Statistical Analysis of Spherical Data: Clustering, Feature Selection and Applications

Ola Amayri

A Thesis
in
The Department
of
Electrical and Computer Engineering

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

May 2014

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Ola Amayri

Entitled: Statistical Analysis of Spherical Data: Clustering, Feature Selection and Applications

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:

Dr. Ali Akgunduz Chair

Dr. Hichem Frigui External Examiner

Dr. Hoi Dick Ng External to Program

Dr. Benjamin Fung Examiner

Dr. Abdessamad Ben Hamza Examiner

Dr. Nizar Bouguila Thesis Supervisor

Approved by

Chair of Department or Graduate Program Director

2014

Dean of Faculty

Abstract

Statistical Analysis of Spherical Data: Clustering, Feature Selection and Applications

Ola Amayri, Ph.D.

Concordia University, 2014

In the light of interdisciplinary applications, data to be studied and analyzed have witnessed a growth in volume and change in their intrinsic structure and type. In other words, in practice the diversity of resources generating objects have imposed several challenges for decision maker to determine informative data in terms of time, model capability, scalability and knowledge discovery. Thus, it is highly desirable to be able to extract patterns of interest that support the decision of data management. Clustering, among other machine learning approaches, is an important data engineering technique that empowers the automatic discovery of similar object's clusters and the consequent assignment of new unseen objects to appropriate clusters. In this context, the majority of current research does not completely address the true structure and nature of data for particular application at hand. In contrast to most previous research, our proposed work focuses on the modeling and classification of spherical data that are naturally generated in many data mining and knowledge discovery applications. Thus, in this thesis we propose several estimation and feature selection frameworks based on Langevin distribution which are devoted to spherical patterns in offline and online settings.

In this thesis, we first formulate a unified probabilistic framework, where we build probabilistic kernels based on Fisher score and information divergences from finite Langevin mixture for Support Vector Machine. We are motivated by the fact that the blending of generative and discriminative approaches has prevailed by exploring and adopting distinct characteristic of each approach toward constructing a complementary system combining the best of both.

Due to the high demand to construct compact and accurate statistical models that are automatically adjustable to dynamic changes, next in this thesis, we propose probabilistic frameworks for high-dimensional spherical data modeling based on finite Langevin mixtures that allow simultaneous clustering and feature selection in offline and online settings. To this end, we adopted finite mixture models which have long been heavily relied on deterministic learning approaches such as maximum likelihood estimation. Despite their successful utilization in wide spectrum of areas, these approaches have several drawbacks as we will discuss in this thesis. An alternative approach is the adoption of Bayesian inference that naturally addresses data uncertainty while ensuring good generalization. To address this issue, we also propose a Bayesian approach for finite Langevin mixture model estimation and selection.

When data change dynamically and grow drastically, finite mixture is not always a feasible solution. In contrast with previous approaches, which suppose an unknown finite number of mixture components, we finally propose a nonparametric Bayesian approach which assumes an infinite number of components. We further enhance our model by simultaneously detecting informative features in the process of clustering.

Through extensive empirical experiments, we demonstrate the merits of the proposed learning frameworks on diverse high dimensional datasets and challenging real-world applications.

Acknowledgements

I would like to express my deepest gratitude to my thesis advisor Dr. Nizar Bouguila. This work would not be possible without his constant support and guidance.

I would like to thank the members of the examination committee for their feedback, insightful comments and helpful suggestions through my research. I, also, thank my fellow lab-mates at Concordia University. The completion of this research was made possible thanks to Alexander Graham Bell Canada Graduate Scholarships in Natural Sciences and Engineering Research Council of Canada (NSERC CGS) and Le Fonds québécois de la recherche sur la nature et les technologies (FQRNT).

Last but not least, I would like to thank my husband, who never stopped believing in me, my dearest parents, sisters and brothers, especially Abeer, for their love and support over the years.

Table of Contents

List of Tables ix

List of Figures xi

List of Symbols xiii

1 Introduction 1

1.1 Background . . . 2

1.1.1 Langevin Distribution . . . 2

1.1.2 Finite Langevin Mixture Model . . . 4

1.2 Contributions . . . 5

1.3 Thesis Overview . . . 5

2 Hybrids of Generative Discriminative Models for Spherical Data 7

2.1 Introduction . . . 7

2.2 Langevin Mixture Model Parameter Estimation . . . 11

2.2.1 Mixture of Langevin Selection Using MML . . . 13

2.3 Support Vector Machines Kernels Generation . . . 16

2.3.1 Fisher Kernels . . . 18

2.3.2 Probability Product Kernels . . . 19

2.3.3 Kernels Based on Information Divergence . . . 20

2.4 Experimental Results . . . 22

2.4.1 Synthetic Dataset . . . 23

2.4.2 Email Categorization . . . 26

2.4.3 Spam email Filtering . . . 28

3 Simultaneous Online Spherical Data Clustering and Feature Selection 41
3.1 Introduction . . . 41
3.2 Mixture Density and Feature Selection . . . 45
3.3 Unsupervised Learning of the Model . . . 46
3.3.1 Model Selection . . . 46
3.3.2 Parameter Estimation . . . 49
3.4 Online Learning with Feature Selection . . . 51
3.5 Experimental Results . . . 54
3.5.1 Filtering Using Bag of Visual Words Approach . . . 56
3.5.2 Filtering Using Hybrid Generative/Discriminative Learning . . . 60
3.5.3 Online Web pages Clustering . . . 61

4 A Bayesian Clustering and Feature Selection for Spherical Data 69
4.1 Introduction . . . 69
4.2 Bayesian Estimation . . . 71
4.3 Mixture Density and Feature Selection . . . 75
4.3.1 Bayesian Learning using RJMCMC . . . 77
4.4 Experimental Results . . . 85
4.4.1 Synthetic Data . . . 85
4.4.2 Image Categorization . . . 86
4.4.3 Topic Detection and Tracking . . . 92

5 On Nonparametric Bayesian Spherical Data Clustering and Feature Selection 99
5.1 Introduction . . . 99
5.2 Infinite Langevin Mixture Model . . . 100
5.2.1 Priors and Posteriors . . . 102
5.2.2 The Complete Algorithm . . . 104
5.3 Infinite Langevin Clustering and Feature Selection . . . 105
5.3.1 Bayesian Hierarchial Model . . . 106
5.3.2 Nonparametric Bayesian Learning . . . 107
5.4 Experimental Results . . . 111
5.4.1 Text Categorization . . . 112
5.4.2 Topic Detection and Tracking . . . 113

6 Conclusion 115

A Proof of Equation 22 118

B Proof of Equation 28 119

C Proof of Equation 34 120

D Proof of Equation 36 121

E Proof of equation 15 122

F Proof of equation 16 124

G Proof of equation 17 126

H Proof of equation 20 128

List of References 129

List of Tables

2.1 True and estimated parameters for synthetic data using one-dimensional Langvin mixture. N denotes the total number of elements, N_j denotes the number of elements in cluster j, μ_j, κ_j and p_j are the true parameters. $\hat{\mu}_j$, $\hat{\kappa}_j$ and $\hat{p}_j$ are estimated parameters. 24

2.2 True and estimated parameters for synthetic data generated from four and five components LMM. N denotes the total number of elements, N_j denotes the number of elements in cluster j, μ_j, κ_j and p_j are the true parameters. $\hat{\mu}_j$, $\hat{\kappa}_j$ and $\hat{p}_j$ are estimated parameters. 25

2.3 True and estimated parameters for four-dimensional synthetic data generated from six-components LMM. N denotes the total number of elements, N_j denotes the number of elements in cluster j, μ_j, κ_j and p_j are the true parameters. $\hat{\mu}_j$, $\hat{\kappa}_j$ and $\hat{p}_j$ are estimated parameters. 26

2.4 Enron Dataset Statistics. 27

2.5 Classification performance (average±variance) obtained for Enron using LMM, GMM, SVM-p, SVM-RBF and SVM-s. 30

2.6 Global Contingency Table for Spam Classifier. 34

2.7 Average rounded confusion matrices for Text-based 2.7(a)2.7(b), Image-based 2.7(c)2.7(d), “Text+ Image”-based 2.7(e)2.7(f) spam classifiers by LMM (left column) and GMM (right column). 35

2.8 Results (in%) on terc05-p1 using different tested kernels for LMM and GMM. All results reported in the table are for “Image +Text” as it has shown to provide best performance. 36

2.9 $WAcc$ (in%) when considering terc05-p1 using LMM and GMM. 39

2.10 Accuracy (in$\%$) of different approaches on Princeton and Dredze dataset 40

3.1 Performance For Spam Filtering based on movM and GMM with and without feature selection. 57

3.2 Performance For Spam Filtering based on movM and GMM with feature selection in online settings. 59
3.3 Performance For Spam Filtering based on hybrid framework for both models in different scenarios. 61
3.4 Performance For Yahoo20 dataset in off-line and online settings for both movM and GMM using Feature selection. 63

4.1 True and estimated parameters for synthetic data generated from LMM. N and j denote the total number of elements and component number in each dataset. μ_j, κ_j and p_j are the true parameters. $\hat{\mu}_j^B$, $\hat{\kappa}_j^B$ and $\hat{p}_j^B$ are estimated parameters using Bayesian approach. $\hat{\mu}_j^{EM}$, $\hat{\kappa}_j^{EM}$ and $\hat{p}_j^{EM}$ are estimated parameters using the EM algorithm. 86
4.2 Performance of TDT framework for TDT-2 dataset based on different models averaged over 5 runs 96
4.3 Performance of TDT framework for 20 Newsgroup dataset based on different models averaged over 5 runs. 97

5.1 Classification F1 results for the Yahoo20 and WebKB datasets. 112
5.2 Performance of TDT framework for TDT-2 dataset based on different models averaged over 5 runs 114
5.3 Performance of TDT framework for 20 Newsgroup dataset based on different models averaged over 5 runs. 114

List of Figures

2.1 Artificial Histograms. 23

2.2 Number of clusters found using MML (left) and MDL (right) criteria for the first dataset. 24

2.3 Number of clusters found using MML (left) and MDL (right) criteria for the second dataset. 24

2.4 Number of clusters found using MML (left) and MDL (right) criteria for 4-components LMM for 2-dimensional datasets. 25

2.5 Number of clusters found using MML (left) and MDL (right) criteria for 5-components LMM for 2-dimensional datasets. 26

2.6 Number of clusters found using MML (left) and MDL (right) criteria for six-components LMM for four-dimensioal dataset. 27

2.7 Number of clusters of Enron dataset found using MML when considering LMM and GMM. 29

2.8 Classification accuracy as a function of the number of training documents. 29

2.9 Spam Filter Architecture 31

2.10 ROC graphs for different kernels using LMM and GMM. 38

3.1 Classification accuracy for the spam data set as a function of (a) the vocabulary size, (b) the number of aspects. 57

3.2 Number of clusters determined to represent spam and legitimate emails. 58

3.3 Micro-averaged F1 and macro-averaged F1 vs number of categories on Yahoo20 dataset with and without feature selection based on movM and GMM in off-line framework. 63

3.4 Micro-averaged F1 and macro-averaged F1 in Online Framework with and without feature selection based on movM and GMM. 64

3.5 Number of clusters determined to represent Yahoo20 categories based on online movM with feature selection (movMFS). 66

3.6 Number of clusters determined to represent Yahoo20 categories based on online GMM with feature selection (GMMFS). 68

4.1 Proposed Bayesian framework. 76
4.2 Time series plot of Gibbs-within-Metropolis for the first dataset. 88
4.3 Number of clusters found for the different generated datasets using both the Bayesian (a,c,e,g) and EM (b,d,f,h) approaches. 90
4.4 Sample images from each category in the VOC data set. 91
4.5 Classification accuracy for the PASCAL VOC2005 dataset as a function of (a) the number of aspects, (b) the vocabulary size. 92
4.6 ROC curves for the categorization of four object categories in PASCAL VOC2005. 93
4.7 Analysis of datasets. 96
4.8 F1(micro-averaged) vs number of topics in TDT framework for both Bayesian and EM approaches based on Langevin mixture model (LMM)& Gaussian mixture model (GMM). 97

5.1 Graphical Model representation of the Bayesian hierarchical Langevin mixture model. Nodes in this graph represent random variables, rounded boxes are fixed hyperparameters, boxes indicate repetition (with the number of repetitions in the upper left) and arcs describe conditional dependencies between variables. 104

List of Symbols

MLE: Maximum Likelihood Estimation

EM: Expectation Maximization

ML: Machine Learning

SVMs: Support Vector Machines

MML: Minimum Message Length

BoW: Bag of Words

BoVW: Bag of Visual Words

GMM: Gaussian Mixture Model

LMM: Langevin Mixture Model

MCMC: Markov Chain Monte Carlo

M-H: Metropolis Hasting

RSEM: Recursive EM

TDT: Topic Detection and Tracking

AIC : Akaike's Information Criterion

BIC: Bayes Information Criterion

MDL: Minimum Description Length

OCR: Optical Character Recognition

vM: Von Mises distribution

vMF: Von Mises Fisher distribution

movM: mixture of Von Mises distribution

MI: Mutual Information

RJMCMC: Reversible jump Markov chain Monte Carlo

ARS: Adaptive rejection sampling

CHAPTER 1

Introduction

As various disciplines have witnessed integration of digital technologies, high-dimensional sparse data are becoming more prevalent in every field of human endeavor. In the particular case of machine learning, such problems have been tackled using statistical learning, providing a rich and flexible techniques that can be applied to model data randomness and uncertainty. In this context, often one tries to understand this mass of data through analyzing informative patterns and describing the best possible model which succeeds in capturing the regularities in the data generating process. Nonetheless selecting an appropriate model that solves all aspects of application at hand is a major challenge as different approaches are needed, for distinct aspects, and often depend on different representational choices. For instance, although modeling based on Gaussian mixtures has provided good performance in some applications, recent works have shown that Gaussian model is sensitive to noise and irresistible to outliers when dealing with high-dimensional data. Indeed, among the challenges when using finite mixture modeling, there is the choice of appropriate parametric form of the probability density functions to represent the components.

Compared to the Gaussian, Langevin distribution has been shown to be a good alternative [1–3]. Usually, it is adopted to model problems involving high-dimensional spherical (L_2-normalized) vectors [1]. Indeed, it implicitly uses cosine similarity that is easy to interpret and simple to compute for sparse vectors, and has been widely used in text mining [4], spam filtering [2], gene expression analysis [5], and topic detection [6–8]. Works about directional data in general and spherical ones in particular have been developed thanks to the efforts of Watson, Stephens and others [9–19]. Thus, in this thesis we shall consider Langevin model to build our statistical frameworks.

1.1 Background

In ML applications, data points are usually represented as vectors of features. During the preprocessing step, normalization is generally applied to resolve some domain-related problems (e.g. long document domination in case of text classification). Different normalization approaches have been extensively used in the past such as L_1 [20] and L_2 [21] (known as Euclidean norm). In this thesis, we shall concentrate on L_2 normalization which has been shown repeatedly to improve the performance of classification [22]. More importantly, once the feature vectors are L_2 normalized they can be visualized as points on the circumference of a circle (two dimensions) or on the surface of a sphere (three dimensions) or generally on a hypersphere (for a higher number of dimensions). Spherical data are also common in astronomy, biology, geology, medicine and meteorology [1].

1.1.1 Langevin Distribution

Let $\vec{X} = (X_1, \ldots, X_D)$ be a random unit vector in $\mathbb{R}^D$. $\vec{X}$ has D-variate Langevin distribution if its probability density function is given by [24]:

$$p_D(\vec{X}|\vec{\mu}, \kappa) = \exp\{\kappa \vec{\mu}^T \vec{X} - a_D(\kappa)\} \tag{1}$$

on the $(D-1)$-dimensional unit sphere $\mathbb{S}^{D-1} = \{\vec{X}|\vec{X} \in \mathbb{R}^D : ||\vec{X}|| = \sqrt{\vec{X}\vec{X}^T} = 1\}$, with mean direction unit vector $\vec{\mu} \in \mathbb{S}^{D-1}$, where $\vec{\mu}^T$ denotes the transpose of $\vec{\mu}$ and non-negative real

[1] More details and thorough discussions about the statistics of spherical data in particular and directional data in general can be found in [23].

concentration parameter $\kappa \geq 0$. The normalizing constant function $a_D(\kappa)$ is given by:

$$a_D(\kappa) = -\log\Big\{\frac{\kappa^{\frac{D}{2}-1}}{(2\pi)^{\frac{D}{2}} I_{\frac{D}{2}-1}(\kappa)}\Big\} \tag{2}$$

where $I_D(\kappa)$ denotes the modified Bessel function of first kind [24]. From Eq.1 we can notice that Langevin distribution is a member of (curved)-exponential family of order D, whose shape is symmetric and unimodal, with minimal canonical parameter $\kappa\vec{\mu}$ and minimal canonical statistic $\vec{X}$. Moreover, its mean and covariance values are given by [25]:

$$E_{(\vec{\mu},\kappa)}\{\vec{X}\} = \acute{a}_D(\kappa)\vec{\mu} \tag{3}$$

$$\Sigma = Cov(\vec{X}_i, \vec{X}_j) = \acute{a}_D(\kappa)\vec{\mu}_i\vec{\mu}_j + \frac{a_D(\kappa)}{\kappa}(I_D - \vec{\mu}_i\vec{\mu}_j) \tag{4}$$

where $\acute{a}_D(\kappa)$ is given by:

$$\acute{a}_D(\kappa) = A_D(\kappa) = \frac{I_{\frac{D}{2}}(\kappa)}{I_{\frac{D}{2}-1}(\kappa)} \tag{5}$$

If $\kappa = 0$ the distribution is uniform, and for small κ, it is close to uniform. While if κ is large ($\kappa \to \infty$), the distribution becomes very concentrated about the angle $\vec{\mu}$ with κ being a measure of the concentration. In fact, as κ increases, the distribution approaches a normal distribution in $\vec{X}$ with mean $\vec{\mu}$ and variance $1/\kappa$.

For the 2-dimensional ($D = 2$) and 3-dimensional ($D = 3$) cases we find vM and vMF distributions, respectively. vM distribution [1] [26] is a probability distribution in which the data are concentrated on the circumference of a unit circle. Using Eq.1 the vM probability density function can be written as follows

$$p_2(\vec{X}|\vec{\mu},\kappa) = \frac{1}{2\pi I_0(\kappa)} \exp\{\kappa\vec{\mu}^T\vec{X}\} \tag{6}$$

where I_0 is the modified Bessel function of the first kind and order zero [24]. When directions in free space are of interest, a circular distribution will no longer be sufficient. It is necessary to

[1] Also known as the circular normal distribution [26].

consider a distribution on the surface of a sphere. Hence, vMF [27] can be adopted as a generative model for modeling spherical data in spherical unit and has been used successfully in text, speech and video clustering applications [1, 28, 29] [29]. For vMF, Eq.1 is reduced to

$$p_3(\vec{X}|\vec{\mu},\kappa) = \frac{\kappa}{4\pi\sinh(\kappa)}\exp(\kappa\vec{\mu}^T\vec{X}) \tag{7}$$

It is noteworthy that maximum likelihood estimations for von Mises and von Mises Fisher distributions have been proposed in [30] and [31], respectively.

