound of \mathcal{R}_j : Proof of Equation (2.18)

The function \mathcal{R}_j in Eq. (A.3) is analytically intractable, a non-linear approximation of the lower bound can be obtained by using the second order Taylor expansion as done in [78] where the authors have used the first and second Taylor expansions to approximate lower bounds for variational Beta mixture model. In our work, first, we define the following function

$$\mathcal{H}(\vec{\alpha}_j) = \mathcal{H}(\alpha_{j1}, \dots, \alpha_{jD}) = \ln \frac{\Gamma(\sum_{l=1}^D \alpha_{jl})}{\prod_{l=1}^D \Gamma(\alpha_{jl})} \quad (\text{B.1})$$

where $\alpha_{jl} > 1$. The lower bound of $\mathcal{H}(\vec{\alpha}_j)$ can be obtained by using the second order Taylor expansion for $\ln \vec{\alpha}_j = (\ln \alpha_{j1}, \dots, \ln \alpha_{jD})$ at $\ln \vec{\alpha}_{j,0} = (\ln \alpha_{j1,0}, \dots, \ln \alpha_{jD,0})$ as

$$\begin{aligned} \mathcal{H}(\vec{\alpha}_j) \geq & \mathcal{H}(\vec{\alpha}_{j,0}) + (\ln \vec{\alpha}_j - \ln \vec{\alpha}_{j,0})^T \nabla \mathcal{H}(\vec{\alpha}_{j,0}) \\ & + \frac{1}{2!} (\ln \vec{\alpha}_j - \ln \vec{\alpha}_{j,0})^T \nabla^2 \mathcal{H}(\vec{\alpha}_{j,0}) (\ln \vec{\alpha}_j - \ln \vec{\alpha}_{j,0}) \end{aligned} \quad (\text{B.2})$$

where $\nabla \mathcal{H}(\vec{\alpha}_{j,0})$ represents the gradient of \mathcal{H} evaluated at $\vec{\alpha}_j = \vec{\alpha}_{j,0}$ and $\nabla^2 \mathcal{H}(\vec{\alpha}_{j,0})$ is the Hessian matrix. This gives

$$\begin{aligned} \mathcal{H}(\vec{\alpha}_j) \geq & \mathcal{H}(\vec{\alpha}_{j,0}) + \sum_{l=1}^D \frac{\partial \mathcal{H}(\vec{\alpha}_j)}{\partial \ln \alpha_{jl}} \Big|_{\vec{\alpha}_j = \vec{\alpha}_{j,0}} (\ln \alpha_{jl} - \ln \alpha_{jl,0}) \\ & + \frac{1}{2} \sum_{a=1}^D \sum_{b=1}^D \frac{\partial^2 \mathcal{H}(\vec{\alpha}_j)}{\partial \ln \alpha_{ja} \partial \ln \alpha_{jb}} \Big|_{\vec{\alpha}_j = \vec{\alpha}_{j,0}} (\ln \alpha_{ja} - \ln \alpha_{ja,0}) (\ln \alpha_{jb} - \ln \alpha_{jb,0}) \end{aligned} \quad (\text{B.3})$$

Then, the lower bound of the function \mathcal{R}_j can be obtained by taking the expectation of Eq. (B.3) with respect to $\vec{\alpha}_j$ as