1.1.2 Finite Langevin Mixture Model

Let $p(\vec{X}_i|\Theta)$ be a mixture of M Langevin distributions (i.e. a linear combination of some M component distributions). The probability density function $p(\vec{X}_i|\Theta)$ is then given by

$$p(\vec{X}_i|\Theta) = \sum_{j=1}^{M} p_D(\vec{X}_i|\theta_j)p_j \tag{8}$$

where $\Theta = \{\vec{P} = (p_1,\ldots,p_M), \vec{\theta} = (\theta_1,\ldots,\theta_M)\}$ denotes all the parameters of the mixture model such as $\theta_j = (\mu_j,\kappa_j)$ and $\vec{P}$ represents the vector of clusters probabilities (i.e. mixing weights) such that $p_j \geq 0$ and $\sum_{j=1}^{M} p_j = 1$.

Learning finite mixture models has received lots of attention over the years and can be broadly grouped into deterministic methods and Bayesian methods [32]. In one hand, deterministic methods aim at optimizing the model likelihood function which are generally implemented within the expectation-maximization (EM) framework. The EM algorithm is known to converge to local solutions and is highly dependent on initialization. However, convergence to a globally optimal solution is not guaranteed. On the other hand, there is a growing interest in Bayesian methods which are considered as an alternative way to deal with mixture models in order to overcome the problems generally faced when using deterministic approaches such as EM. Indeed, Bayesian methods have been widely and successfully applied in different domains by allowing powerful frameworks that combine information brought by the data (likelihood function) with expert opinion (prior information) to produce an updated expert opinion (posterior information).

1.2 Contributions

Our main contributions can be summarized as:

- We propose [2, 3] hybrid generative/ discriminative models as we are motivated by the capability of injecting priori information and inferring hidden pattern of objects when using Langevin mixture model, and the speed and good generalization when adopting SVMs. Indeed, instead of using Langevin mixture directly for classification, we build probabilistic kernels based on Langevin mixture and information divergence.
- We propose simultaneous clustering and feature selection frameworks in offline [33] and online sttings [34] devoted to the applications in which spherical data representations are involved [8].
- We propose [35] a Bayesian algorithm based on finite Langevin mixture. Moreover, we extend this work to allow simultaneous clustering and feature selection using Bayesian inference [36].
- We propose a nonparametric Bayesian infinite mixture for spherical patterns. We also consider the problem of feature selection within the same framework by proposing a unified efficient framework [37].

1.3 Thesis Overview

The organization of this thesis is as follows:

- In Chapter 2, we propose a unified probabilistic framework, where we build probabilistic kernels from mixture of Langevin distributions for Support Vector Machine.
- In Chapter 3, a probabilistic framework that allows simultaneous clustering and feature

selection in offline and online settings using finite mixtures of Von Mises distributions (movM) (which are special cases of Langevin distributions) are presented.

- In Chapter 4, we introduce a Bayesian approach for Langevin mixture model estimation and selection. Moreover, we further enhance our model by considering the problem of feature selection, and hence, we propose a framework that allows simultaneous clustering and feature selection in Bayesian settings using movM.
- In Chapter 5, motivated by the need to adapt to dynamic settings, we devote this chapter to develop a clustering framework based on nonparametric Bayesian approach for spherical patterns. Furthermore, we extend our model to simultaneously handle feature selection in infinite settings.
- Finally, Chapter 6 provides concluding remarks and future work directions.

CHAPTER 2

Hybrids of Generative Discriminative Models for Spherical Data

In this chapter we develop an MML criterion for the Langevin mixture model. Moreover, we derive different SVM probabilistic kernels based on information divergence and Fisher score from mixture of Langevin distributions. Finally, we present experimental results of applying our approach on synthetic data, email spam classification and email categorization.

2.1 Introduction

Nowadays, machine learning (ML) and data mining researchers have made substantial breakthroughs in every field of human endeavor and newer applications are adequately emerging. For example, automated categorization of digital multimedia into predefined categories is a major revolution taking place over traditional categorization, due to its effectiveness, savings of labor power and its portability to different domains [38, 39]. Such modern applications have been driving the research in ML to construct flexible statistical representations for different kind of data (i.e. continuous, discrete, binary, etc) which we confront in these particular kind of applications. Broadly speaking, the approaches adopted when using ML techniques can be grouped into: generative models and discriminative models.

In supervised classification problems, discriminative models define the boundaries between different classes (i.e. categories) by maximizing the margin between data of interest in the training phase. Then, in classification phase, the label that will be assigned for new unseen data depends on which side of the boundary the mapped input data point was set according to the constructed model. To illustrate, let $(\mathcal{X}, C) = \{\vec{X}_i, c_j\}$, where $1 < i \leq N$ and $j = 1, \ldots, M$ be a training dataset composed of N labeled data associated with M labels. A discriminative classifier estimates a classification function $C = f(\vec{X})$ directly from data. Therein, the data will be associated with one label defined by the sign of the function. Examples include SVMs [40], Neural network, Gaussian process, Logistic regression [41], etc. Discriminative models have received a great deal of attention in many areas such as bio-informatics, text classification, speech recognition and image classification, object categorization, etc. In contrast, the fundamental purpose of generative approaches is to present a probability density model p over input data points, hidden variables and output labels of tailored system. To achieve this goal, generative models learn the joint probability density function $p(\vec{X}_i, C)$ of input data point $\vec{X}_i$ and class C for each class and then classifying using Bayes rule $(p(C|\vec{X}) \propto (p(C)p(\vec{X}|C)))$ to estimate $p(C|\vec{X})$ and choose the most probable label [42]. Examples include, Hidden Markov Model, Bayesian Network, mixture models, etc.

Diversified applications, sophisticated mechanisms, unique features and immense volume of unlabeled data in unrelated fields has made it difficult and very expensive to prepare potential labeled training examples to design probabilistic models in the scientific and industrial areas. To handle this particular obstacle and unlike discriminative approaches, generative approaches have shown both analytically and experimentally a better performance with small number of training and unlabeled examples and they present a principled framework for handling uncertainty and missing data [43]. Furthermore, in generative learning phase each class is learned individually and we can explicitly define the correspondence between the variables in the underlying model equally. Hence, unlike discriminative models, in which all classes are needed to be considered simultaneously, we can easily retrain old classes or even learn new classes with updating the previous model of learned classes rather than rebuild our model from scratch. Yet, discriminative approaches don't regard the details of the given model for different classes and don't engage prior knowledge of application environment and expert impression, they perform more like a *black box* where they return the class of the input data point without presenting a clear principled reason and way. This is particularity crucial as it makes discriminative models faster than generative models by focusing on constructing the boundaries between different classes rather than wasting the time on finding the structure of the data and trying to figure out the density from which the data were drawn. Another characteristic of discriminative models is the fact that they have lower asymptotic error than generative models [43].

Problem Statement

The blending of generative and discriminative approaches has prevailed by exploring and adopting distinct characteristics of each approach toward constructing a complementary system combining the best of both [44–46]. There are several approaches that have been proposed based on generic properties of the generative and discriminative models which in turn don't provide realistic modeling for specific domains problems as in ML applications, for instance. In the context of ML, evolved learning approaches target applications that vary in their complexity, emerged data representations, modeling capabilities and generalization power. In particular applications, analyzing the data of interest is a vital component in learning procedure such as in multimedia classification. A standard representation method of multimedia examples (e.g. videos, images, text, etc.) is vector space representation which has witnessed a massive shift from well-known extensively used vectorial representation in which each document (or image) is described by a single vector to bags of vectors representation in which each document (or image) is described by a set of vectors.

In the same vein, the adoption of L_2 normalization has been shown to play an important role, as a preprocessing step, in many practical applications especially those generating count data. Indeed,

L_2-normalization[1] has been used in an attempt to overcome sparsity problem and improve clustering performance especially in the domain of text, image and video classification [22]. Consider, for example, text classification, it has been noticed that L_2 normalization alleviates the impact of dominating long documents on the classification decision [22, 48]. Also, L_2 normalization has been shown to increase the robustness to various changes, such as illumination changes in image classification [47, 49]. It is noteworthy that once the feature vectors are L_2 normalized they can be visualized as points on a hypersphere, which can be naturally modeled using spherical distributions. In one hand, normalized set (or bag) of vectors representation has challenged the capability of using discriminative approaches (e.g. SVM) to classify this particular data. On other hand, it has raised the awareness of generative models (e.g. mixtures of distributions) capability to model such rich presentation of the data at hand. In this chapter, our main contributions are:

- We tackle the problem of automatic determination of the number of components (i.e. model selection) of Langevin mixture model by proposing MML criterion [50] for model selection. Previously, the authors in [51] and [52] proposed MML for a low dimensional data $D = 1$ (Von Mises mixture) and $D = 2$ (Von Mises Fisher mixture), respectively and are hence limited solutions to many complicated problems when the data has high dimensionality such as image classification. To this aim, in this chapter, we generalize these expressions to the multidimensional case $D > 2$.
- We develop several well-motivated probabilistic SVM kernels based on Langevin mixture models. In particular, we develop closed-form expressions of the Kullback-Leibler kernel, Rényi kernel, Jensen-Shannon kernel, probability product kernel and Fisher kernel between two Langevin distributions.
- We discuss the properties of proposed framework on abundant (hundreds of thousands), high-dimensional, directional and challenging data: Enron dataset (used for email categorization), trec05-p1 and Princeton dataset (used for spam filtering).

[1] In particular, the authors in [47] recommended strongly the normalization of data in feature space when considering SVM and have shown that normalization leads to considerably superior generalization performance.

- In case of spam filtering, we argue that existing filtering techniques based on either the body text of emails or attached images are no longer effective. Accordingly, in this chapter we propose a general framework to construct an intelligent, adaptive and well-grounded spam filter that utilizes the information provided in different parts of the email and prevents potential defeatism of user privacy. To achieve this goal, we propose a hybrid statistical framework that combines and uses simultaneously both textual and visual email information to filter spam emails.
- We present detailed comparison of Langevin mixture model and the widely used Gaussian mixture model (GMM) in terms of the selection of the number of clusters and SVMs kernels generation performance based on the developed probabilistic kernels.

2.2 Langevin Mixture Model Parameter Estimation

In this section, we describe an EM learning algorithm to estimate the parameters of Langevin mixture model. Subsequently, we derive the equations for estimating the number of Langevin mixture components from data at hand using MML criterion. In order to estimate the parameters Θ of the underlying mixture we use common EM [32] framework which generates a sequence of models with non-decreasing log-likelihood on the data. Following EM, consider the complete data to be $\{\vec{X}_i, \vec{Z}_i\}$, where $\vec{Z}_i = \{Z_{i1}, \ldots, Z_{iM}\}$ denotes the missing vectors, such that $\sum_{j=1}^{M} Z_{ij} = 1$ with $Z_{ij} = 1$ if $\vec{X}_i$ belongs to class j and 0, otherwise. The E-step in the EM computes the posterior probabilities given by the following equation:

$$\hat{Z}_{ij}^{t} = \frac{p_j(\vec{X}_i|\theta_j)p_j}{\displaystyle\sum_{j=1}^{M} p_j(\vec{X}_i|\theta_j)p_j} \qquad (1)$$

where $\hat{Z}_{ij} \in [0,1], \sum_{j=1}^{M} \hat{Z}_{ij} = 1$ and denotes the degree of membership of $\vec{X}_i$ in the jth cluster. In the M-step, given the conditional expectation of complete log-likelihood $\mathcal{Q}(\Theta, \Theta^t) = \sum_{i=1}^{N}\sum_{j=1}^{M} \hat{Z}_{ij}^{t} \log(p(\vec{X}_i|\theta_j)p_j)$, we update the parameters estimate according to $\Theta^{t+1} = \arg\max_{\Theta} \mathcal{Q}(\Theta, \Theta^t)$.

A complete EM algorithm for Langevin mixture has been proposed in [1] and we shall use it in our work. The EM algorithm iterates and depends on initialization because the likelihood function often has numerous local maxima. Thus, good initialization is crucial for finding ML estimates. Many different initialization procedures have been suggested in the literature but no method uniformly outperforms the others. Among proposed approaches, we used spherical K-means [53] since it takes into account the spherical nature of our data. The complete parameter estimation steps for Langevin mixture are summarized in algorithm 1 (interested readers might see [1] [54] for further details). It is noteworthy that algorithm 1 assumes that the number of components is known which is in general not true. Indeed, apart from parameter estimation, another important problem is the selection of the appropriate number of components M. We discuss this issue in the next subsection.

Algorithm 1 Initialization and Complete Estimation Algorithm

```
INPUT: Set of $N$ $D$-dimensional data points $\mathcal{X}$ on $\mathrm{S}^{D-1}$
OUTPUT: Clusters of $\mathcal{X}$ over a mixture of $M$ Langevin distributions
Apply spherical K-means [53] on $N$ $D$-dimensional vectors to obtain the initial parameters for
    each component $\mu_j$, $\kappa_j$, $j = 1, \dots, M$
E step: Compute $\hat{Z}_{ij}^t$ using Eq.1.
M step:
    $p_j^{t+1} = \frac{1}{N}\sum_{i=1}^{N}\hat{Z}_{ij}^t$    $\vec{\mu}_j^{t+1} = \sum_{i=1}^{N}\vec{X}_i\hat{Z}_{ij}^t$
    $\bar{R} = \frac{\|\vec{\mu}_j\|}{Np_j}$    $\vec{\mu}_j^{t+1} = \frac{\sum_{i=1}^{N}\vec{X}_i\hat{Z}_{ij}^t}{\|\sum_{i=1}^{N}\vec{X}_i\hat{Z}_{ij}^t\|}$
    $\kappa_j^{t+1} = \frac{\bar{R}p - \bar{R}^3}{1 - \bar{R}^2}$
```

2.2.1 Mixture of Langevin Selection Using MML

One of the central issues in mixture models is to determine the optimal degree of complexity (i.e. optimal number of clusters). A too limited mixture model (i.e. with few number of components) will not capture the structure of the data, while a too complex one (i.e. with many components) will cause overfitting of the data, and in both cases the generalization capability of the model will be poor. The model selection problem can be viewed then as one that helps finding the optimal trade-off between the complexity of the model and goodness of fit. Thus, many approaches have been proposed and can broadly be divided into deterministic and stochastic approaches. In this chapter, we consider deterministic approaches which are widely deployed in the field of pattern recognition and are less computationally demanding comparing to stochastic approaches. Deterministic approaches are based on the minimization of the negative log likelihood function $p(\mathcal{X}|\Theta)$ to which penalty function is added. Examples include AIC [55], BIC [56], MDL [57] and MML [50] [2]. As a well-established selection criterion, MML has been repeatedly shown to demonstrate good performance in case of Gaussian, Gamma, Poisson, Dirichlet and generalized Dirichlet mixtures and outperforms AIC and MDL approaches [59, 60]. Thus, we propose the consideration of MML criterion to find the optimal number of mixture components by minimizing the following objective function [50]:

$$\begin{aligned} MessLength(M) \simeq & -\log(h(\Theta)) - \log(p(\mathcal{X}|\Theta)) + \tfrac{1}{2}\log(|F(\Theta)|) \\ & + \tfrac{N_p}{2}(1 - \log(12)) \end{aligned} \tag{2}$$

where $h(\Theta)$ is the prior probability, $p(\mathcal{X}|\Theta)$ is the likelihood, $F(\Theta)$ is the expected Fisher information matrix which is generally approximated by complete-data Fisher information matrix in the case of finite mixture models [59], $|F(\Theta)|$ is its determinant, and N_p is the number of free parameters to be estimated which is equal to $M(D+1)-1$ in our case. In the following, we calculate

[2] Other approaches are possible also. For instance, in [58], the authors have used mixture of von Mises distributions learned using maximum likelihood for parameters estimation and bootstrap likelihood ratio approach to assess the optimal number of components and applied to study the problem of sudden infant death.

the Fisher information and we propose a prior to obtain the complete message length expression for a finite Langevin mixture.

Fisher Information ($F(\Theta)$)

Fisher information matrix is the expected value of Hessian matrix of the logarithm of minus the likelihood of the mixture $F(\Theta) \equiv -E[\frac{\partial^2}{\partial_{k_1}\partial_{k_2}} \log p(\mathcal{X}|\Theta)]$, for $k_1 = 1, \ldots, M \times D$ and $k_2 = 1, \ldots, M \times D$. Analytically, for mixture models it is difficult to obtain the Fisher information matrix. Instead, we replace $F(\Theta)$ by complete data Fisher information matrix $F(\Theta)_c \equiv -E[\frac{\partial^2}{\partial_{k_1}\partial_{k_2}} \log p(\mathcal{X}, \mathcal{Z}|\Theta)]$, which has a block diagonal structure and its determinant is given by:

$$|F(\Theta)| \simeq |F(\vec{P})| \prod_{j=1}^{M} |F(\vec{\mu}_j, \kappa_j)| \tag{3}$$

where $|F(\vec{P})|$ is the determinant of the information matrix of $\vec{P}$ and $|F(\vec{\mu}_j, \kappa_j)|$ is the Fisher information of the Langevin distribution representing component j. For $|F(\vec{P})|$, we can easily show that:

$$|F(\vec{P})| = \frac{N^{M-1}}{\prod_{j=1}^{M} p(j)} \tag{4}$$

where N is the number of data vector. As for $|F(\vec{\mu}_j, \kappa_j)|$ we can show that [61]:

$$|F(\vec{\mu}_j, \kappa_j)| = n_j^D u^2(\kappa_j) v^2(\vec{\mu}_{j,0}) \tag{5}$$

where n_j is the number of vectors assigned to cluster j, and

$$u(\kappa_j) = \kappa_j^{\frac{1}{2}(D-2)} A_D(\kappa_j)^{\frac{1}{2}(D-1)} \left(\kappa_j - A_D(\kappa_j) - \kappa_j A_D(\kappa_j)^2 \right)^{\frac{1}{2}} \tag{6}$$

$$v(\vec{\mu}_{j,0}) = \prod_{d=1}^{D-1} \sin^{D-2} \mu_{j,0,d-1} \tag{7}$$

where $\vec{\mu}_{j,0} = (\mu_{j,0,1}, \ldots, \mu_{j,0,D})$ denotes the spherical polar coordinates of $\vec{\mu}_j$. Substituting Eq.4 and Eq.5 into Eq.3, the Fisher information for a mixture of Langevin distribution can be written as:

$$|F(\Theta)| \simeq \frac{N^{M-1}}{\prod_{j=1}^{M} p(j)} \prod_{j=1}^{M} n_j^D u^2(\kappa_j) v^2(\vec{\mu}_{j,0}) \tag{8}$$

Prior distribution $h(\Theta)$

In the absence of any other knowledge about $\vec{P}$, $\vec{\mu}_j$ and κ_j, we suppose that they are mutually independent, which yields to the following prior distribution over the parameters:

$$h(\Theta) = h(\vec{P}) \prod_{j=1}^{M} h(\kappa_j) h(\vec{\mu}_j) \tag{9}$$

For the mixing probabilities $\vec{P}$, a common choice as a prior is the Dirichlet distribution as a prior:

$$h(\vec{P}) = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} p_j^{\eta_j - 1} \tag{10}$$

where $\eta = (\eta_1, \ldots, \eta_M)$ is the parameter vector of the Drichlet distribution. The choice of $\eta_1 = 1, \ldots, \eta_M = 1$ gives uniform prior over the space where $p_1 + \ldots + p_M = 1$. This prior is formulated by:

$$h(\vec{P}) = (M-1)! \tag{11}$$

Note that this uniform prior is defined over the $(M-1)$-dimensional region of hypervolume $1/(M-1)!$. For the parameter $\vec{\mu}_j$, we consider a uniform prior on the surface of the unit $(D-1)$-sphere:

$$h(\vec{\mu}_j) = \frac{1}{S_D} \tag{12}$$

where S_D is the surface of a unit $(D-1)$-sphere which is given by:

$$S_D = \begin{cases} D\frac{\pi^{\frac{D}{2}}}{\frac{D}{2}!}, & \text{if } D \text{ is even} \\ D\frac{2^{\frac{D+1}{2}}\pi^{\frac{D-1}{2}}}{D!!}, & \text{if } D \text{ is odd} \end{cases} \tag{13}$$

where “!!” denotes the double factorial. Furthermore, we consider the following prior, which has been found appropriate according to our experimental results, for the concentration parameter of a Langevin distribution [52]:

$$h(\kappa_j) = \frac{\kappa_j^{D-1}}{(1+\kappa_j^2)^{\frac{D+1}{2}}} \tag{14}$$

Substituting Eq.11, Eq.12 and Eq.14 into Eq.9, we obtain

$$h(\Theta) = \frac{(M-1)!}{S_D^M} \prod_{j=1}^{M} \left[\frac{\kappa_j^{D-1}}{(1+\kappa_j^2)^{\frac{D+1}{2}}} \right] \tag{15}$$

Thus, by substituting $\log |F(\Theta)|$ and $\log(h(\Theta))$ we can obtain the message length of a finite mixture of Langevin distribution and the complete algorithm to learn the Langevin mixture can be summarized as follows:

Algorithm 2 Complete Learning Algorithm

```
1: Estimate the parameters of the Langevin mixture distribution using the estimation algorithm 1
2: Calculate the associated message length $MessLength(M)$ using Eq. 2
3: Select the optimal model $M^*$ such that $M^* = \arg\min MessLength(M)$
```

2.3 Support Vector Machines Kernels Generation

SVMs [40] are known to give accurate discrimination in high-dimensional feature spaces and have received a great deal of attention in categorization and classification applications. Briefly, SVMs have outperformed other learning algorithms with good generalization, global solution, number of tuning parameters and their solid theoretical background. The core concept of SVMs is to discriminate classes with a hyperplane that maximizes the margin by solving quadratic programming (qp) problem with linear equality and inequality constraints. Let $\{(\vec{X}_1, C_1), \ldots, (\vec{X}_l, C_l)\}, \vec{X}_i \in \mathbb{R}^N$ be a training set of random independent identically distributed vectors belonging to two separate classes $C_i \in \{-1, 1\}$. SVMs solution for binary classification is given by [40]

$$f(\vec{X}) = \text{sign}(\sum_{i=1}^{l} C_i \alpha_i^* (\vec{X}.\vec{X}_i) + b^*) \tag{16}$$

where α^* is [1]a Lagrangian parameter. SVM evolves non-linear mapping of learned data from input space $\mathcal{X}$ into higher dimensional feature space $\mathcal{Z}$ where the classification performance is increased. This has been developed by applying several kernels $\mathcal{K}(\vec{X}, \vec{X}_i)$ that measure the similarities between vectors. Note that the inner product in Eq. 16 in the non-linear mapping is replaced by $\Phi(\vec{X}).\Phi(\vec{X}_i)$, which is simply the kernel function in input space $\mathcal{X}$. For background reading on SVMs and kernel methods, the reader is referred to [40].

It has been proved that the choice of the kernel is crucial to to provide reliable results and good generalization. In particular, the adoption of classic kernels (i.e. polynomial, Gaussian RBF) over the years has been challenged lately by the need to build problem oriented classifier that regards the nature of the problem at hand by exploiting a prior knowledge about the problem through data representation. For instance, in multimedia classification, multimedia objects O are generally represented as sets of local descriptors[3] $\mathcal{X} = (\vec{X}_1, \dots, \vec{X}_N)$, rather than one high-dimensional vector. Thereby, examples are represented as set of features which may vary in cardinality and as a result classical kernels cannot be deployed in this situation. To address certain limitation of SVMs, based on the complementary attributes of generative approaches, we propose a hybrid framework that models these descriptors, in an unsupervised way, using finite Langevin mixtures model from which probabilistic kernels are generated for SVMs.

Let $\mathcal{X} = (\vec{X}_1, \dots, \vec{X}_N)$ and $\acute{\mathcal{X}} = (\vec{\acute{X}}_1, \dots, \vec{\acute{X}}_{\acute{N}})$ be two sequences of spherical feature vectors representing two multimedia objects O and $\acute{O}$, respectively, and modeled by two Langevin finite mixtures $p(\vec{X}|\Theta)$ and $q(\vec{X}|\acute{\Theta})$, respectively, defined on Ω space (Ω is the p-dimensional space of Langevin distribution). In the following subsections, we derive different kernels, from Langevin mixture, based on probabilistic distances and Fisher score to tackle the problem of spherical data

[1] Using the superscript * to denote the *optimal* values of the cost function.

[3] This localized data presentation alleviates many problems associated with representing data in complex applications (e.g. video categorization) such as data sparsity and curse of dimensionality.

sequences classification using SVM. It is noteworthy that these kernels will allow to take into account the generative process of the data and could then be also called generative kernels [62].

2.3.1 Fisher Kernels

Authors in [63] have shown that a generative model can be used in a discriminative context by extracting Fisher scores $U_{\mathcal{X}}(\Theta) = \nabla \log(p(\mathcal{X}|\Theta))$ from the generative model and converting them into a Gram Kernel usable by SVMs. Each component of $U_{\mathcal{X}}(\Theta)$ is the derivative of the log-likelihood of the sequence $\mathcal{X}$ with respect to particular parameter. In the following, we shall show the derivations of the Fisher kernel

$$\mathcal{K}(\mathcal{X}, \mathcal{X}') = U_{\mathcal{X}}^{tr}(\Theta) I^{-1}(\Theta) U_{\mathcal{X}'}(\Theta') \tag{17}$$

for M-Langevin mixture models, where $I(\Theta)$ is the Fisher information matrix that has less significant role as was shown in [63] and can be approximated using the identity matrix. Through the computation of gradient of the log probability with respect to our model parameters: p_j, κ_j and $\vec{\mu}_j$, where $j = 1, \dots, M$, we obtain

$$\frac{\partial \log p(\mathcal{X}|\Theta)}{\partial \kappa_j} = \sum_{i=1}^{N} \hat{Z}_{ij} \Big[\vec{\mu}_j^T \vec{X}_i - a'_D(\kappa_j) \Big] \tag{18}$$

$$\frac{\partial \log p(\mathcal{X}|\Theta)}{\partial \vec{\mu}_j} = \frac{\sum_{i=1}^{N} \hat{Z}_{ij} \kappa_j \vec{X}_i}{\| \sum_{i=1}^{N} \hat{Z}_{ij} \kappa_j \vec{X}_i \|} \tag{19}$$

$$\frac{\partial \log p(\mathcal{X}|\Theta)}{\partial p_j} = \sum_{i=1}^{N} \Big[\frac{\hat{Z}_{ij}}{p_j} - \frac{\hat{Z}_{i1}}{p_1} \Big] \quad j = 2, \dots, M \tag{20}$$

where $\hat{Z}_{ij} = \frac{p(\vec{X}_i|\Theta)p_j}{\sum_{j=1}^{M} p(\vec{X}_i|\Theta)p_j}$ represents the probability that a vector $\vec{X}_i$ will be assigned to cluster j. It is noteworthy that in Eq.32, we take into account the fact that the sum of the mixing parameters equals one and thus there are only $M - 1$ free mixing parameters.

2.3.2 Probability Product Kernels

Another approach is developing kernels between probabilistic distributions $\mathcal{K}$:$\mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}$ that injects the domain-knowledge and invariance of generative models to SVMs [64]. PPK [42], for instance, maps data points in the input space to distributions over the sample space and a general inner product is then evaluated as the integral of the product of pairs of distributions and defined as

$$\mathcal{K}(p(\vec{X}|\Theta), q(\vec{X}|\acute{\Theta})) = \int_{\Omega} p(\vec{X}|\Theta)^{\rho} q(\vec{X}|\acute{\Theta})^{\rho} d\vec{X} \tag{21}$$

where ρ is a positive parameter. In the case of Langevin distribution, we can find a closed-form expression for the PPK and is given by (See Appendix A)

$$\int_{\Omega} p(\vec{X}|\theta)^{\rho} q(\vec{X}|\acute{\theta})^{\rho} d\vec{X} = \Big[\Big(\frac{\kappa\acute{\kappa}}{4}\Big)^{\frac{D}{2}-1} \frac{1}{(2\pi)^{D} I_{\frac{D}{2}-1}(\kappa) I_{\frac{D}{2}-1}(\acute{\kappa})}\Big]^{\rho} \Big[\frac{(2\pi)^{\frac{D}{2}} I_{\frac{D}{2}-1}(\xi_{\kappa,\acute{\kappa}}\rho)}{(\xi_{\kappa,\acute{\kappa}}\rho)^{\frac{D}{2}-1}}\Big] \tag{22}$$

A special case of PPK is when $\rho = 1$, which is called Expected Likelihood Kernel (ELK). Using Eq. (22) ELK for a Langevin distribution has the following form:

$$\int_{\Omega} p(\vec{X}|\theta) q(\vec{X}|\acute{\theta}) d\vec{X} = \Big(\frac{\kappa\acute{\kappa}}{4}\Big)^{\frac{D}{2}-1} \frac{I_{\frac{D}{2}-1}(\xi_{\kappa,\acute{\kappa}})}{(2\pi)^{\frac{D}{2}} I_{\frac{D}{2}-1}(\kappa) I_{\frac{D}{2}-1}(\acute{\kappa}) (\xi_{\kappa,\acute{\kappa}})^{\frac{D}{2}-1}} \tag{23}$$

When $\rho = \frac{1}{2}$ PPK has the form of Bhattacharyya kernel (BK) based on Bhattacharyya's measure of affinity between distributions. For Langevin distribution, BK is given by:

$$\int_{\Omega} p(\vec{X}|\theta)^{\frac{1}{2}} q(\vec{X}|\acute{\theta})^{\frac{1}{2}} d\vec{X} = \sqrt{\Big(\frac{\kappa\acute{\kappa}}{4}\Big)^{\frac{D}{2}-1} \frac{1}{(2\pi)^{D} I_{\frac{D}{2}-1}(\kappa) I_{\frac{D}{2}-1}(\acute{\kappa})}} \Big[\frac{(2\pi)^{\frac{D}{2}} I_{\frac{D}{2}-1}(\frac{\xi_{\kappa,\acute{\kappa}}}{2})}{(\frac{\xi_{\kappa,\acute{\kappa}}}{2})^{\frac{D}{2}-1}}\Big] \tag{24}$$

In the absence of closed forum for mixture models, we can approximate PPK using Monte Carlo simulations [64]

$$K(p(\vec{X}|\Theta), q(\vec{X}|\acute{\Theta})) \approx \frac{\beta}{N} \sum_{i=1}^{N} \frac{p^{\rho}(\vec{X}_i|\Theta)}{Z} p^{\rho}(\vec{X}_i|\Theta) + \frac{1-\beta}{\acute{N}} \sum_{\acute{i}=1}^{\acute{N}} \frac{q^{\rho}(\vec{X}_{\acute{i}}|\acute{\Theta})}{\acute{Z}} q^{\rho}(\vec{X}_{\acute{i}}|\acute{\Theta}) \tag{25}$$

Where $\beta \in [0,1]$ and $\vec{X}_1, \ldots, \vec{X}_N$ and $\vec{X}_1, \ldots, \vec{X}_{\acute{N}}$ are generated from $p(\vec{X}|\Theta)$ and $q(\vec{X}|\acute{\Theta})$, respectively. And Z and $\acute{Z}$ are normalizer for $p(\vec{X}|\Theta)$ and $q(\vec{X}|\acute{\Theta})$ after they are taken to the power of ρ, respectively.

2.3.3 Kernels Based on Information Divergence

An alternative method is to generate SVM kernels based on information divergence between distributions. In particular, it is a group of kernels obtained by exponentiating divergence measure between $p(\vec{X}|\Theta)$ and $q(\vec{X}|\acute{\Theta})$. For instance, the authors in [64, 65] have derived kernel distances between Gaussian mixtures and Liouville mixtures, respectively, using the following expression:

$$\mathcal{K}(p(\vec{X}|\Theta), q(\vec{X}|\acute{\Theta})) = e^{-aF(p(\vec{X}|\Theta), q(\vec{X}|\acute{\Theta}))} \tag{26}$$

where $a > 0$ is a kernel parameter included for numerical stability and $F(p(\vec{X}|\Theta), q(\vec{X}|\acute{\Theta}))$ can be any information divergence method as we shall explain in the following.

1. Kullback-Leibler Kernel (KL): is based on symmetric Kullback-Leibler divergence that measures the dissimilarity between two probability distributions $p(\vec{X}|\Theta)$ and $q(\vec{X}|\acute{\Theta})$ which is given by [66]:

$$F(p(\vec{X}|\Theta), q(\vec{X}|\acute{\Theta})) = \int_{\Omega} \Big[p(\vec{X}|\Theta) \log \frac{p(\vec{X}|\Theta)}{q(\vec{X}|\acute{\Theta})} + q(\vec{X}|\Theta) \log \frac{q(\vec{X}|\Theta)}{p(\vec{X}|\acute{\Theta})} d\vec{X} \Big] \tag{27}$$

   The fact that the Langevin distribution belongs to the exponential family of distributions allows us to find a closed-form expression for the KL divergence between two Langevin distributions (See Appendix B)

$$KL(p(\vec{X}|\theta), q(\vec{X}|\acute{\theta})) = -\log \frac{\kappa^{\frac{D}{2}-1}}{(2\pi)^{\frac{D}{2}} I_{\frac{D}{2}-1}(\kappa)} + \log \frac{\acute{\kappa}^{\frac{D}{2}-1}}{(2\pi)^{\frac{D}{2}} I_{\frac{D}{2}-1}(\acute{\kappa})} \tag{28}$$
$$+ [\kappa\vec{\mu} - \acute{\kappa}\vec{\acute{\mu}}]^T \acute{a}_D(\kappa)\vec{\mu}$$

   However, a closed form expression does not exist in the case of finite mixture models. Thus, we propose the use of different sampling approaches that have been proposed in [67] for Gaussian mixture models. Let $p(\vec{X}|\Theta) = \sum_{j=1}^{M} p_j p(\vec{X}|\theta_j)$ and $q(\vec{X}|\acute{\Theta}) = \sum_{\acute{j}=1}^{\acute{M}} \acute{p}_j q(\vec{X}|\acute{\theta}_j)$, the KL is given by finding the mapping π from one distribution to another:

$$KL_{MA}(p(\vec{X}|\Theta), q(\vec{X}|\acute{\Theta})) \approx \sum_{j=1}^{M} \Big(KL(p(\vec{X}|\theta_j), q(\vec{X}|\acute{\theta_j})) + \log \frac{p_j}{\acute{p}_{\pi(j)}} \Big) \tag{29}$$

where $\pi(j) = arg\min_{\acute{j}}\Big(KL(p(\vec{X}|\theta_j), q(\vec{X}|\acute{\theta_j})) - \log\acute{p_{\acute{j}}}\Big)$. Moreover, variational (KL_{VAR}) and upper bound (Kl_{UPP}) approximations have been proposed in the case of Gaussian mixtures and are given by [68]

$$KL_{VAR}(p(\vec{X}|\Theta), q(\vec{X}|\acute{\Theta})) \approx \sum_{j=1}^{M} p_j \log \frac{\sum_{j=1}^{M} p_j e^{-KL(p(\vec{X}|\theta_j), q(\vec{X}|\acute{\theta} j))}}{\sum_{\acute{j}=1}^{\acute{M}} \acute{p_{\acute{j}}} e^{-KL(p(\vec{X}|\theta_j), q(\vec{X}|\acute{\theta} j))}} \tag{30}$$

$$KL_{UPP}(p(\vec{X}|\Theta), q(\vec{X}|\acute{\Theta})) \approx \sum_{j=1}^{M} p_j \log \frac{p_j}{\acute{p} j} + \sum_{j=1}^{M} p_j KL(p(\vec{X}|\theta_j), q(\vec{X}|\acute{\theta_j})) \tag{31}$$

Another idea is to use Monte Carlo numerical approximation method [64]:

$$KL_{MC}(p(\vec{X}|\Theta), q(\vec{X}|\acute{\Theta})) \approx \frac{1}{L}\sum_{l=1}^{L} \log \frac{p(\vec{X}|\Theta)}{q(\vec{X}|\acute{\Theta})} \tag{32}$$

where $\vec{X}_1, \ldots, \vec{X}_L$ is a sample generated from $p(\vec{X}|\Theta)$.

2. Jensen-Shannon kernel (JS): is based on the Jensen-Shannon divergence which is given by [69]:

$$\begin{aligned} JS(p(\vec{X}|\Theta), q(\vec{X}|\acute{\Theta})) = H[\beta p(\vec{X}|\Theta) + (1-\beta) q(\vec{X}|\acute{\Theta})] - \beta H[p(\vec{X}|\Theta)] \\ - (1-\beta) H[q(\vec{X}|\acute{\Theta})] \end{aligned} \tag{33}$$

where β is a parameter and $H[p(\vec{X}|\Theta)] = -\int p(\vec{X}|\Theta)\log p(\vec{X}|\Theta) d\vec{X}$ is the Shannon entropy of $p(\vec{X}|\Theta)$. Thus, the Shannon entropy of Langevin distribution is given by (See Appendix C)

$$H[p(\vec{X}|\theta)] = \kappa a'_D(\kappa) + \Big(\frac{\kappa}{2}\Big)^{\frac{D}{2}-1} \frac{1}{(2\pi)^{\frac{D}{2}} I_{\frac{D}{2}-1}(\kappa)} \tag{34}$$

It is clear that when $\beta = \frac{1}{2}$, the JS is average the distance of Kullback-Leibler.