$$\begin{aligned}
\mathcal{R}_j \geq \tilde{\mathcal{R}}_j &= \ln \frac{\Gamma(\sum_{l=1}^D \alpha_{jl,0})}{\prod_{l=1}^D \Gamma(\alpha_{jl,0})} \\
&+ \sum_{l=1}^D \alpha_{jl,0} [\Psi(\sum_{l=1}^D \alpha_{jl,0}) - \Psi(\alpha_{jl,0})] [\langle \ln \alpha_{jl} \rangle - \ln \alpha_{jl,0}] \\
&+ \frac{1}{2} \sum_{l=1}^D \alpha_{jl,0}^2 [\Psi'(\sum_{l=1}^D \alpha_{jl,0}) - \Psi'(\alpha_{jl,0})] \langle (\ln \alpha_{jl} - \ln \alpha_{jl,0})^2 \rangle \\
&+ \frac{1}{2} \sum_{a=1}^D \sum_{b=1(a \neq b)}^D \left\{ \alpha_{ja,0} \alpha_{jb,0} \Psi'(\sum_{l=1}^D \alpha_{jl,0}) (\langle \ln \alpha_{ja} \rangle - \ln \alpha_{ja,0}) (\langle \ln \alpha_{jb} \rangle - \ln \alpha_{jb,0}) \right\}
\end{aligned} \tag{B.4}$$

In order to prove that the second order Taylor expansion of $\mathcal{H}(\vec{\alpha}_j)$ is indeed a lower bound of $\mathcal{H}(\vec{\alpha}_j)$, we need to show that $\Delta \mathcal{H}(\vec{\alpha}_j) \geq 0$, where $\Delta \mathcal{H}(\vec{\alpha}_j)$ denotes the difference between $\mathcal{H}(\vec{\alpha}_j)$ and its second order Taylor expansion. The Hessian of $\Delta \mathcal{H}(\vec{\alpha}_j)$ with respect to $(\ln \alpha_{j1}, \dots, \ln \alpha_{jD})$ is given by Eq. (B.5). By substituting $(\ln \alpha_{j1}, \dots, \ln \alpha_{jD})$ with the critical point $(\ln \alpha_{j1,0}, \dots, \ln \alpha_{jD,0})$,

$$\text{Hess} = \begin{bmatrix} \alpha_{j1} [\Psi(\sum_{l=1}^D \alpha_{jl}) - \Psi(\alpha_{j1})] & & & & & & \\ +\alpha_{j1}^2 [\Psi'(\sum_{l=1}^D \alpha_{jl}) - \Psi'(\alpha_{j1})] & \dots & \alpha_{j1} \alpha_{jD} \Psi'(\sum_{l=1}^D \alpha_{jl}) - \bar{\alpha}_{j1} \bar{\alpha}_{jD} \Psi'(\sum_{l=1}^D \bar{\alpha}_{jl}) & & & & \\ -\bar{\alpha}_{j1}^2 [\Psi'(\sum_{l=1}^D \bar{\alpha}_{jl}) - \Psi'(\bar{\alpha}_{j1})] & & & & & & \\ \vdots & \ddots & & & \vdots & & \\ \alpha_{j1} \alpha_{jD} \Psi'(\sum_{l=1}^D \alpha_{jl}) - \bar{\alpha}_{j1} \bar{\alpha}_{jD} \Psi'(\sum_{l=1}^D \bar{\alpha}_{jl}) \dots & & & & \alpha_{jD} [\Psi(\sum_{l=1}^D \alpha_{jl}) - \Psi(\alpha_{jD})] & & \\ & & & & +\alpha_{jD}^2 [\Psi'(\sum_{l=1}^D \alpha_{jl}) - \Psi'(\alpha_{jD})] & & \\ & & & & -\bar{\alpha}_{jD}^2 [\Psi'(\sum_{l=1}^D \bar{\alpha}_{jl}) - \Psi'(\bar{\alpha}_{jD})] & & \end{bmatrix} \tag{B.5}$$

Eq. (B.5) is reduced to a positive definite diagonal matrix. Since $(\ln \alpha_{j1,0}, \dots, \ln \alpha_{jD,0})$ is the only critical point and $\Delta \mathcal{H}(\vec{\alpha}_j)$ is continuous and differentiable through all α_{jl} (for $\alpha_{jl} > 1$), the critical point $(\ln \alpha_{j1,0}, \dots, \ln \alpha_{jD,0})$ is also the global minimum of $\Delta \mathcal{H}(\vec{\alpha}_j)$. The global minimum value 0 is reached when $(\ln \alpha_{j1}, \dots, \ln \alpha_{jD}) = (\ln \alpha_{j1,0}, \dots, \ln \alpha_{jD,0})$. Therefore, the second order Taylor expansion is indeed a lower bound.

B.2 Lower Bound of $\mathcal{J}(\alpha_{js})$: Proof of Equation (A.12)

Since the first order Taylor expansion of a convex function is a tangent line of that function at a specific value, the lower bound of $\mathcal{J}(\alpha_{js})$ in Eq. (A.11) can be approximated by a first order Taylor expansion. In [78], the authors evaluate the lower bound of the Log-inverse-Beta function by using the first order Taylor expansion. In our work, we extent this idea to the multivariate case. Let us define the function $\mathcal{F}(\alpha_{js})$ as

$$\mathcal{F}(\alpha_{js}) = \ln \frac{\Gamma(\alpha_{js} + \sum_{l \neq s}^D \alpha_{jl})}{\Gamma(\alpha_{js}) \prod_{l \neq s}^D \Gamma(\alpha_{jl})} \quad (\text{B.6})$$

B.2.1 Convexity of $\mathcal{F}(\alpha_{js})$

It is not straightforward to show directly that $\mathcal{F}(\alpha_{js})$ is a convex function of α_{js} . Yet, by adopting the *relative convexity* as in [78], we can show that $\mathcal{F}(\alpha_{js})$ is convex relative to $\ln \alpha_{js}$. A function is considered to be convex on an interval if and only if its second derivative is non-negative there. The first derivative of $\mathcal{F}(\alpha_{js})$ with respect to $\ln \alpha_{js}$ is

$$\frac{\partial \mathcal{F}(\alpha_{js})}{\partial \ln \alpha_{js}} = \left[\Psi(\alpha_{js} + \sum_{l \neq s}^D \alpha_{jl}) - \psi(\alpha_{js}) \right] \alpha_{js} \quad (\text{B.7})$$

Then, the second derivative with respect to $\ln \alpha_{js}$ is

$$\begin{aligned} \frac{\partial^2 \mathcal{F}(\alpha_{js})}{\partial (\ln \alpha_{js})^2} &= \left[\Psi(\alpha_{js} + \sum_{l \neq s}^D \alpha_{jl}) - \Psi(\alpha_{js}) \right] \alpha_{js} + \left[\Psi'(\alpha_{js} + \sum_{l \neq s}^D \alpha_{jl}) - \Psi'(\alpha_{js}) \right] \alpha_{js}^2 \\ &= \alpha_{js} \int_0^\infty \frac{1 - e^{-(\sum_{l \neq s}^D \alpha_{jl})t}}{1 - e^{-t}} e^{-\alpha_{js}t} (1 - \alpha_{js}t) dt \end{aligned} \quad (\text{B.8})$$

where the integral representations of $\Psi(x)$ and $\Psi'(x)$ are defined by

$$\Psi(x) = \int_0^\infty \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1 - e^{-t}} \right) dt \quad (\text{B.9})$$

and

$$\Psi'(x) = \int_0^\infty \frac{te^{-xt}}{1 - e^{-t}} dt \quad (\text{B.10})$$

We can re-write Eq. (B.8) as

$$\frac{\partial^2 \mathcal{F}(\alpha_{js})}{\partial (\ln \alpha_{js})^2} = \alpha_{js} \int_0^\infty f_1(t) f_2(t) dt \quad (\text{B.11})$$

where

$$f_1(t) = \frac{1 - e^{-(\sum_{l \neq s}^D \alpha_{jl})t}}{1 - e^{-t}} \quad (\text{B.12})$$