3. Rényi Kernel (RK) is another common kernel which is based on symmetric Rényi divergence of order σ [70]:

$$F(p(\vec{X}|\Theta), q(\vec{X}|\acute{\Theta})) = \frac{1}{\sigma - 1}\int \Big[p(\vec{X}|\Theta)^{\sigma} q(\vec{X}|\acute{\Theta})^{1-\sigma} + q(\vec{X}|\Theta)^{\sigma} p(\vec{X}|\acute{\Theta})^{1-\sigma} d\vec{X}\Big] \tag{35}$$

where σ controls the amount of smoothing for the distributions, $\sigma > 0$ and $\sigma \neq 1$. Thus, we find the Rényi divergence of order σ between two Langevin distributions (See Appendix D)

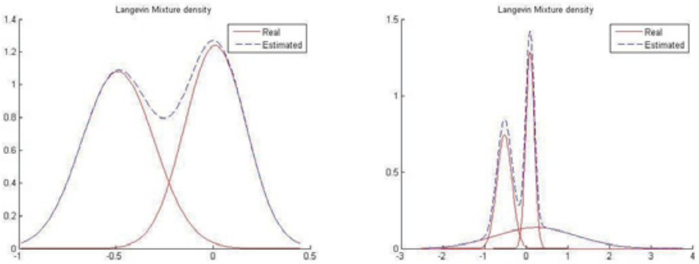
$$\int_\Omega p(\vec{X}|\theta)^\sigma q(\vec{X}|\acute{\theta})^{1-\sigma} d\vec{X} = \Big[\Big(\frac{\kappa}{2}\Big)^{\frac{D}{2}-1} \frac{1}{(2\pi)^{\frac{D}{2}} I_{\frac{D}{2}-1}(\kappa)}\Big]^\sigma \times \Big[\Big(\frac{\acute{\kappa}}{2}\Big)^{\frac{D}{2}-1} \frac{1}{(2\pi)^{\frac{D}{2}} I_{\frac{D}{2}-1}(\acute{\kappa})}\Big]^{1-\sigma} \Big[\frac{(2\pi)^{\frac{D}{2}} I_{\frac{D}{2}-1}(\zeta_{\kappa,\acute{\kappa}})}{(\zeta_{\kappa,\acute{\kappa}})^{\frac{D}{2}-1}}\Big] \tag{36}$$

Note that when $\sigma = \frac{1}{2}$ RK is reduced to BK. Because of the absence of closed form solutions for mixture models in case of JS and RK, we shall use in our experiments Monte Carlo simulation.

2.4 Experimental Results

Experiments were conducted to assess the performance of the proposed framework by comparing it to other techniques mentioned previously in the literature. To achieve this goal, we used synthetic data and two challenging problems which are: email categorization and spam email classification. The libsvm[4] software was used for SVMs classifier. We used stratified 10-fold cross validation to train and test each dataset, and averaged results were reported. We trained and tested our data sets with the five probabilistic kernels we proposed in section 2.3 which are: KL divergence kernel using the approximations in Eq.29 Eq.30 and Eq.31 that we call KL_{MA}, KL_{VAR}, KL_{UPP}, respectively, Rényi kernel (RK), Bhattacharyya kernel (BK), Expected likelihood kernel (ELK), Jensen-Shannon kernel (JS) and Fisher kernel (FK). For KL, RK, and JK parameters a and σ were selected from $\{2^{-10}, 2^{-9}, \ldots, 2^4\}$ and $\{0.1, 0.2, ..., 0.8\}$, respectively [64]. In the case where a closed-form expression does not exist for a given probabilistic kernel, we have used Monte Carlo approximation with 5000 generated points.

[4] http://www.csie.ntu.edu.tw/ cjlin/libsvm/



(a) Two-components (b) Three-components

Figure 2.1: Artificial Histograms.

2.4.1 Synthetic Dataset

In this section, we use synthetic datasets to validate the correctness of our finite Langevin mixture model (LMM) learning algorithm (Algorithm 2 in Section 3.3.1). Langevin distribution can be efficiently sampled using the simulation algorithm developed in [71], which was further improved in [72]. For the sake of verification, we generated artificial datasets from LMM, and we estimated the mixture parameters and tried to find the corresponding number of components. In our first experiment, we generated two one-dimensional datasets, where the first dataset represents LMM of two well separated components while the second one represents LMM of three overlapped components each of which has a total of 300 vectors grouped into two and three clusters, respectively. Figure 2.1 shows artificial histograms for those datasets. The real and estimated parameters for those datasets are given in Table 2.1. Figures 2.2 and 2.3 show the number of components found by our proposed algorithm and when adopting MDL criterion, which is given by [57]

$$MDL = -\log p(\mathcal{X}|\Theta) + \frac{N_p}{2}\log(N) \tag{37}$$

Clearly, our algorithm was able to find the exact number of clusters for each dataset. Accordingly, our algorithm was able to accurately estimate the mixture parameters and find the corresponding number of components for well-separated and overlapped data. Moreover, we tested our algorithm on multidimensional synthetic datasets. In particular, we generated two two-dimensional

Table 2.1: True and estimated parameters for synthetic data using one-dimensional Langvin mixture. N denotes the total number of elements, N_j denotes the number of elements in cluster j, μ_j, κ_j and p_j are the true parameters. $\hat{\mu}_j$, $\hat{\kappa}_j$ and $\hat{p}_j$ are estimated parameters.

	N_j	j	μ_j	$\hat{\mu}_j$	κ_j	$\hat{\kappa}_j$	p_j	$\hat{p}_j$
Dataset 1	100	1	-0.5	-0.49	10	11.50	0.48	0.50
(N =300)	200	2	0.1	0.09	10	11.50	0.52	0.50
Dataset 2	100	1	-0.40	-0.40	10	10.20	0.30	0.28
(N =300)	100	2	0.20	0.18	6.56	6.60	0.50	0.52
	100	3	0.60	0.60	2.10	2.44	0.20	0.20

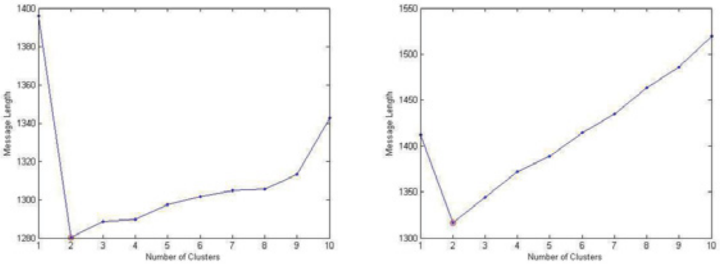


Figure 2.2: Number of clusters found using MML (left) and MDL (right) criteria for the first dataset.

datasets from four and five components LMM with different parameters as shown in Table 2.2. A total of 100 vectors for each of the densities were taken for the first dataset. For the second dataset a total of 100 vectors were taken for the first three densities and a total of 200 for the fourth and fifth

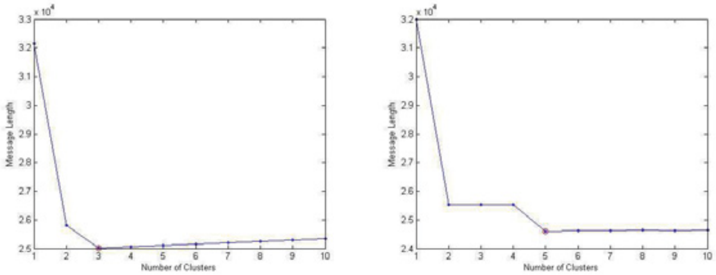


Figure 2.3: Number of clusters found using MML (left) and MDL (right) criteria for the second dataset.

Table 2.2: True and estimated parameters for synthetic data generated from four and five components LMM. N denotes the total number of elements, N_j denotes the number of elements in cluster j, μ_j, κ_j and p_j are the true parameters. $\hat{\mu}_j$, $\hat{\kappa}_j$ and $\hat{p}_j$ are estimated parameters.

	N_j	j	μ_j	$\hat{\mu}_j$	κ_j	$\hat{\kappa}_j$	p_j	$\hat{p}_j$
Dataset 3	100	1	(0.9547,0.2976)	(0.9565,0.2918)	100.20	99.78	0.35	0.35
(N =400)	100	2	(0.8570,0.5153)	(0.8614,0.5079)	40.56	39.04	0.20	0.19
	100	3	(-0.9993,0.0383)	(-0.9997,0.0264)	60.10	57.87	0.20	0.21
	100	4	(-0.8556,0.5176)	(-0.8667,0.4989)	64.89	63.45	0.25	0.25
Dataset 4	100	1	(-0.4885,0.8725)	(-0.4934,0.8698)	10	10.20	0.20	0.21
(N =700)	100	2	(0.2001,0.9798)	(0.2107,0.9776)	10	11.00	0.30	0.29
	100	3	(-0.0191,0.9998)	(-0.0134,0.9999)	10	9.082	0.20	0.19
	200	4	(-0.2269,0.9739)	(-0.2280,0.9737)	10	11.10	0.15	0.15
	200	5	(0.4360,0.8999)	(0.4444,0.8958)	10	10.45	0.15	0.16

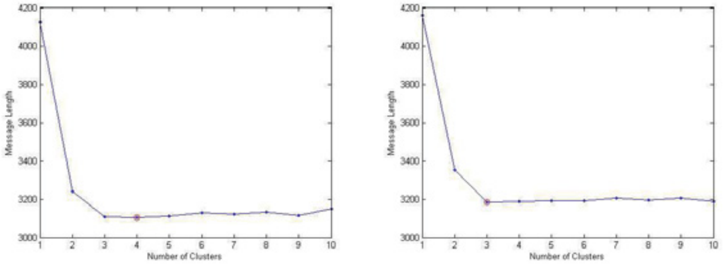


Figure 2.4: Number of clusters found using MML (left) and MDL (right) criteria for 4-components LMM for 2-dimensional datasets.

densities. Reported results were averaged over 10 runs. It is clear that the clustering performed by different mixtures is accurate when the dataset is small and well separated. Furthermore, Tables 2.1 and 2.2 show the true and estimated parameters for those two datasets. Figures 2.4 and 2.5 show the components found by MML and MDL for the generated datasets. Obtained results show that our algorithm was able to get the exact number of clusters and good approximation for mixture parameters. Finally, we verified our algorithm on four-dimensional dataset that was generated from six-components mixture model each of which densities has a total of 100 samples. Table 2.3 shows the real and estimated parameters found by our algorithm. Obviously, our algorithm was able to estimate accurately the parameters and the correct number of components (See Figure 2.6).

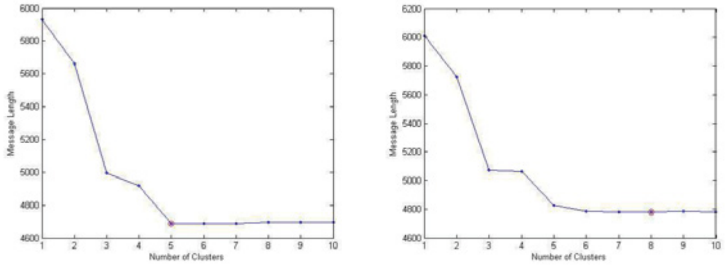


Figure 2.5: Number of clusters found using MML (left) and MDL (right) criteria for 5-components LMM for 2-dimensional datasets.

Table 2.3: True and estimated parameters for four-dimensional synthetic data generated from six–components LMM. N denotes the total number of elements, N_j denotes the number of elements in cluster j, μ_j, κ_j and p_j are the true parameters. $\hat{\mu}_j$, $\hat{\kappa}_j$ and $\hat{p}_j$ are estimated parameters.

	N_j	j	μ_j	$\hat{\mu}_j$	κ_j	$\hat{\kappa}_j$	p_j	$\hat{p}_j$
Dataset 5	100	1	(0.1997,0.0189,-0.3685,0.9077)	(0.2226,0.0191,-0.3561,0.9074)	10	11.00	0.14	0.16
(N =600)	100	2	(-0.1588,-0.3477,-0.1244,0.9156)	(-0.1629,-0.3412,-0.1340,0.9160)	10	10.81	0.16	0.14
	100	3	(0.1011,-0.0485,0.0471,0.9926)	(0.0985,-0.0465,0.0470,0.9929)	100.20	100.1	0.30	0.28
	100	4	(0.1242,0.3738,-0.0043,0.9191)	(0.1200,0.3753,-0.0019,0.9191)	40.56	39.85	0.15	0.16
	100	5	(-0.2645,0.1520,0.0227,0.9521)	(-0.2664,0.1522,0.0238,0.9515)	60.1	59.12	0.12	0.12
	100	6	(-0.0526,-0.1399,0.5361,0.8308)	(-0.0526,-0.1399,0.5361,0.8308)	64.89	63.36	0.13	0.14

2.4.2 Email Categorization

Email categorization is a rich and multifarious problem that poses several challenges. In particular, email folders may vary across different users and more importantly are richer than simple semantic topics since they may correspond to project groups, certain recipients, etc. Indeed, the manner that email users organize their files might change overtime, for instance, users may create new folders, while stop using already existing ones. The goal of this first application is twofold. First, validate the efficiency of the proposed learning algorithm of LMM (See algorithm 2). Second, compare its performance to another generative model and one discriminative model that were widely used in the past, namely, GMM and SVMs.

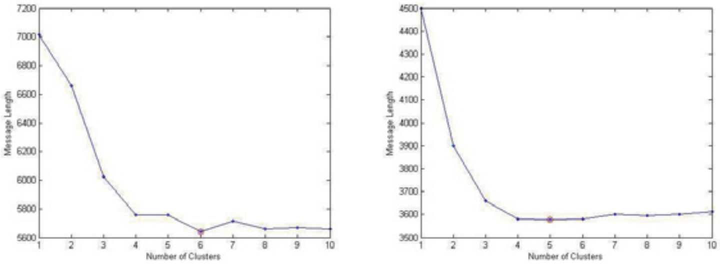


Figure 2.6: Number of clusters found using MML (left) and MDL (right) criteria for six-components LMM for four-dimensioal dataset.

Table 2.4: Enron Dataset Statistics.

	No. of Folders	No. of Emails	No. of Terms
beck-s	101	1971	97690
farmer-d	25	3672	315153
kaminski-v	41	4477	513460
kitchen-l	47	4015	60921
lokay-m	11	2489	15870
sanders-r	30	1188	56004
williams-w3	18	2769	81719

We conducted our experiments on a challenging dataset that has been widely considered in the past called Enron Email dataset [5] and is composed of 200,399 emails belonging to 158 users. We conducted experiments on the largest email directories that have been used in the past. The data set involves seven users: *beck-s*, *farmer-d*, *kaminski-v*, *kitchen-l*, *lokay-m*, *sanders-r*, and *williams-w3*. Each of those directories has subfolders, where we characterize them as topical and non-topical (i.e. computer generated). We removed the non-topical folders from all the directories, examples include *calender*, *sent_items*, *sent*, *notes_inbox*, *discussion threads*, *_sent_mail*, *contacts*, *deleted_items*, *inbox* (See Table 2.4 for Enron dataset statistics). Next, we flatten all the folder hierarchies and remove all folders that contain less than three emails. In order to classify given data, we need to present the data in vector space. First, we start by tokenizing emails where we use extracted words to build dataset dictionary after all stop words and words which occur less

[5] http://www.cs.cmu.edu/ enron/

than three times are removed. Second, we present each document by a vector of counts, which is in turn normalized using L_2 normalization, thus, we start classification. In classification, we first sort all emails chronologically (according to the time stamp) and start training our classifier incrementally. Indeed, we split the emails into K splits each has the same portion of emails, say N. We start training on the first N emails and then test on the next N emails, then we train on the next $2N + 1$ emails and test on the following N and we continue until we reach the last split of emails.

Figure 2.7 shows the number of clusters obtained using LMM and GMM, to categorize emails into folders. According to this figure, LMM and GMM were able to find the exact number of clusters. This can be explained by the fact that MML criterion uses prior information that allows better comprehension to the application and data at hand. Figure 2.8 shows the influence of the number of training documents on classifier performance using different approaches. According to this figure, we clearly observe the incremental increase of performance of all classifier as the number of documents involved in the training increased. However, at some point the performance of generative models (i.e. LMM and GMM) has reached its optimal value when smaller number of training documents where used comparing to SVM which needed more documents. Table 2.5 shows the classification using LMM, GMM and SVM with different kernels, namely, polynomial kernel (SVM-p), RBF kernel (SVM-RBF) and sigmoid kernel (SVM-s) respectively. According to these results, when using LMM the average classification accuracy was better than the accuracy achieved by GMM which itself performs better than SVM classifier using different kernels. These results again prove that LMM is a good choice.

2.4.3 Spam email Filtering

One of the major problems of today's Internet is email spam. This problem is costly, poses serious security threats, and is getting worse. According to some studies spam emails constitute up to 75-80% of total amount of email messages and have caused financial losses of $50 billion in

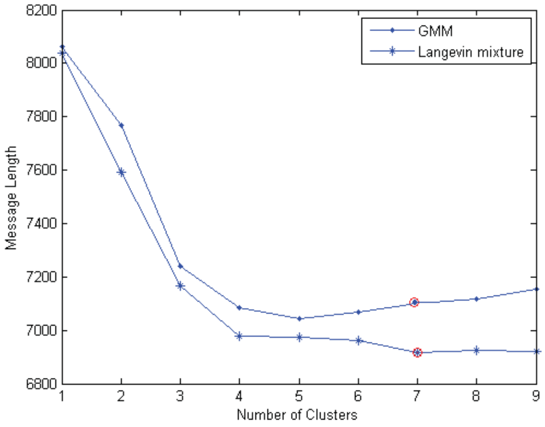


Figure 2.7: Number of clusters of Enron dataset found using MML when considering LMM and GMM.

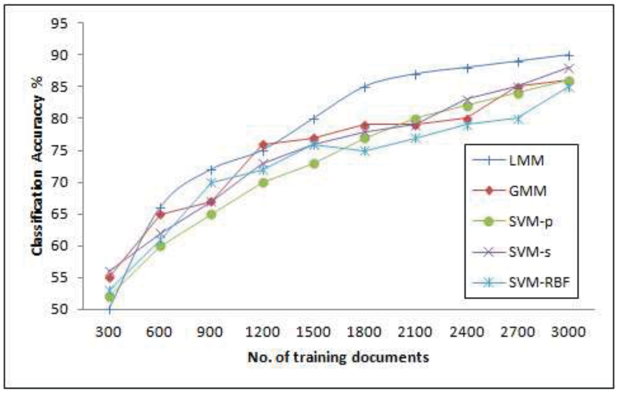


Figure 2.8: Classification accuracy as a function of the number of training documents.

2005 [73] and $130 billion in 2009 [74]. The main approach to build automated adaptive classifiers to discriminate legitimate and spam emails has been the content-based analysis of emails. In particular, the analysis of the semantic textual content via the adoption of text categorization techniques [75], based generally on machine learning and pattern recognition approaches, has received a lot of attention in the past and has achieved acceptable results as compared to techniques

Table 2.5: Classification performance (average±variance) obtained for Enron using LMM, GMM, SVM-p, SVM-RBF and SVM-s.

Enron User	LMM	GMM	SVM-p	SVM-RBF	SVM-s
$beck - s$	60.0 ± 1.5	55.34 ± 0.9	54.01 ± 0.1	55.9 ± 0.3	56.22 ± 0.2
$farmer - d$	80.12 ± 1.15	78.7± 0.8	70.1± 1.0	71.77± 0.4	72.51± 1.3
$kaminski - v$	70.4± 1.0	71.21 ± 0.2	68.19 ± 0.5	70.31 ± 0.8	66.54 ± 0.22
$kitchen - l$	66.78±3.4	61.13± 1.14	61.87± 0.1	60.24± 0.33	59.23± 1.02
$lokay - m$	81.98 ± 1.0	75.01 ± 3.2	71.42 ± 2.1	74.31 ± 1.0	73.22 ± 0.2
$sanders - r$	73.01± 1.1	69.1 ± 0.5	68.40 ± 0.6	66.35 ± 0.1	68.88 ± 1.3
$williams - w3$	93.2± 4.5	58.10± 1.4	59.11± 0.3	55.20± 1.0	57.39± 0.7

based on hand-made rules (see, for instance, [48, 76–78]). Many open sources and commercial spam filters, such as *SpamBayes* and *SpamAssassin*, currently adopt this approach where emails are described as bags of words (BoW). However, this approach has been defeated in the past by spammers using very simple tricks such as misspelling words or adding bogus text to their emails. Thus, some approaches went to analyze document space density [79], the non-content knowledge of the email social network [80, 81], or additional information presented by header of the email, time of delivery, the batch size of delivered emails, etc [82]. Unfortunately, the dynamic nature and the diversity of spam emails have easily circumvented the majority of the existing spam filters especially if we take into account the fact that email's content has massive shift from text content only to enriched multimedia content. Indeed, very recently researchers have figured out that text-based techniques might be ineffective because of a novel spammers trick namely image-based spam (i.e. email which includes embedded image) [83]. Studies conducted in 2006 suggested that more than 30% of all spam emails were image-based and most of these spams were undetected by filters.

Proposed Approach

In this chapter, we propose a novel approach that combines and uses simultaneously both textual and visual email information to filter spam emails within the hybrid framework we proposed in section 2.3. While the textual content is represented using the classic BOW formalism, the visual image spam information is represented as a bag of local descriptors extracted from detected image keypoints. To the best of our knowledge, there is no prior research work in considering simultaneously, within the same statistical framework, textual and low-level visual contents of spam. Particularly, previous work have considered statistical frameworks base on learning approaches using either generative (e.g. GMM) or discriminative models (e.g. SVM, maximum entropy). See Figure 2.9 for proposed spam filter.

Problem Statement: in this chapter, we consider a supervised email classification task. The main goal is to build a classifier using training emails in order to be able to correctly classify new unseen emails. Formally, consider that we have a corpus of emails $\mathcal{E} = \{E_1, \ldots, E_{|\mathcal{E}|}\}$ that were pre-classified under a set of categories $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$. We first split the corpus into two sets which are: training $\mathcal{E}_{Train} = \{E_1, \ldots, E_{|\mathcal{E}_{Train}|}\}$ set and testing $\mathcal{E}_{Test} = \{E_{|\mathcal{E}_{Train}|+1}, \ldots, E_{|\mathcal{E}|}\}$ set. Once email classifier has been constructed based on the characteristics of emails that were given in training set $\mathcal{E}_{Train}$. Then, test set $\mathcal{E}_{Test}$ is used to evaluate the effectiveness of the classifier, where each new email is fitted to a LMM and the classifier decision for each pair $(E_n, c_i) \subset \mathcal{E}_{Test} \times \mathcal{C}, 1 < n \leq |\mathcal{E}|$ is compared to the ground truth label. The class of a new email (in the test set) is based on the probabilistic distances developed in section 2.3 and hence

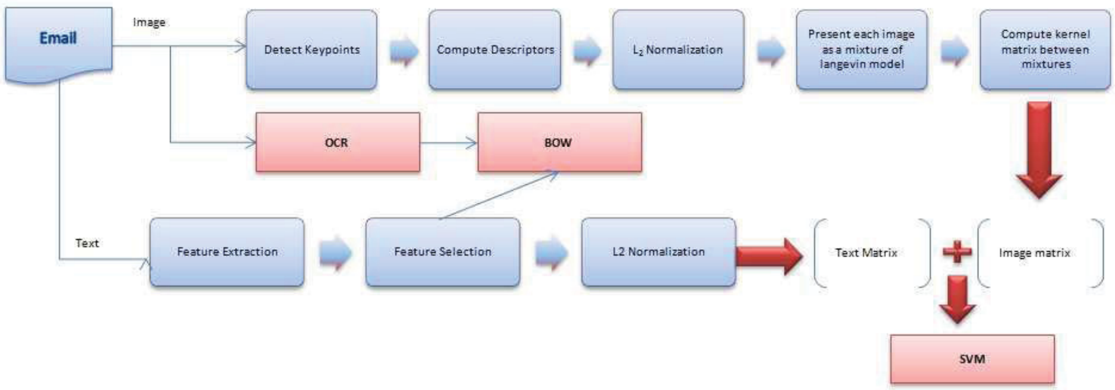


Figure 2.9: Spam Filter Architecture

the email is classified accordingly. In particular, given email E_n the classifier assigns a label $\mathcal{C} = \{c_1 = spam, c_2 = legitimate\}$ to each pair $(E_n, c_i) \in \mathcal{E} \times \mathcal{C}$.

Preprocessing of the Dataset The body of a given email E_n might have textual content only, graphical content (images) only or mix of both. In this chapter we build our framework based on the following assumptions. First, we consider the textual part and the graphical part simultaneously in the classification. Second, we assume that each email contains only one image. Third, text embedded in the images has the same importance as text extracted from the body and subject fields of the email and both extracted words are added to the same dictionary. In the case where no text can be found in the images, we still consider images in the classification.

For text preprocessing: we extracted features from the body, subject and header information presented in sender (From, Reply-to) and recipient (To, CC, Bcc) fields. Each email was tokenized using symbol delimiters (i.e. whitespace). Next, we removed words that only appear less than three times in whole emails. As a result, the initial number of unique terms is reduced from about 100000 to 45329 for each corpus. We didn't apply any feature selection, stemming and stop words as it has proven not to effect the accuracy of classifier [48]. Each email was presented by a single BoW vector. Therein, each component consists of feature (word) frequency in the dictionary $\mathcal{D}_{Term}$. Let $\vec{X}_i$ presents a given email described in terms of counts of features that appear in the dictionary $\vec{X}_i = (X_{i1}, \ldots, X_{ij}, \ldots, X_{i|\mathcal{D}_{Term}|})$, where X_{ij} presents the frequency of the j-th word in the dictionary that appears in the i-th email. In order to resist *sparse data attack* [22] we normalized each feature vector $\vec{X}_i$ using L_2 normalization $\parallel \vec{X}_i \parallel = \sqrt{\vec{X}_i^T \vec{X}_i}$. Then, we model the probability distribution for each class by LMM given in Eq.8. Subsequently, we computed the probabilistic distance presented in section 2.3 between each of these LMM giving us textual kernel matrix G_{Text} to feed SVM.

Moreover, in image preprocessing: for image spam we classified each image according to its textual and visual content. Tremendous efforts have been done for image spam preprocessing, particularly it is noteworthy that the choice of the features in order to discriminate future unseen legitimate emails from spam emails is crucial to the performance of the classifier. We chose local features since global representation of images is known to be sensitive to imaging conditions, noise and geometric transformations [84]. In particular, to extract visual features, first we use difference-of-Gaussian (DoG) to extract patches around detected interest points [85]. Then, we used SIFT descriptor computed on detected key points of all images where we used a 4×4 descriptor with 8 orientations, resulting in feature vectors with 128 dimensions for each detected interest point in each image [85]. Finally, we normalize extracted vectors using L_2 normalization. Formally, each image is presented now as $I = \{\vec{I_1}, \ldots, \vec{I_n}\}$ where n is the total number of detected key points and $\vec{I_i}$ is the SIFT vector associated with that detected point. Thus, we can model each image I by a mixture of Langevin distribution (see Eq. 8) and learn each LMM using algorithm 1. After all the LMM have been learnt for every image, as in textual part, we develop kernel matrix for image spam visual content G_{Visual} based on different probabilistic distances presented in section 2.3 between each of these mixtures models. Nonetheless, we simultaneously performed OCR to extract the written part (if presented) from the image and add it to the BOW of text extracted from email textual parts. The Tesseract OCR suite [6] was used to recognize the embedded text in images. Now, we need to discriminate spam emails from legitimate emails. In particular, we have two kernel matrices to feed SVMs classifier: kernel matrix for textual content of the email G_{Text} (includes text in the body and in the image) and kernel matrix for image spam visual content G_{Visual}. In order to simultaneously consider the textual and visual features we use Absolute Value (AV) approach that has proved its efficiency to combine kernels when it has high dimensional data. Using AV, gives us the ability to control the importance of information among given kernel matrices.

Evaluation Criteria

To evaluate the performance of spam email classifier, we calculate accuracy (Acc), weighted accuracy ($WAcc$), spam recall (SR), spam precision (See Table 2.6 for definitions) and Compute Receiver Operating Characteristic (ROC) curve. ROC graph is a visualization tool for selecting

[6] (open source by Google) http://code.google.com/p/tesseract-ocr/

Category set $\mathcal{C}$		Expert decision	
Decision	Folder c_i	$TP = \sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i$	$FP = \sum_{i=1}^{\lvert\mathcal{C}\rvert} FP_i$
	Folder $\bar{c}_i$	$FN = \sum_{i=1}^{\lvert\mathcal{C}\rvert} FN_i$	$TN = \sum_{i=1}^{\lvert\mathcal{C}\rvert} TN_i$

Table 2.6: Global Contingency Table for Spam Classifier.

classifiers based on their performance where it draws the tradeoff between hit rates and false alarm rates for a given classifiers. One of the main goals of our experiments is to calculate the conditional probability with respect to the category c_i, that is, the probability that set of emails were classified as spam they were truly spam (filter protection) and this can be measured using SP. Similarly, SR with respect to the category c_i measures the probability that if email is classified to be spam the filter will consider this decision (filter effectiveness to block). Moreover, to validate the effectiveness of classifying emails considering the cost of losing legitimate emails authors in [86] suggested the use of $WAcc$, in which we consider a certain threshold (λ) to correctly classifying legitimate emails instead of typical accuracy measure which assigns equal weights to the accurate classification of spam and legitimate emails. In particular, when a legitimate email is misclassified, this counts as λ error. Similarly, when legitimate email is correctly classified, this also counts as λ success. It is noteworthy, that λ[7] can be any value, but in our experiments we considered the same values (i.e. $\lambda = 1$, $\lambda = 9$, $\lambda = 999$) were given by [86]. SP, SR, Acc and $WAcc$ are thus given by:

$$\begin{aligned} SP &= \tfrac{TP}{TP+FP}, \qquad & Acc &= \tfrac{TP+TN}{TP+FN+TN+FP} \\ SR &= \tfrac{TP}{TP+FN}, \qquad & WAcc &= \tfrac{TP+\lambda TN}{TP+FN+\lambda(TN+FP)} \end{aligned} \tag{38}$$

Results

To evaluate the performance of our proposed hybrid framework for spam classification, we used publicly available datasets that have been used in the past. The trec05-p1 dataset contains [88]

[7] In [87] the threshold (t) has been set to 0.5, 0.9, 0.999, respectively, where $t = \frac{\lambda}{1+\lambda}$.

	Spam	legitimate
Spam	52400	345
legitimate	344	39100

(a)

	Spam	legitimate
Spam	50431	450
legitimate	2341	38967

(b)

	Spam	legitimate
Spam	1200	80
legitimate	20	230

(c)

	Spam	legitimate
Spam	1000	40
legitimate	50	440

(d)

	Spam	legitimate
Spam	52430	300
legitimate	339	39120

(e)

	Spam	legitimate
Spam	51200	200
legitimate	1801	38988

(f)

Table 2.7: Average rounded confusion matrices for Text-based 2.7(a)2.7(b), Image-based 2.7(c)2.7(d), "Text+ Image"-based 2.7(e)2.7(f) spam classifiers by LMM (left column) and GMM (right column).

92,189 emails, where 57% are spam emails. In addition, we extracted 1530 images form terc05-p1 including 1256 spam images and 274 legitimate images. Tables 2.7 show the average confusion matrices for spam filtering using LMM and GMM, respectively, when considering different scenarios (image, text, image+text). Obtained results show that "text+image"-based classifier provides a slight improvement over text-based classifier. However, image-based classifier reported the worst performance, that might be because of the lack of images in our dataset and particularly the images in legitimate emails. Moreover, in all cases results of LMM are more accurate and precise than those obtained using Gaussian mixture model.

Experiment 1: in the next experiment, we fed SVM spam classifier with probabilistic kernels generated from LMM and GMM (see section 2.3). When using LMM (See Table 2.8) the results vary significantly across kernels. For instance, SP results are between 89.30% and 69.34%

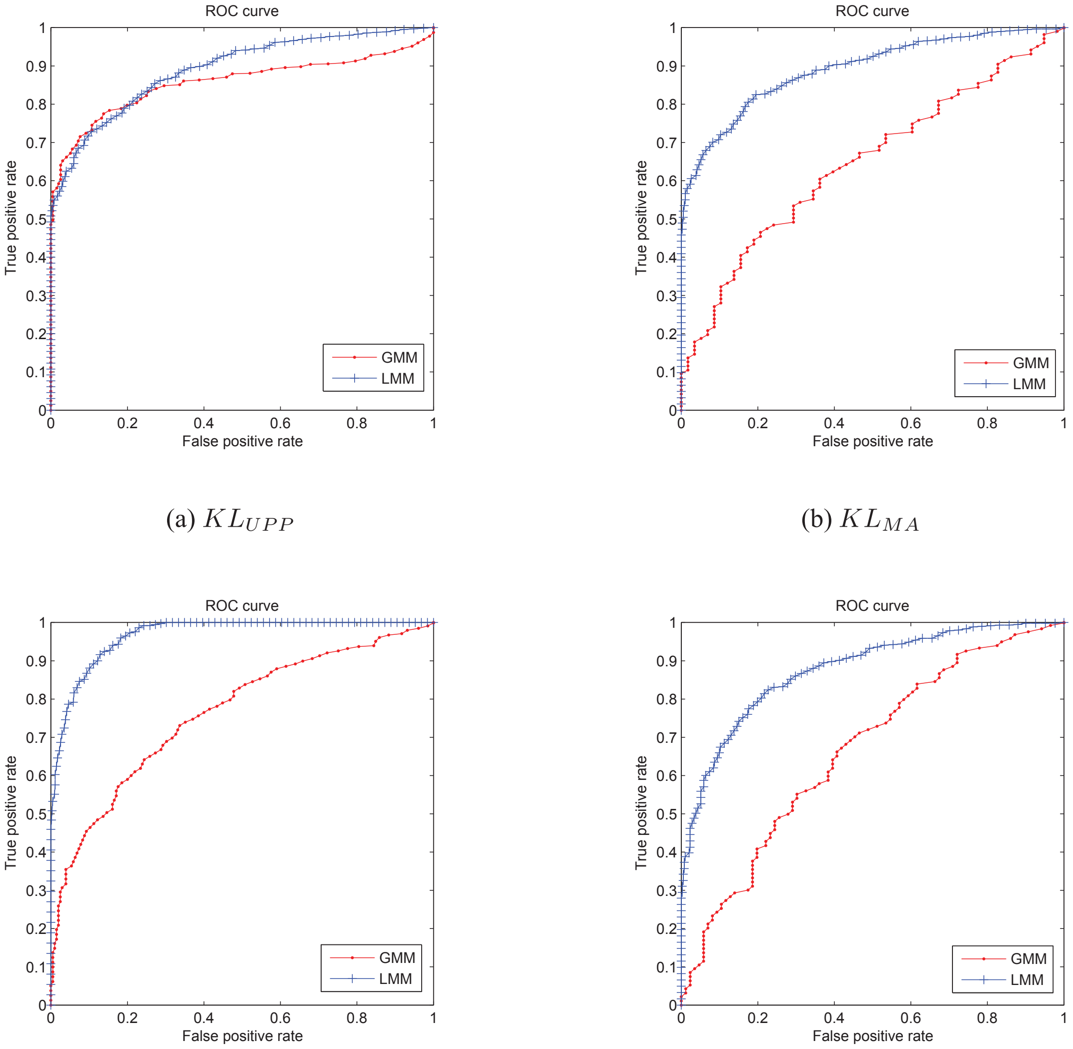
Table 2.8: Results (in%) on terc05-p1 using different tested kernels for LMM and GMM. All results reported in the table are for "Image +Text" as it has shown to provide best performance.

Kernels	LMM		GMM	
	SP	SR	SP	SR
KL_{VAR}	86.26	82.14	86.01	70.28
KL_{UPP}	89.30	87.43	81.00	79.49
KL_{MA}	82.41	84.12	85.07	81.61
ELK	75.29	69.15	56.98	66.01
BK	88.59	91.00	85.19	88.90
RK	83.10	89.81	88.41	88.09
JSK	81.16	82.89	81.10	84.56
FK	82.05	85.11	81.73	81.97

and SR results are between 91.00% and 53.00%. Furthermore, we can notice that the information divergence-based kernels (accuracy 90.74% for KL_{MA}) has a comparable results with PPK (95.92% for BK), while Fisher kernel has shown a slight degradation in the performance. In particular, ELK has the worst accuracy among those kernels. This degradation in performance supports the fact that linear kernels are not expressive enough for nonlinear high dimensional spaces. It is noteworthy that a ROC graphs (see Figure 2.10), with 95% confidence, show that the kernels generated from LMM tends to the left-top corner of the graph which supports the results previously presented in Table 2.8 that hybrid LMM model outperform hybrid GMM model. The cost sensitive evaluation on trec05-p1 shows that all kernels were quite sensitive to the higher weight of legitimate misclassification (see Table 2.9). By comparing $WAcc$, SP and SR, we find that overall kernels derived based on LMM is better choice than GMM. This can be justified by the fact that the Gaussian mixture, which clustering is based implicitly on the Euclidean distance or Mahalanobis, is inadequate for characterizing L_2 normalized data which clustering structure is better uncovered by considering the cosine similarity as assumed by the LMM.