$$f_2(t) = e^{-\alpha_{js}t} (1 - \alpha_{js}t) \quad (\text{B.13})$$

By analyzing Eqs. (B.12) and (B.13), we can find that when $\sum_{l \neq s}^D \alpha_{jl} > 1$: if $t > 1/\alpha_{js}$, then $f_1(t) < f_1(1/\alpha_{js})$ and $f_2(t) < 0$; if $t < 1/\alpha_{js}$, then $f_1(t) > f_1(1/\alpha_{js})$ and $f_2(t) > 0$. Hence, we can rewrite Eq. (B.11) as

$$\begin{aligned} \frac{\partial^2 \mathcal{F}(\alpha_{js})}{\partial (\ln \alpha_{js})^2} &= \alpha_{js} \left\{ \int_0^{\frac{1}{\alpha_{js}}} f_1(t) f_2(t) dt + \int_{\frac{1}{\alpha_{js}}}^\infty f_1(t) f_2(t) dt \right\} \\ &> \alpha_{js} \left\{ \int_0^{\frac{1}{\alpha_{js}}} f_1\left(\frac{1}{\alpha_{js}}\right) f_2(t) dt + \int_{\frac{1}{\alpha_{js}}}^\infty f_1\left(\frac{1}{\alpha_{js}}\right) f_2(t) dt \right\} \\ &= \alpha_{js} f_1\left(\frac{1}{\alpha_{js}}\right) \int_0^\infty f_2(t) dt \\ &= \alpha_{js} f_1\left(\frac{1}{\alpha_{js}}\right) \lim_{t \rightarrow \infty} t e^{-\alpha_{js}t} = 0 \end{aligned} \quad (\text{B.14})$$

Therefore, when $\sum_{l \neq s}^D \alpha_{jl} > 1$, the convexity of $\mathcal{F}(\alpha_{js})$ relative to $\ln \alpha_{js}$ is proved.

B.2.2 Evaluating Lower Bound by The First Order Taylor Expansion

Since $\mathcal{F}(\alpha_{js})$ is a convex function relative to $\ln \alpha_{js}$, its lower bound can be obtained by applying the first order Taylor expansion of $\mathcal{F}(\alpha_{js})$ for $\ln \alpha_{js}$ at $\ln \alpha_{js,0}$ as following

$$\begin{aligned} \mathcal{F}(\alpha_{js}) &\geq \mathcal{F}(\alpha_{js,0}) + \frac{\partial \mathcal{F}(\alpha_{js})}{\partial \ln \alpha_{js}} \Big|_{\alpha_{js}=\alpha_{js,0}} (\ln \alpha_{js} - \ln \alpha_{js,0}) \\ &= \mathcal{F}(\alpha_{js,0}) + \frac{\partial \mathcal{F}(\alpha_{js})}{\partial \alpha_{js}} \frac{\partial \alpha_{js}}{\partial \ln \alpha_{js}} \Big|_{\alpha_{js}=\alpha_{js,0}} (\ln \alpha_{js} - \ln \alpha_{js,0}) \\ &= \ln \frac{\Gamma(\alpha_{js,0} + \sum_{l \neq s}^D \alpha_{jl})}{\Gamma(\alpha_{js,0}) \prod_{l \neq s}^D \Gamma(\alpha_{jl})} + [\Psi(\alpha_{js,0} + \sum_{l \neq s}^D \alpha_{jl}) - \Psi(\alpha_{js,0})] \alpha_{js,0} (\ln \alpha_{js} - \ln \alpha_{js,0}) \end{aligned} \quad (\text{B.15})$$