Experiment 2: in order to compare our proposed hybrid framework with previous studies, experiments were conducted on datasets that were used recently in [89], namely, Princeton dataset[8] and Dredze et al. [90] (we will call it Dredze dataset in our experiments) dataset. Princeton dataset

[8] Available at http://www.princeton.edu/cass/spam/spam_bench/

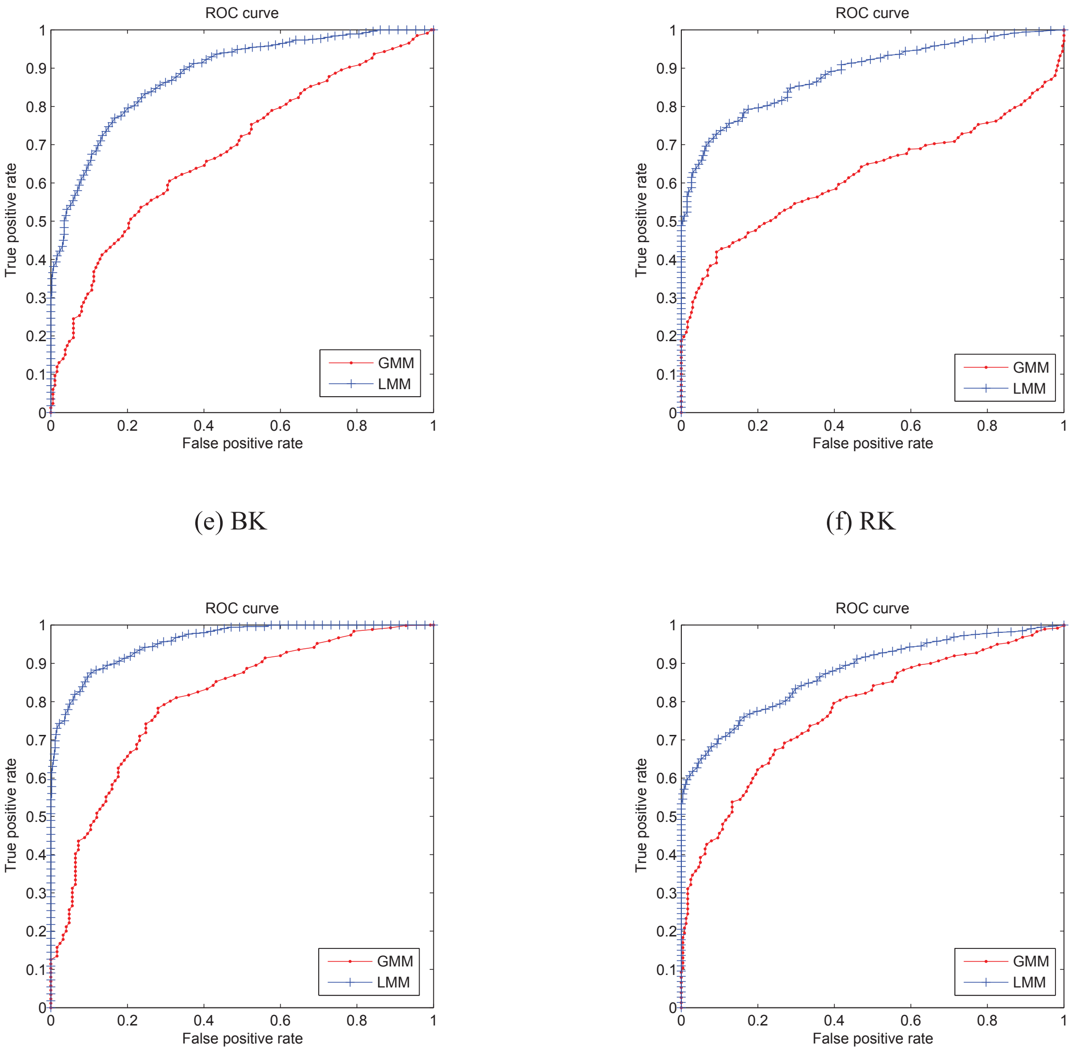


(a) KL_{UPP}

(b) KL_{MA}

(c) KL_{VAR}

(d) ELK



(e) BK

(f) RK

(g) JSK

(h) FK

Figure 2.10: ROC graphs for different kernels using LMM and GMM.

Table 2.9: $WAcc$ (in%) when considering terc05-p1 using LMM and GMM.

Kernel	LMM			GMM		
	$\lambda = 1$	$\lambda = 9$	$\lambda = 999$	$\lambda = 1$	$\lambda = 9$	$\lambda = 999$
KL_{VAR}	83.80	94.9	89.10	83.60	91.30	80.70
KL_{UPP}	91.70	78.00	87.80	89.10	76.50	78.80
KL_{MA}	70.50	83.00	61.00	71.70	76.90	59.70
ELK	87.10	69.50	85.90	84.60	66.30	81.30
BK	89.10	90.80	80.60	88.00	86.60	78.50
RK	88.90	96.10	95.70	86.00	93.90	79.00
JSK	90.00	97.70	95.80	79.50	82.38	79.91
FK	80.98	87.21	80.06	80.54	72.00	77.79

has 1071 emails which spread into 178 categories. Dredze dataset contains emails from publicly well-known SpamArchive datasets along with many personal emails of the authors. The main goal of this experiment is to investigate the hybrid framework based on SVMs probabilistic kernels between Lanegvin mixtures (see section 2.3) by following the approach proposed in [89]. The approach in [89] is based on modeling of the data using generative models, represented by Gaussian mixtures, and then using the Jensen-Shannon divergence to classify new emails. In the following experiments we used our hybrid framework (based on LMM and GMM) on the same datasets where we applied the same settings for a fair comparison. Following [89], we start by resizing all images to 100 $\times$100 pixels, which has been shown to have reasonable computation time and it is robust to pixels scaling and randomization. Then, we start extracting visual features from each image $\vec{I_i}$, namely, colors, pixel coordinates and texture. In particular, we presented each pixel in the image by a vector of seven tuples: two parameters for pixel coordinates (x_i, y_i), three for (L^*, a^*, b^*) color attributes and two for texture attributes for anisotropy and contrast features. Next, we L_2 normalize all vectors and subsequently we can model each image by a mixture of Langevin distributions using algorithm 2. Next, we split our datasets into two halves 75$\%$ for training and 25$\%$ for testing. A summary of the classification as displayed in table 2.10 clearly shows that our hybrid framework achievers superior performance than that in [89] where the best result achieved was 84$\%$.

A potential future work can focus on online learning where the learning will be updated and re-

Table 2.10: Accuracy (in$\%$) of different approaches on Princeton and Dredze dataset

Kernel	LMM	GMM
KL_{VAR}	91.01	84.09
KL_{UPP}	82.24	79.22
KL_{MA}	83.11	78.58
ELK	85.89	82.00
BK	90.61	86.56
RK	89.72	77.03
JSK	93.45	87.4

fined to be able to adapt the dynamic nature of multimedia data. Furthermore, we can investigate proposed approach on different applications.

Chapter 3

Simultaneous Online Spherical Data Clustering and Feature Selection

In this chapter we start by a brief introduction to movM and describe our feature selection approach. Next, we propose an approach to learn our model parameters and to find the optimal number of components. Later, we propose an online learning algorithm for our simultaneous clustering and feature selection model. Finally, the merits of proposed approaches are validated through extensive experiments.

3.1 Introduction

Finite mixture models are among the most applied approaches to data clustering [32]. Finite mixture models as a formal approach to unsupervised learning allows the extraction of hidden structure in data which results in salient and compact representation. Thereby, to group these objects, beforehand, each object is usually preprocessed in order to generate input in a form usable by mixtures. The dominant approach is the description of objects as vectors of features by finding an effective representation space (i.e. feature engineering) [91]. Indeed, features can be used as means of intelligently describing objects, so that one can simply extract and select diverse features and then decides which features require most attention according to the problem at hand [92]. To illustrate, using finite mixtures, the assignment of $\vec{X}_i = (X_{i1}, \dots, X_{iD})$ to cluster j is defined by the membership that is given as a posterior probability according to the Bayes decision rule. However, not all the D features in the document $\vec{X}_i$ contribute equally to the determination of cluster membership. To cope with this problem, many feature selection approaches have been devoted to select the subset of relevant (i.e. most informative) features that boost the performance of the models and hold good generalization to unseen data [92].

Generally, there are many potential advantages of feature selection such as resisting the curse of dimensionality, helping in construction of realistic models that are more flexible, have a room for error recovery and more efficient in terms of time and storage saving. Feature selection importance can be seen in wide range of practical problems such as detection of spam emails [48] and text clustering [93]. Indeed, if we take the example of text clustering, in most cases the vocabulary of features (i.e. words) is large and the vectors of features are extremely sparse. Not all features presented in the vocabulary contribute equally to classification such as stop words which have been shown to degrade clustering performance. Thus, selecting the most relevant features to reduce the dimensionality of feature space is an important step that generally improve generalization capabilities of the model [94–99].

The majority of research works on feature selection, based on finite mixtures, have been blindly directed to develop generic models that do not take the nature of the data into account. In spite of that, there has been some recent works [93, 94, 100–102] directed for particular data types in the literature. For instance, authors in [100] have proposed a simultaneous clustering and feature selection approach for non-Gaussian data applied to images and videos segmentation. Feature selection approaches for proportional, discrete and binary data have been proposed in [101], [102] and [93], respectively, and successfully applied to the problems of documents classification and images categorization. In contrast to these previous efforts, the work in this chapter is devoted to high-dimensional spherical vectors. In previous works, we have shown that we can infer this kind of data (i.e. L_2-normalized) via spherical distributions. For instance, movM models L_2-normalized data that reside on unit circle [2, 103, 104]. The movM has been a subject of study in wide spectrum of areas ranging from biology, medicine, geology [26], Deoxyribonucleic acid (DNA) microarray analysis [105] to pattern recognition, data mining and computer vision as in video surveillance [106], image classification [107] and audio signals modeling [108]. Thus, in this work we shall consider movM in our statistical framework.

Our approach combines clustering and feature selection based on movM learned using EM. The interrelatedness challenge to find the optimal number of clusters while assessing the features relevancy is tackled also using MML criterion [50] that permits model selection. In addition, we illustrate the potential of hybrid generative discriminative frameworks upon integrating feature selection, based on both movM and Support Vector Machines (SVM), on data described by bags of L_2-normalized vectors. Despite the numerous researches on feature selection, to the best of our knowledge none of them has considered the case where these feature vectors are spherical so far.

In applications where time plays an increasingly essential role, defining a prior subset of informative relevant features for diverse problems is not necessarily satisfactory to cope with new features that may be introduced continuously in online learning [109, 110]. Recently, researchers tend to update models incrementally [111–115]. Unlike supervised learning [114], online learning in unsupervised manner confront several challenges on error constraints, restoring objects labels, finding optimal number of model clusters and finding relevant features at each time step. In [116], an algorithm for online learning of von Mises Fisher mixture (a.k.a Langevin mixture [2]) was proposed based on online spherical K-means for text clustering. The model proposed in [116], however, updates clusters centroids up on the arrival of new documents while keeping the variance and mixing proportions unchanged. Author in [117] proposed a recursive version of EM (RSEM) algorithm to update mixture model parameters based on stochastic gradient descent[1] in the case of Gaussian

[1] It is noteworthy that stochastic gradient have been utilized in different approaches for incremental updates due to its simplicity, its ability to scale to large dataset without losing the accuracy and its fast convergence as compared to other approaches [118–120]. Accordingly, other online clustering algorithms are possible, for instance, in [121] authors proposed online classification EM (CEM) based on the stochastic gradient ascent algorithm, where they proved the superior performance of online CEM compared to online k-means algorithm. Another example is using Stochastic Gradient Descent in reproducing Hilbert space for online learning of SVM [118]. The comparison of different learning algorithms is beyond the scope of this chapter, interested reader may refer to [110, 118–120] and references therein.

mixture model (GMM). The RSEM was also used in [111] where the authors proposed online finite Dirichlet mixture to the problem of image databases summarization. Yet, previous works do not consider the problem of feature selection. In contrast to these previous works, we propose unsupervised approach that incrementally learns and adjusts the weights of all features. The challenge here is in defining a subset of relevant and irrelevant features at each time step which yields comparably to better optimal performance rate. We approach this problem by extending RSEM to simultaneously consider feature relevancy while gradually updating given model. In this chapter, our main contributions are:

- We first propose [33] a simultaneous feature selection and clustering in the case of spherical data in off-line settings. However, this is limited solution to many dynamic problems when the data appears as a steam of sequence with time. To this aim, in this chapter, we extend this work to online settings extending RSEM to simultaneously consider feature relevancy while gradually updating given model.
- We tackle the problem of automatic determination of the number of components (i.e. model selection) of Von Mises mixture model in both off-line and online settings by developing a minimum message length objective that was minimized using EM in off-line scenario and RSEM for online scenario.
- Due to the fact that hybrid generative discriminative frameworks have shown improved performance as compared to their generative or discriminative counterparts, we propose the combination of movM and SVM upon integrating feature selection. Indeed, we develop a hybrid framework that models image descriptors, in an unsupervised way, using movM from which Fisher kernel is generated for SVMs.
- We present detailed comparison of the proposed Framework using Von Mises mixture model with the widely used Gaussian mixture model (GMM). We discuss the properties of proposed framework on abundant (hundreds of thousands), high-dimensional, directional and challenging data: Yahoo20 dataset (used for web categorization), and Dredze dataset (used for spam filtering).

3.2 Mixture Density and Feature Selection

Given a dataset $\mathcal{X} = \{\vec{X}_1, \ldots, \vec{X}_N\}$ of N documents (or images). Therein, each document (image) $\vec{X}_i = (X_{i1}, \ldots, X_{iD})$ is described by a L_2-normalized D-dimensional feature vector, such that $(\vec{X}_i)^T \vec{X}_i = 1$. As we consider L_2 normalized vectors, then each document is best modeled using von Mises (vM) distribution. The vM distribution[2] [26] is a probability distribution in which the data are concentrated on the circumference of a unit circle. In particular, we shall suppose that the features in each vector $\vec{X}_i$ are independent and follows a vM distribution which gives the following:

$$p(\vec{X}_i|\vec{\mu}, \vec{\kappa}) = \prod_{d=1}^{D} p(\vec{Y}_{id}|\theta_d) = \prod_{d=1}^{D} \frac{1}{2\pi I_0(\kappa_d)} \exp\{\kappa_d \vec{\mu}_d^T \vec{Y}_{id}\} \tag{1}$$

where I_0 is the modified Bessel function of the first kind and order zero [24], $\theta_d = (\vec{\mu}_d, \kappa_d)$, $\vec{\mu} = (\vec{\mu}_1, \ldots, \vec{\mu}_D)$, $\vec{\mu}_d = (\mu_{d1}, \mu_{d2})$ is the mean direction, $\vec{\kappa} = (\kappa_1, \ldots, \kappa_D)$, κ_d is the concentration parameter [3] and $\vec{Y}_{id} = (Y_{id1}, Y_{id2})$ such that $Y_{id1} = X_{id}$ and $(\vec{Y}_{id})^T \vec{Y}_{id} = 1$ and D is the number of features. Generally, a set of vectors is comprised of examples that vary in their characteristics and represent dissimilar information and hence belong to many clusters which can be modeled by a finite mixture of distributions. Thus, let $p(\vec{X}_i|\Theta_M)$ be a mixture of M distributions represented by Eq. 1. The probability density function of a M-components movM is given by

$$p(\vec{X}_i|\Theta_M) = \sum_{j=1}^{M} p_j \prod_{d=1}^{D} p(\vec{Y}_{id}|\theta_{jd}) \tag{2}$$

where $\Theta_M = \{\vec{P} = (p_1, \ldots, p_M), \vec{\theta}_{jd} = (\theta_{1d}, \ldots, \theta_{Md})\}$ denotes all the parameters of the mixture model such that $\theta_{jd} = (\vec{\mu}_{jd}, \vec{\kappa}_{jd})$ are the parameters of the j^{th} movM component for feature d, p_j represents the weight of the j^{th} movM component and $\vec{P}$ is the vector of mixing parameters that are positive and sum to one.

The clustering based on finite mixture models is explored by grouping similar documents, where

[2] Also known as the circular normal distribution [26, 122] and maximum entropy distribution in [123].

[3] κ_d reflects the dispersion of the distribution and can be viewed then as the analogue of the inverse of the variance (i.e. invariance or reciprocal of the variance) in the case of the Gaussian [124, 125].

this similarity depends basically on the features that represent each document. Indeed, researchers have proven over the years the fallacy assumption that the more features representing the document the better discrimination capability the classifier has [95–98]. This can be due to the presence of noisy and non-informative (i.e. irrelevant) features that generally highly drop the performance. In order to overcome this problem, we adopt the approach proposed in [94], in the case of the Gaussian mixture, that assigns smaller weights to irrelevant features by introducing the notion of feature saliency using the assumption that a given feature is irrelevant if it follows a common density $p(\vec{Y}_{id}|\lambda_{jd})$ across clusters while maintaining the independency of class labels [94, 101, 102]. In particular, each d^{th} feature is represented by movM of two components $p(\vec{Y}_{id}|\theta_{jd})$ and $p(\vec{Y}_{id}|\lambda_{jd}))$ governed by ρ_{jd} that denotes the weight of the d^{th} feature on cluster j. In fact, if ρ_{jd} is very high, then there is no significant difference with the classical movM model $p(\vec{Y}_{id}|\theta_{jd})$ without any saliency. Thus, our model, to take feature selection into account, can be written as:

$$p(\vec{X}_i|\Theta) = \sum_{j=1}^{M} p_j \prod_{d=1}^{D} (\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1 - \rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})) \tag{3}$$

where $\Theta = \{\Theta_M, \{\rho_{jd}\}, \{\lambda_{jd}\}\}$, $\lambda_{jd} = (\vec{\mu}_{jd|\lambda}, \kappa_{jd|\lambda})$ are the parameters of vM from which the irrelevant feature is drawn.

3.3 Unsupervised Learning of the Model

In the following, we present our unsupervised learning approach for simultaneous clustering and feature selection. In particular, we propose an approach to find the optimal number of model components and to estimate the different parameters.

3.3.1 Model Selection

One of finite mixture learning's crucial factors is the determination of the optimal number of components that best describe the data. The model selection problem can be viewed as one that helps finding the optimal trade-off between the complexity of the model and goodness of fit. Over the years, many approaches have been proposed [55–57, 126]. In this work, we propose the consideration of MML criterion to find the optimal number of mixture components by minimizing the following function [126]:

$$MessLen(M) \simeq -\log h(\Theta) + \frac{1}{2}\log|F(\Theta)| + \frac{N_p}{2}(1+\log\frac{1}{12}) - \log p(\mathcal{X}|\Theta) \tag{4}$$

where $h(\Theta)$ is the prior probability, $p(\mathcal{X}|\Theta)$ is the likelihood, $F(\Theta)$ is the expected Fisher information matrix which is generally approximated by complete-data Fisher information matrix in the case of finite mixture models, $|F(\Theta)|$ is its determinant, and $N_p = M(1+5D)-1$ is the number of free parameters to be estimated in the case of our unsupervised feature selection model.

In order to define $MessLen$ for our model, in what follow, we assume the independence of the different groups of parameters, which facilitates the factorization of both prior $h(\Theta)$ and Fisher information matrix $F(\Theta)$. In particular, this yields to the following prior distribution over the parameters:

$$h(\Theta) = h(\vec{P})\prod_{j=1}^{M}\prod_{d=1}^{D} h(\rho_{jd})h(\theta_{jd})h(\lambda_{jd}) \tag{5}$$

with [94]:

$$h(\vec{P}) \propto \prod_{j}^{M} p_j^{-D}, \quad h(\rho_{jd}) \propto [\rho_{jd}(1-\rho_{jd})]^{-M} \tag{6}$$

Moreover, we consider the following priors that we found efficient through our experiments for the concentration [4] and mean parameters:

$$h(\vec{\mu}_{jd}) = \frac{1}{2\pi}, \quad h(\vec{\kappa}_{jd}) = \frac{2}{\pi(1+\kappa_{jd}^2)}, \quad h(\vec{\mu}_{jd|\lambda}) = \frac{1}{2\pi}, \quad h(\kappa_{jd|\lambda}) = \frac{2}{\pi(1+\kappa_{jd|\lambda}^2)} \tag{7}$$

[4] Another prior is possible: $\kappa_{jd} = \frac{\kappa_{jd}}{(1+\kappa_{jd}^2)^{\frac{3}{2}}}$ as in [127]. However, according to [123] the prior we consider in Eq.7 has shown superior performance.

For Fisher Information matrix, we replace $F(\Theta)$ by complete data Fisher information matrix, which has a block diagonal structure

$$\begin{aligned} F_c(\Theta) = \text{block-diagonal}[F(\vec{P}), \frac{1}{\rho_{11}(1-\rho_{11})}, \ldots, \frac{1}{\rho_{MD}(1-\rho_{MD})}, \\ p_1\rho_{11}F(\theta_{11}), \ldots, p_M\rho_{MD}F(\theta_{MD}), \\ p_1(1-\rho_{11})F(\lambda_{11}), \ldots, p_M(1-\rho_{MD})F(\lambda_{MD})] \end{aligned} \tag{8}$$

where $F(\vec{P})$ is the Fisher information of the mixing proportions $\vec{P}$, $(\rho_{jd}(1-\rho_{jd}))^{-1}$ for $j = 1, \ldots, M$ and $d = 1, \ldots, D$ is the information matrix $F(\rho_{jd})$ corresponding to ρ_{jd}, $F(\theta_{jd})$ and $F(\lambda_{jd})$ are the information matrices related to model parameters θ_{jd} for relevant features and λ_{jd} for irrelevant features, respectively. Thus, the determinant of complete Fisher information matrix $|F_c(\Theta)|$ is given by:

$$|F_c(\Theta)| = |F(\vec{P})| \prod_{j=1}^{M} |F(\theta_j)||F(\lambda_j)| \Big[\prod_{d=1}^{D} |F(\rho_{jd})|\Big] \tag{9}$$

Following [101], we can approximate the determinant of the Fisher information matrix of $\vec{P}$ and ρ_{jd} as follow:

$$|F(\vec{P})| = N^{(M-1)} \prod_{j=1}^{M} \frac{1}{p_j}, \quad |F(\rho_{jd})| = \frac{N}{\rho_{jd}(1-\rho_{jd})} \tag{10}$$

As for $F(\theta_{jd})$ and $F(\lambda_{jd})$, we need to consider expected value of the negative log likelihood for each feature in $\vec{X}_i$ separately. Thus,

$$|F(\theta_j)| = N_j^2 \prod_{d=1}^{D} |F_1(\theta_{jd})|, \quad |F(\lambda_j)| = N_j^2 \prod_{d=1}^{D} |F_1(\lambda_{jd})| \tag{11}$$

where N_j is the number observations affected to cluster j and $|F_1(\theta_{jd})|$ is given by [123]:

$$F_1(\theta_{jd}) = \begin{pmatrix} \kappa_{jd}A(\kappa_{jd}) & 0 \\ 0 & A'(\kappa_{jd}) \end{pmatrix}$$

where $A(\kappa_{jd}) = \frac{I_1(\kappa_{jd})}{I_0(\kappa_{jd})}$ and $A'(\kappa_{jd}) = 1 - (\frac{I_1(\kappa_{jd})}{I_0(\kappa_{jd})})^2 - \frac{I_1(\kappa_{jd})}{I_0(\kappa_{jd})}$. Similarly, we can find $F_1(\lambda_{jd})$. By substituting $\log|F_c(\Theta)|$ and $\log h(\Theta)$ for model parameters Θ, we can find:

$$\begin{aligned} MessLen(M) \quad &= \frac{1}{2}(M + 5MD - 1)\log N + \frac{D}{2}\sum_{j=1}^{M}\log p_j + \frac{M}{2}\sum_{d=1}^{D}\log \rho_{jd} \qquad (12) \\ &+ \frac{M}{2}\sum_{d=1}^{D}\log(1-\rho_{jd}) + \frac{1}{2}\sum_{j=1}^{M}\sum_{d=1}^{D}\log(|F_1(\theta_{jd})|) \\ &+ \frac{1}{2}\sum_{j=1}^{M}\sum_{d=1}^{D}\log(|F_1(\lambda_{jd})|) + \frac{N_p}{2}(1 + \log\frac{1}{12}) - \log p(\mathcal{X}|\Theta) \\ &+ \sum_{j=1}^{M}\sum_{d=1}^{D}[4\log\pi + \log(1+\kappa_{jd}^2) + \log(1+\kappa_{jd|\lambda}^2)] \end{aligned}$$

It is worth mentioning that Eq. 12 is MML-Laplace which can be simplified to MML-Jeffrey [33] by adopting Jeffrey's prior for Θ [101]. In what follow, we will use MML-Laplace to learn our model.

3.3.2 Parameter Estimation

In this section, we develop the equations that learn movM while simultaneously considering the relevancy of features. To achieve this goal, we adopt common EM approach which generates a sequence of models with non-decreasing log-likelihood on the data. The main goal is to optimize the following objective function:

$$S(\Theta, \mathcal{X}) = -MessLen(M) + \xi\Big(1 - \sum_{j=1}^{M} p_j\Big) + \sum_{d=1}^{D}\nu_{jd}\Big(1 - \rho_{jd1} - \rho_{jd2}\Big) \qquad (13)$$

where $\rho_{jd1} = \rho_{jd}$ and $\rho_{jd2} = 1 - \rho_{jd}$, ξ and ν_{jd} are Lagrange multipliers to satisfy the constraints $\sum_{j=1}^{M} p_j = 1$ and $\rho_{jd1} + \rho_{jd2} = 1$, respectively. Thus, straightforward manipulations allow us to obtain the following by maximizing Eq. 13 (See Appendices A, B, C, D):

$$\hat{Z}_{ij} = \frac{p_j \prod_{d=1}^{D}\Big(\rho_{jd}p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd})p(\vec{Y}_{id}|\lambda_{jd})\Big)}{\sum_{j=1}^{M} p_j \prod_{d=1}^{D}\Big(\rho_{jd}p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd})p(\vec{Y}_{id}|\lambda_{jd})\Big)} \qquad (14)$$

$$p_j = \frac{\max(\sum_{i=1}^{N} \hat{Z}_{ij} - D, 0)}{\sum_{j=1}^{M} \max(\sum_{i=1}^{N} \hat{Z}_{ij} - D, 0)} \tag{15}$$

$$\rho_{jd} = \frac{\max\Big(\sum_{i=1}^{N} \hat{Z}_{ij} \frac{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd})}{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})} - M, 0\Big)}{\nu_{jd}} \tag{16}$$

where

$$\begin{aligned}\nu_{jd} = &\max\Big(\sum_{i=1}^{N} \hat{Z}_{ij} \Big[\frac{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd})}{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})}\Big] - M, 0\Big) \\ &+ \max\Big(\sum_{i=1}^{N} \hat{Z}_{ij} \Big[\frac{(1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})}{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})}\Big] - M, 0\Big)\end{aligned}$$

$$\vec{\mu}_{jd} = \frac{\sum_{i=1}^{N} \hat{Z}_{ij} \frac{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd})}{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})} \vec{Y}_{id}}{\sum_{i=1}^{N} \hat{Z}_{ij} \frac{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd})}{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})}} \tag{17}$$

$$\vec{\mu}_{jd|\lambda} = \frac{\sum_{i=1}^{N} \hat{Z}_{ij} \frac{(1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})}{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})} \vec{Y}_{id}}{\sum_{i=1}^{N} \hat{Z}_{ij} \frac{(1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})}{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})}} \tag{18}$$

and

$$\vec{\mu}_{jd} = \frac{\vec{\mu}_{jd}}{\|\vec{\mu}_{jd}\|}, \qquad \vec{\mu}_{jd|\lambda} = \frac{\vec{\mu}_{jd|\lambda}}{\|\vec{\mu}_{jd|\lambda}\|} \tag{19}$$

Since, we cannot find tractable form for $A^{-1}(\kappa_{jd})$, we use Newton-Raphson iterations to find κ_{jd}, where:

$$\kappa_{jd}^{new} = \kappa_{jd}^{old} - \frac{\partial S(\Theta, \mathcal{X})}{\partial \kappa_{jd}} \Big(\frac{\partial^2 S(\Theta, \mathcal{X})}{\partial^2 \kappa_{jd}}\Big)^{-1} \tag{20}$$

Similarly we can calculate $\kappa_{jd|\lambda}^{new}$. Having our selection criterion and estimation equations in hand, the complete learning algorithm is as follows:

Algorithm 1. For each candidate value of M:

Step 0: Apply spherical K-means [53] to obtain the initial parameters for each component.

Step 1: Iterate the two following steps until convergence:

1. **E-Step**: Update $\hat{Z}_{ij}$ using Eq. 14

2. **M-Step**: Update p_j, ρ_{jd}, $\vec{\mu}_{jd}$, $\vec{\mu}_{jd|\lambda}$, κ_{jd} and $\kappa_{jd|\lambda}$ using Eqs. 15,16, 19, 20, respectively.

Step 2: Calculate the associated message length $MessLength(M)$ using Eq 12.

- Select the optimal model M^* such that $M^* = \arg\min MessLength(M)$.

Note that the computational cost of proposed framework is based on the EM estimation framework cost where both E- and M-steps have a complexity of $O(NMD)$ which is the same complexity associated with standard EM-based learning approaches.

3.4 Online Learning with Feature Selection

As aforementioned, in many dynamic applications data appear in stream of sequences and hence online learning is favored over off-line counterpart. The main idea is to update mixture model parameters incrementally as data are presented to the classifier [117]. Formally, assume that at time t we have a dataset $\mathcal{X} = \{\vec{X}_1, \ldots, \vec{X}_N\}$ of N documents which is represented by a M-component movM with parameters Θ_N^t. At time $t+1$ a new document $\vec{X}_{N+1}$ is introduced and the model should be updated considering the new document. In this section, our intention is to develop flexible and accurate online mixture model that lets us to simultaneously choose relevant features and optimal number of model components. To achieve this goal, in the following we adopt RSEM approach proposed in [117]. The same way as EM, the RSEM is mainly obtained by computing the conditional expectation of the complete available data at time $t+1$ in the E-step, thus:

$$P(\vec{X}_{N+1}, \vec{Z}_{N+1}, \vec{\Lambda}_{N+1}) = p(\vec{Z}_{N+1})p(\vec{\Lambda}_{N+1})p(\vec{X}_{N+1}|\vec{Z}_{N+1}, \vec{\Lambda}_{N+1}) \tag{21}$$

and

$$\begin{aligned} E[\log P(\vec{X}_{N+1}, \vec{Z}_{N+1}, \vec{\Lambda}_{N+1})] \quad &= \sum_{i=1}^{N}\sum_{j=1}^{M} P(Z_{N+1,j} = j|X_{N+1,d}) \log p_j \\ &+ \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{d=1}^{D}\sum_{\phi=0}^{1} P(Z_{N+1,j} = j, \phi_{jd}|X_{N+1,d}) \\ &\times \Big[\phi_{jd}\Big(\log p(Y_{N+1,d}|\theta_{jd}) + \log \rho_{jd}\Big) \\ &+(1-\phi_{jd})\Big(\log p(Y_{N+1,d}|\lambda_{jd}) + \log(1-\rho_{jd})\Big)\Big] \end{aligned}$$

where $(\vec{X}_{N+1}, \vec{Z}_{N+1}, \vec{\Lambda}_{N+1})$ is the complete data, and $\vec{Z}_{N+1} = (Z_{N+1,1}, \ldots, Z_{N+1,M})$ is a random vector of missing data that indicates if the new document is associated with component j, such as $Z_{N+1,j} = 1$ if the new vector $\vec{X}_{N+1}$ belongs to class j, 0 otherwise. $\vec{\Lambda}_{N+1} = \{\phi_{11}, \ldots, \phi_{MD}\}$ is a random vector of missing data that indicates if feature d is relevant to cluster j, where $\rho_{jd} = p(\phi_{jd} = 1)$.

Using RSEM, in the M-step we update the model parameters with respect to $\Theta^t = \{\Phi^t = \{\mu^t_{jd}, \kappa^t_{jd}, \mu^t_{jd|\lambda}, \kappa^t_{jd|\lambda}\}, \{\rho^t_{jd}\}\}$ and with the constraints $0 < p_j \leq 1$ and $\sum_{j=1}^{M} p_j = 1$ [117]:

$$\begin{cases} w(j)^{t+1} = w(j)^t + \gamma_N(\hat{Z}_{N+1,j} - p^t_j), \quad 1 \leq j < M \\ \Phi^{(t+1)} = \Phi^{(t)} + \gamma_N \frac{\partial \log(p(\vec{X}_{N+1}, \vec{Z}_{N+1}, \vec{\Lambda}_{N+1}|\Phi^{(t)}))}{\partial \Phi}, \quad 1 \leq j \leq M \end{cases} \tag{22}$$

where

$$\hat{Z}_{N+1,j} = \frac{p^t_j p(\vec{X}_{N+1}|\Theta^t)}{\sum_{j=1}^{M} p^t_j p(\vec{X}_{N+1}|\Theta^t)} \tag{23}$$

is the posterior probability, and γ_N represents any sequence of positive numbers which decreases to zero or positive definite matrix such that $\sum |\gamma_N| = \infty$ and $\sum |\gamma_N|^2 < \infty$. In our case we have chosen $\gamma_N = \frac{1}{N+1}$ [111, 117]. Note that $w(j)^{t+1}$ in Eq. 22 is used to ensure the unity of the mixing proportion p_j by introducing the Logit transform $w(j) = \log \frac{p_j}{p_M}$ such that $w_M = 0$, where:

$$p^{t+1}_j = \frac{\exp(w(j)^{t+1})}{1 + \sum_{j=1}^{M-1} \exp(w(j)^{t+1})}, \quad j = 1, ..., M-1 \tag{24}$$

$$p^{t+1}_M = \frac{1}{1 + \sum_{j=1}^{M-1} \exp(w(j)^{t+1})} \tag{25}$$

In order to figure out the relevancy of features for the new vector, we need to update ρ_{jd} such that $\rho_{jd} \in [0, 1]$. Let $\rho_{jd1} = \rho_{jd}$ and $\rho_{jd2} = 1 - \rho_{jd}$ such that $\rho_{jd1} + \rho_{jd2} = 1 \; \forall d = 1, \dots, D$. Hence, we propose to use parametrization based on Logit transform $h_{jd} = \log(\rho_{jd}) = \log \frac{\rho_{jd}}{1-\rho_{jd}}$, and we obtain:

$$h_{jd}^{t+1} = h_{jd}^{t} + \frac{\hat{\Lambda}_{jd}}{N+1} \Big[\frac{\partial \log(p(\vec{X}_{N+1}, \vec{Z}_{N+1}, \vec{\Lambda}_{N+1} | \rho_{jd}^{(t)}))}{\partial \rho_{jd}} \Big] \tag{26}$$

$$\rho_{jd1}^{t+1} = \frac{\exp(h_{jd}^{t+1})}{1 + \exp(h_{jd}^{t+1})}, \quad d = 1, \dots, D \tag{27}$$

$$\rho_{jd2}^{t+1} = \frac{1}{1 + \exp(h_{jd}^{t+1})}, \quad d = 1, \dots, D \tag{28}$$

where

$$\hat{\Lambda}_{jd} = \begin{cases} \hat{Z}_{N+1,j} \dfrac{\rho_{jd}^{t} p(Y_{N+1,d} | \theta_{jd}^{t})}{\rho_{jd}^{t} p(\vec{Y}_{N+1,d} | \theta_{jd}^{t}) + (1 - \rho_{jd}^{t}) p(\vec{Y}_{N+1,d} | \lambda_{jd}^{t})}, & \text{If } d \text{ is relevant} \\ \hat{Z}_{N+1,j} \dfrac{(1 - \rho_{jd}^{t}) p(Y_{N+1,d} | \lambda_{jd}^{t})}{\rho_{jd}^{t} p(\vec{Y}_{N+1,d} | \theta_{jd}^{t}) + (1 - \rho_{jd}^{t}) p(\vec{Y}_{N+1,d} | \lambda_{jd}^{t})}, & \text{If } d \text{ is irrelevant} \end{cases} \tag{29}$$

Note that it is straight forward to calculate ρ_{jd2} based on Eq. 27. Finally, the RSEM algorithm in the case of our model can be defined as follows:

Algorithm 2. For each candidate M:

Step 0: (at iteration t) Initialization: $\Theta_{M_{min}}^{(t)}, \dots, \Theta_{M_{max}}^{(t)}$

Step 1: (at iteration $t+1$) new document $\vec{X}_{N+1}$ is introduced.

- Compute the posterior probabilities using Eq. 23.
- Assign $\vec{X}_{N+1}$ to cluster which maximizes $\hat{Z}_{N+1,j}$

Step 2: (at iteration $t+1$) Update the different mixtures models using Eqs. 22, 24 and 27 for $j \in \{M_{min}, ..., M_{max}\}$.

Step 3: if ρ_{jd}^{t+1} approaches to zero we can discard feature d.

Step 4: Calculate the associated message length $MessLength(M)$ using Eq 12.

- Select the optimal model M^* such that $M^* = \arg\min MessLength(M)$ for $M^* \in \{M_{min}, ..., M_{max}\}$.

Note that in our model, we find the optimal number of clusters by running MML model concurrently for models $\{M_{min}, \ldots, M_{max}\}$ and select the solution which minimize message length. For computational complexity sake, when the number of components is large and the estimation of the candidate model is slow, we keep all the candidate model fitting with $\{\Theta_{M_{min}}, \ldots, \Theta_{M_{max}}\}$.