Note that we reach the equality when $\alpha_{js} = \bar{\alpha}_{js}$. By substituting Eq. (B.15) into Eq. (A.11), we obtain

$$\begin{aligned} \mathcal{J}(\alpha_{js}) &\geq \left\langle \ln \frac{\Gamma(\alpha_{js,0} + \sum_{l \neq s}^D \alpha_{jl})}{\Gamma(\alpha_{js,0}) \prod_{l \neq s}^D \Gamma(\alpha_{jl})} + \left[\Psi(\alpha_{js,0} + \sum_{l \neq s}^D \alpha_{jl}) - \Psi(\alpha_{js,0}) \right] \alpha_{js,0} (\ln \alpha_{js} - \ln \alpha_{js,0}) \right\rangle_{\bar{\alpha} \neq \alpha_{js}} \\ &= \ln \alpha_{js} \alpha_{js,0} \left\{ \left\langle \Psi(\alpha_{js,0} + \sum_{l \neq s}^D \alpha_{jl}) \right\rangle_{\bar{\alpha} \neq \alpha_{js}} - \Psi(\alpha_{js,0}) \right\} + \text{const.} \end{aligned} \quad (\text{B.16})$$

We can notice that in Eq. (B.16), the calculation of the expectation $\langle \Psi(\alpha_{js,0} + \sum_{l \neq s}^D \alpha_{jl}) \rangle_{\bar{\alpha} \neq \alpha_{js}}$ is also analytically intractable. Using a similar proof as shown in Appendix B.2.1, it is straightforward to conclude that $\Psi(\alpha_{js,0} + \sum_{l \neq s}^D \alpha_{jl})$ is a convex function relative to $\ln \alpha_{jl,0}$, for $l = \{1, \dots, D\}$ and $l \neq s$. We can apply a first order Taylor expansion for the function $\Psi(\sum_{i=1}^n x_i + y)$ at $\ln \hat{\mathbf{x}}$, where $\hat{\mathbf{x}} = (\hat{x}_1, \dots, \hat{x}_n)$, to obtain its lower bound as

$$\Psi\left(\sum_{i=1}^n x_i + y\right) \geq \psi\left(\sum_{i=1}^n \hat{x}_i + y\right) + \sum_{i=1}^n (\ln x_i - \ln \hat{x}_i) \Psi'\left(\sum_{i=1}^n \hat{x}_i + y\right) \hat{x}_i \quad (\text{B.17})$$

Using the previous equation, the approximation lower bound of expectation $\langle \Psi(\alpha_{jl,0} + \sum_{l \neq s}^D \alpha_{jl}) \rangle_{\bar{\alpha} \neq \alpha_{js}}$ is given by

$$\left\langle \Psi\left(\sum_{l \neq s}^D \alpha_{jl} + \alpha_{js,0}\right) \right\rangle_{\bar{\alpha} \neq \alpha_{js}} \geq \Psi\left(\sum_{l=1}^D \alpha_{jl,0}\right) + \sum_{l \neq s}^D \alpha_{jl,0} \Psi'\left(\sum_{l=1}^D \alpha_{jl,0}\right) (\langle \ln \alpha_{jl} \rangle - \ln \alpha_{jl,0}) \quad (\text{B.18})$$

Finally, the lower bound of $\mathcal{J}(\alpha_{js})$ can be calculated by substituting Eq. (B.18) back into Eq. (B.16):

$$\mathcal{J}(\alpha_{js}) \geq \ln \alpha_{js} \alpha_{js,0} \left\{ \Psi\left(\sum_{l=1}^D \alpha_{jl,0}\right) - \Psi(\alpha_{js,0}) + \sum_{l \neq s}^D \alpha_{jl,0} \Psi'\left(\sum_{l=1}^D \alpha_{jl,0}\right) (\langle \ln \alpha_{jl} \rangle - \ln \alpha_{jl,0}) \right\} + \text{const.} \quad (\text{B.19})$$

Appendix C

Variational Learning of Online Infinite Beta-Liouville Mixture

C.1 Variational lower bound $\mathcal{L}(Q)$

By substituting Eqs. (5.11), (5.12) and (5.13) into Eq. (5.8), we can obtain the parameterized form of the lower bound $\mathcal{L}(Q)$ as

$$\begin{aligned}\mathcal{L}(Q) &= \sum_{\mathcal{Z}} \int Q(\mathcal{Z}, \Lambda) \ln \left\{ \frac{p(\mathcal{X}, \mathcal{Z}, \Lambda)}{Q(\mathcal{Z}, \Lambda)} \right\} d\Lambda \\ &= \langle \ln p(\mathcal{X} | \mathcal{Z}, \vec{\lambda}, \vec{\alpha}_d, \vec{\alpha}, \vec{\beta}) \rangle + \langle \ln p(\mathcal{Z} | \vec{\lambda}) \rangle + \langle \ln p(\vec{\lambda}) \rangle \\ &\quad + \langle \ln p(\vec{\alpha}_d) \rangle + \langle \ln p(\vec{\alpha}) \rangle + \langle \ln p(\vec{\beta}) \rangle - \langle \ln Q(\mathcal{Z}) \rangle \\ &\quad - \langle \ln Q(\vec{\lambda}) \rangle - \langle \ln Q(\vec{\alpha}_d) \rangle - \langle \ln Q(\vec{\alpha}) \rangle - \langle \ln Q(\vec{\beta}) \rangle\end{aligned}\tag{C.1}$$

C.2 Variational solution to $Q(\mathcal{Z})$

We calculate the variational parameter r_{ij} by setting the derivative of $\mathcal{L}(Q)$ in Eq. (C.1) with respect to r_{ij} to 0. Notice that we must take account of the constraint that $\sum_{j=1}^M r_{ij} = 1$. This can be achieved by adding a Lagrange multiplier φ to $\mathcal{L}(Q)$. Taking the derivative with respect to r_{ij}

and setting the result to zero, we get

$$\begin{aligned}
\frac{\partial \mathcal{L}(Q)}{\partial r_{ij}} = & (\bar{\alpha}_j - \sum_{d=1}^D \bar{\alpha}_{jd}) \ln \left(\sum_{d=1}^D X_{id} \right) + \sum_{d=1}^D (\bar{\alpha}_{jd} - 1) \ln X_{id} \\
& + (\bar{\beta}_j - 1) \ln \left(1 - \sum_{d=1}^D X_{id} \right) + \sum_{s=1}^{j-1} \langle \ln(1 - \lambda_s) \rangle - (\ln r_{ij} + 1) + \varphi \\
& + \mathcal{S}_j + \mathcal{H}_j + \langle \ln \lambda_j \rangle
\end{aligned} \tag{C.2}$$

where $\mathcal{S}_j = \langle \ln \frac{\Gamma(\sum_{d=1}^D \alpha_{jd})}{\prod_{d=1}^D \Gamma(\alpha_{jd})} \rangle$, and $\mathcal{H}_j = \langle \ln \frac{\Gamma(\alpha_j + \beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)} \rangle$. Since \mathcal{S}_j and \mathcal{H}_j are analytically intractable, we apply Taylor expansion to calculate lower bound approximations to these terms to obtain closed-form expressions. This is motivated by the fact that the first-order and second-order Taylor series expansion techniques have been successfully applied in variational inference for providing tractable approximations in many works [71, 78]. Thus, the second-order Taylor expansion technique is used to approximate the function \mathcal{S}_j about $\bar{\alpha}_{jd}$ (the expected value of α_{jd}), and to approximate \mathcal{H}_j about $\bar{\alpha}_j$ and $\bar{\beta}_j$ (the expected values of α_j and β_j) as

$$\begin{aligned}
\mathcal{H}_j = & \ln \frac{\Gamma(\bar{\alpha}_j + \bar{\beta}_j)}{\Gamma(\bar{\alpha}_j)\Gamma(\bar{\beta}_j)} + \bar{\alpha}_j [\psi(\bar{\alpha}_j + \bar{\beta}_j) - \psi(\bar{\alpha}_j)] (\langle \ln \alpha_j \rangle - \ln \bar{\alpha}_j) \\
& + \bar{\beta}_j [\psi(\bar{\alpha}_j + \bar{\beta}_j) - \psi(\bar{\beta}_j)] (\langle \ln \beta_j \rangle - \ln \bar{\beta}_j) \\
& + \frac{1}{2} \bar{\alpha}_j^2 [\psi'(\bar{\alpha}_j + \bar{\beta}_j) - \psi'(\bar{\alpha}_j)] \langle (\ln \alpha_j - \ln \bar{\alpha}_j)^2 \rangle \\
& + \frac{1}{2} \bar{\beta}_j^2 [\psi'(\bar{\alpha}_j + \bar{\beta}_j) - \psi'(\bar{\beta}_j)] \langle (\ln \beta_j - \ln \bar{\beta}_j)^2 \rangle \\
& + \bar{\alpha}_j \bar{\beta}_j \psi'(\bar{\alpha}_j + \bar{\beta}_j) (\langle \ln \alpha_j \rangle - \ln \bar{\alpha}_j) (\langle \ln \beta_j \rangle - \ln \bar{\beta}_j)
\end{aligned} \tag{C.3}$$

$$\begin{aligned}
\mathcal{S}_j = & \ln \frac{\Gamma(\sum_{d=1}^D \bar{\alpha}_{jd})}{\prod_{d=1}^D \Gamma(\bar{\alpha}_{jd})} \\
& + \sum_{d=1}^D \bar{\alpha}_{jd} [\Psi(\sum_{d=1}^D \bar{\alpha}_{jd}) - \Psi(\bar{\alpha}_{jd})] [\langle \ln \alpha_{jd} \rangle - \ln \bar{\alpha}_{jd}] \\
& + \frac{1}{2} \sum_{d=1}^D \bar{\alpha}_{jd}^2 [\Psi'(\sum_{l=1}^D \bar{\alpha}_{jl}) - \Psi'(\bar{\alpha}_{jd})] \langle (\ln \alpha_{jd} - \ln \bar{\alpha}_{jd})^2 \rangle \\
& + \frac{1}{2} \sum_{a=1}^D \sum_{\substack{b=1 \\ (b \neq a)}}^D [\bar{\alpha}_{ja} \bar{\alpha}_{jb} \Psi'(\sum_{d=1}^D \bar{\alpha}_{jd})] (\langle \ln \alpha_{ja} \rangle - \ln \bar{\alpha}_{ja}) (\langle \ln \alpha_{jb} \rangle - \ln \bar{\alpha}_{jb})
\end{aligned} \tag{C.4}$$

By substituting Eq. (C.4) and Eq. (C.3) back into Eq. (C.2), we then have

$$\begin{aligned}
\varphi=1 - \ln \sum_{j=1}^M \exp \left\{ \mathcal{S}_j + \mathcal{H}_j + (\bar{\alpha}_j - \sum_{d=1}^D \bar{\alpha}_{jd}) \ln \left(\sum_{d=1}^D X_{id} \right) \right. \\
+ (\bar{\beta}_j - 1) \ln \left(1 - \sum_{d=1}^D X_{id} \right) + \sum_{d=1}^D (\bar{\alpha}_{jd} - 1) \ln X_{id} \\
\left. + \langle \ln \lambda_j \rangle + \sum_{s=1}^{j-1} \langle \ln(1 - \lambda_s) \rangle \right\} \quad (C.5)
\end{aligned}$$

Then, by substituting Eq. (C.5) back into Eq. (C.2), we can obtain the variational solution to r_{ij} as shown in Eq. (5.14).

C.3 Variational solution to $Q(\vec{\lambda})$

For the variational factor $Q(\vec{\lambda})$, instead of using the gradient method, it is more straightforward to use Eq. (5.10) to compute the variational solution. Notice that these two method have equivalent results for variational inference. Therefore, the logarithm of $Q(\vec{\lambda})$ is given by

$$\ln Q(\lambda_j) = \ln \lambda_j \sum_{i=1}^N \langle Z_{ij} \rangle + \ln(1 - \lambda_j) \left(\sum_{i=1}^N \sum_{s=j+1}^M \langle Z_{is} \rangle + \langle \psi_j \rangle - 1 \right) + \text{Const.} \quad (C.6)$$

It is obvious that Eq. (C.6) has the logarithmic form of a Beta distribution as its conjugate prior distribution Eq. (5.7). By taking the exponential of its both sides, we obtain the variational solution to $Q(\vec{\lambda})$ as in Eq. (5.11).

C.4 Variational solutions to $Q(\vec{\alpha}_d)$, $Q(\vec{\alpha})$ and $Q(\vec{\beta})$

The logarithm form of the variational factor $Q(\vec{\alpha}_d)$ is given by

$$\ln Q(\alpha_{jd}) = \sum_{i=1}^N \langle Z_{ij} \rangle \left[\mathcal{B}_{jd} - \alpha_{jd} \ln \left(\sum_{d=1}^D X_{id} \right) + \alpha_{jd} \ln X_{id} \right] + (u_{jd} - 1) \ln \alpha_{jd} - v_{jd} \alpha_{jd} + \text{const.} \quad (C.7)$$

Since the term $\mathcal{B}_{jd} = \left\langle \ln \frac{\Gamma(\alpha_{jd} + \sum_{l \neq d}^D \alpha_{jl})}{\Gamma(\alpha_{jd}) \prod_{l \neq d}^D \Gamma(\alpha_{jl})} \right\rangle_{\neq \alpha_{jd}}$ is analytically intractable, we can not perform the variational inference directly and Eq. (C.7) does not have the same form as the logarithm of a

Gamma distribution as its conjugate prior. Thus, we approximate it using the first-order Taylor expansion as

$$\mathcal{B}_{jd} \simeq \ln \alpha_{jd} \bar{\alpha}_{jd} \left[\Psi \left(\sum_{d=1}^D \bar{\alpha}_{jd} \right) - \Psi(\bar{\alpha}_{jd}) + \Psi' \left(\sum_{d=1}^D \bar{\alpha}_{jd} \right) \sum_{l \neq d}^D (\langle \ln \alpha_{jl} \rangle - \ln \bar{\alpha}_{jl}) \bar{\alpha}_{jl} \right] + \text{const.} \quad (\text{C.8})$$

By substituting Eq. (C.8) back into Eq. (C.7), we have

$$\begin{aligned} \ln Q(\alpha_{jd}) \approx & \ln \alpha_{jd} \left\{ \sum_{i=1}^N \langle Z_{ij} \rangle \bar{\alpha}_{jd} \left[\Psi \left(\sum_{d=1}^D \bar{\alpha}_{jd} \right) - \Psi(\bar{\alpha}_{jd}) \right. \right. \\ & \left. \left. + \Psi' \left(\sum_{d=1}^D \bar{\alpha}_{jd} \right) \sum_{l \neq d}^D (\langle \ln \alpha_{jl} \rangle - \ln \bar{\alpha}_{jl}) \bar{\alpha}_{jl} \right] + u_{jd} - 1 \right\} \\ & - \alpha_{jd} \left\{ v_{jd} - \sum_{i=1}^N \langle Z_{ij} \rangle \left[\ln X_{id} - \ln \left(\sum_{d=1}^D X_{id} \right) \right] \right\} + \text{const.} \end{aligned} \quad (\text{C.9})$$

We can see that Eq. (C.9) has the logarithmic form of a Gamma distribution. By taking the exponential of both sides of Eq. (C.9), we then have the variational solutions to $Q(\vec{\alpha}_d)$ in Eq. (5.12).

Since $\vec{\alpha}$ and $\vec{\beta}$ also have Gamma prior, it is straightforward to obtain the variational solutions to $Q(\vec{\alpha})$ and $Q(\vec{\beta})$ in a similar way as for $Q(\vec{\alpha}_d)$.