3.5 Experimental Results

In this section, we present the experimental results of applying proposed framework on high dimensional data extracted from challenging applications, namely, image-based spam filtering and web page categorization problem. The goal of the experiments is twofold. First, we investigate the impact of feature selection in improving the overall clustering performance. Second, we determine how online learning is desirable with dynamic data. In all experiments, we initialize the feature saliency values to $\rho_{jd} = 0.5$ as we assume that each feature has equal probability of being relevant or irrelevant.

Datasets

In our experiments we used publicly available datasets for both applications to evaluate the performance of our proposed framework.

- For spam filtering, we use Dredze[5] dataset [90] which contains emails from publicly well-known SpamArchive datasets along with many personal emails of the authors. In particular, it consists of 2021 ham images (HPer) and 3299 spam images (SPer) and a SpamArchive spam (SArc) set with 16035 images. After preprocessing, we remove

[5] To the best of our knowledge, due to the privacy issues, this is the only publicly available spam dataset that includes private legitimate images. Yet, many researchers have added their own private legitimate images which makes it harder to have a fair comparison [128].

images smaller than 10×10 and those cannot be recognized with image processer we end up having 1770 ham and 3112 spam images and 8719 spams from SpamArchive.

- For web categorization, we used Yahoo20[6] dataset which contains 2340 articles belonging to 20 categories: Business (142), Entertainment (9), art (24), cable (44), culture (74), film (278), industry (70), media (21), multimedia (14), music (125), online (65), people (248), review (158), stage (18), television (187), variety (54), Health (494), Politics (114), Sports (141), Technology(60).

Evaluation Criteria

For evaluation we used typical measures for spam filtering and web categorization. We reported the execution time of the batch framework on an Intel(R) Core(TM) 64 Processor PC with the Windows XP Service Pack 3 operating system and a 4 GB main memory. While in the online, we reported the average time to assign new document (image). Moreover, we calculated accuracy, micro-averaged F_1 and macro-averaged F_1 as follows:

$$\text{Accuracy} = \frac{\text{Number of documents correctly clustered}}{\text{Total number of documents}}$$

$$F_1(\text{micro-averaged}) = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$F_1(\text{macro-averaged}) = \frac{\sum_{j=1}^{M} F_j}{M}, \quad F_j = \frac{2 \times \text{Precision}_j \times \text{Recall}_j}{\text{Precision}_j + \text{Recall}_j}$$

where

$$\text{Precision} = \frac{\text{number of documents correctly predicted in class}i}{\text{number of documents in class}i}$$

$$\text{Recall} = \frac{\text{number of documents correctly predicted in class}i}{\text{number of correct prediction of class}i}$$

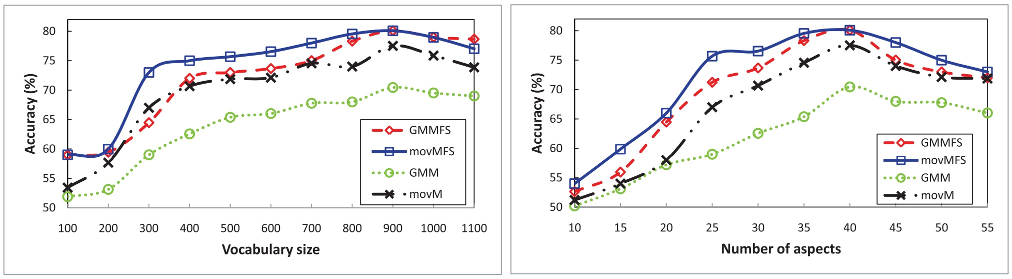
It is worth mentioning that larger values of $F_1 \in (0, 1)$ represent higher classification quality.

[6] fttp://fttp.cs.umn.edu/dept/users/boley

3.5.1 Filtering Using Bag of Visual Words Approach

Image-based spam email circumvents easily classic text based spam filters, thus some approaches have been proposed to detect the nature of email from its image content. Most of proposed approaches consider, however, only the textual content of the image and ignore its rich low-level visual content (e.g. color, texture, shape) which can be very helpful as clearly shown for instance in previous works about content-based image indexing and retrieval [89]. Moreover, spam images may not contain text (e.g. the picture of an object without text to advertise a website). Only few papers have considered the low level visual content of spam images as a solution to make filters more robust and smarter [90, 129]. Motivated by the recent success of local descriptors in computer vision applications, the authors in [130] proposed the modeling of images using the so called visual keywords (i.e. quantization of local descriptors) which are then classified as spam or ham using SVM. This approach has some merits since the local descriptor used (i.e. scale-invariant feature transform (SIFT)) is robust to several geometric transformations that may be used by spammers, but the quantization step applied can cause the loss of important information about the image content as shown in [131].

The goal of this first application is to investigate the impact of feature selection in clustering performance and comparing movM to GMM both with feature selection (movMFS, GMMFS) and without feature selection (movM, GMM). An important step in our application is the extraction of local features which well-describe given images. To this sake, we adopt the Bag-of-visual-words (BoVW) approach; thereby each image is represented by a single vector of frequencies. We start by detecting local regions on each image, using difference-of-Gaussian (DoG) detector, which we describe using their SIFT descriptors [85], giving 128 dimensional vector for each local region. Extracted vectors are clustered using the K-Means algorithm providing 900 visual-words vocabulary. We have tested several vocabulary sizes and the best classification results were obtained with 900 visual words, as illustrated in Fig. 3.1(a). Thus, each image in the dataset is then represented by a 900-dimensional vector describing the frequencies of a set of visual words, provided from



(a)

(b)

Figure 3.1: Classification accuracy for the spam data set as a function of (a) the vocabulary size, (b) the number of aspects.

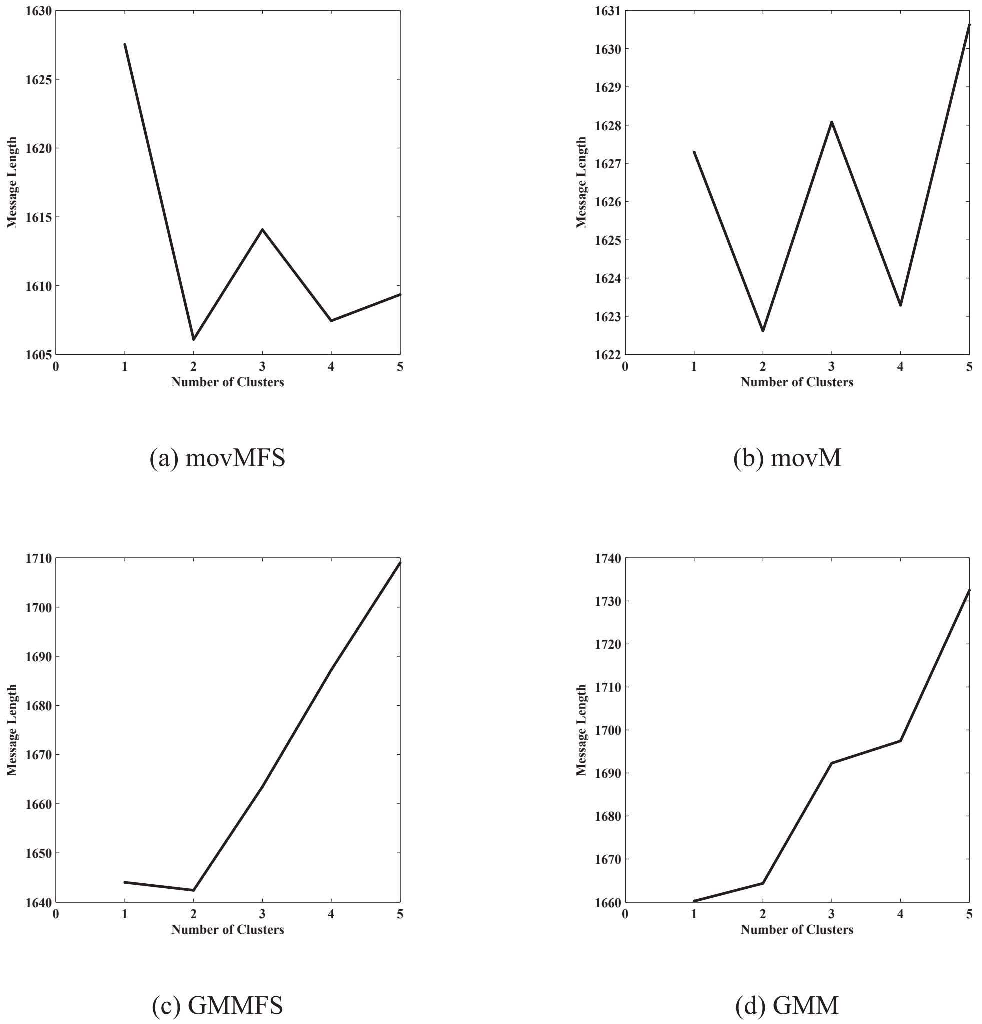
Table 3.1: Performance For Spam Filtering based on movM and GMM with and without feature selection.

	with Feature Selection		Without Feature Selection	
	movM	GMM	movM	GMM
Accuracy (%)	80.09	80.10	77.52	70.45
Run Time (sec)	24.15	31.34	40.26	47.05

the constructed visual vocabulary. Having these feature vectors, the Probabilistic Latent Semantic Analysis (pLSA) model is applied by considering 40 topics, for dimensionality reduction which has been shown to improve classification performance. Fig. 3.1(b) shows that the choice of the number of aspects has a real impact on the accuracy of filtering and the optimal accuracy obtained when the number of aspects is set to 40. After we prepared our dataset we randomly split dataset 10 times into training and testing sets, then we start spam filtering.

Figure 3.2 shows the number of components found by our proposed algorithm when adopting movM and GMM. Evaluation results are shown in Table 3.1. Note that in most cases we found the optimal number of mixture components. According to these results, it is clear that the presence of irrelevant features affects both the classification accuracy and estimation of model components. Moreover, using movM shows a slight improvement over GMM in the majority of the scenarios.

In our next experiments, we investigate the performance of the online framework with feature selection (Section 3.4) on the spam dataset. We first arranged images in chronological order. Then,



(a) movMFS

(b) movM

(c) GMMFS

(d) GMM

Figure 3.2: Number of clusters determined to represent spam and legitimate emails.

we select the oldest 1000 articles from both classes (i.e. spam ad legitimate) to initialize, where we clustered using the algorithm proposed in 3.3.2. Later, we used the online algorithm (Section 3.4) each time we insert new image until the end. We initialize the number of components to be $M_{min} = 1$ and $M_{max} = 4$ for all our experiments. It is worth mentioning that the choice of M_{min} and M_{max} is user defined and depends on the dataset at hand. In our experiments, after trying several values we find our choice a feasible one to achieve the optimal performance without

Table 3.2: Performance For Spam Filtering based on movM and GMM with feature selection in online settings.

	movM	GMM
Accuracy (%)	79.81	76.03
Run Time (sec)	1.06	1.17

compromising the model complexity.

Table 3.2 presents the results of spam filter after inserting the whole images. In terms of running time, in all the experiments, both movM and GMM show a quite similar speed. However, GMM shows worse performance in terms of accuracy. Moreover, we can observe that the online algorithm gives worse clustering performance than its batch counterpart, which is expected since the online algorithm can only update the cluster statistics incrementally.

It is worth mentioning that in this chapter we have considered generative models for feature selection and online learning. However, for spam filtering problem, many discriminative approaches have been used in the past. For instance, in our previous work [48, 132] we have studied the use of SVM in off-line and online settings to solve the problem of spam filtering for textual part of the email (i.e. we didn't use images). In one hand, feature selection is performed as a preprocessing step in the case of SVM and the computational cost increases linearly with the number of feature vectors [110, 133]. On the other hand, we argue that our proposed model presents a unified framework to simultaneously consider feature relevancy while gradually updating given model in unsupervised way. In addition, unlike discriminative models, we can further engage prior knowledge of application environment and expert impression where we can describe each image as a bag of vectors instead of one single vector as we will show in our next experiments which has shown to improve the quality of clustering. For further comparison interested user may refer to [134] and references therein.

3.5.2 Filtering Using Hybrid Generative/Discriminative Learning

Many researchers have paid attention to the complementary characteristics of generative and discriminative approaches and have attempted to merge the flexibility of generative approaches and the performance of discriminative approaches, and hence many procedures have been proposed. One common approach that has been suggested is "Fisher Kernel" [63] which has been shown to provide an elegant way to build hybrid models and has been widely used in different applications. Thus, we develop a hybrid framework that models image descriptors using movM from which Fisher kernel is generated for SVMs [40]. Thereby, we calculate the Fisher score of each component by finding the derivative of the log-likelihood of the sequence $\mathcal{X}$ with respect to particular parameter. Through the computation of gradient of the log likelihood with respect to our model parameters: p_j, κ_{jd}, $\vec{\mu}_{jd}$, ρ_{jd}, $\vec{\mu}_{jd|\lambda}$ and $\kappa_{jd|\lambda}$ where $j = 1, \dots, M$, we obtain

$$\frac{\partial \log p(\mathcal{X}|\Theta)}{\partial p_j} = \sum_{i=1}^{N} \Big[\frac{\hat{Z}_{ij}}{p_j} - \frac{\hat{Z}_{i1}}{p_1} \Big], \quad 1 < j \leq M \tag{30}$$

$$\frac{\partial \log p(\mathcal{X}|\Theta)}{\partial \mu_{jd}} = \frac{\sum_{i=1}^{N} \hat{Z}_{ij} \vec{Y}_{id} \kappa_{jd} \rho_{jd}}{\| \sum_{i=1}^{N} \hat{Z}_{ij} \vec{Y}_{id} \kappa_{jd} \rho_{jd} \|} \tag{31}$$

$$\frac{\partial \log p(\mathcal{X}|\Theta)}{\partial \kappa_{jd}} = \sum_{i=1}^{N} \hat{Z}_{ij} \rho_{jd} \Big[\vec{\mu}_{jd}^{T} \vec{Y}_{id} - A(\kappa_{jd}) \Big] \tag{32}$$

$$\frac{\partial \log p(\mathcal{X}|\Theta)}{\partial \rho_{jd}} = \sum_{i=1}^{N} \hat{Z}_{ij} \Big[\frac{p(\vec{Y}_{id}|\theta_{jd}) - p(\vec{Y}_{id}|\lambda_{jd})}{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})} \Big] \tag{33}$$

where $\hat{Z}_{ij}$ is the posterior we found previously in E-step. Similarly we can find the Fisher scores for $\vec{\mu}_{jd|\lambda}$ and $\kappa_{jd|\lambda}$. It is noteworthy that in Eq. 30, we take into account the fact that the sum of the mixing parameters equals one and thus there are only $M-1$ free mixing parameters. The main goal of this experiment is to evaluate the advantages of performing feature selection in hybrid generative discriminative framework by comparing: hybrid learning of movM and GMM with (HmovMFS,

Table 3.3: Performance For Spam Filtering based on hybrid framework for both models in different scenarios.

	with Feature Selection		Without Feature Selection	
	HmovM	HGMM	HmovM	HGMM
Accuracy (%)	91.43	88.82	86.45	84.78
Run time (sec)	22.01	31.41	49.3	50.42

HGMMFS) and without feature selection (HmovM, HGMM). Details about the learning of GMM can be found in [94]. The libsvm[7] software was used for SVMs classifier.

In this experiment, we replaced the visual words generation by fitting directly a given generative model (movM, GMM) to the local SIFT feature vectors, normalized using L_2 normalization, extracted from the images (i.e. each image is encoded as a bag of SIFT feature vectors). As a result each image (in both training and testing sets) was represented by a finite mixture model which can be viewed actually as the generative stage. Then, the Fisher scores between each of these mixture models were computed giving us kernel matrices to feed SVM classifier which represents our discriminative stage. A summary of the classification results obtained for the different classification tasks, is shown in Table 3.3. These results show that combining mixture models and SVMs outperforms classification using pure generative models only. Note that the best results were obtained when hybrid framework was applied using feature selection. The performance of hybrid framework is rather promising, comparing to Maximum Entropy (.91±.006), Naive Bayes (.80±.007) and an ID3 Decision Tree (.87±.020) in [90].

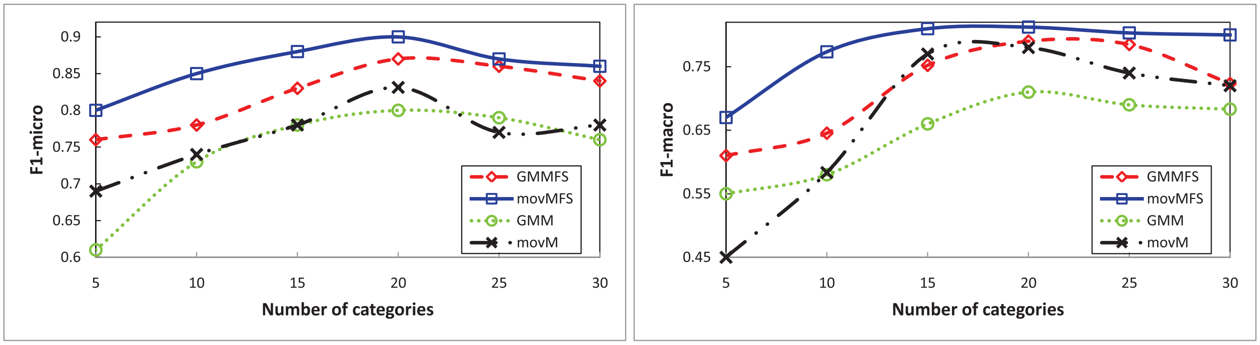
3.5.3 Online Web pages Clustering

The revolution of the Internet has made Web a popular place for individuals and organizations to share and collect information. Nevertheless, the dissemination of useful information on the Web in many cases has been accompanied by a large amount of noise that can seriously ruin automated

[7] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

information collection and mining on the Web such in Web pages clustering and information retrieval. Online web pages clustering goal is to automatically discover the clusters of similar web pages as long as they arrive in a stream of sequence [135]. This is particularly a challenging problem as one would need to deal with high-dimensional vectors sometimes tens of thousands of features, and as a result, the number of clusters can be considerably large [135]. Two clustering tasks were considered for this application. The first one is to study the impact of feature selection on web pages clustering in off-line settings using Algorithm 1 that we have proposed in Section 3.3.2. The second one is to explore the influence of feature selection in online settings using Algorithm 2 that we have proposed in Section 3.4.

We start by preparing our data by extracting the text of the news articles. Next, we applied stemming and removed words that occur less than 5 times and rare words while we kept stop words. This gives us a vocabulary of 21839 words, in the second step we presented each document as a vector of words frequencies that we normalize using L_2 normalization. Experimental results (see Figure 3.3) show that both movMFS and GMMFS with Feature selection outperform movM and GMM without feature selection. This is actually justified by the fact that most of the features that were assigned lower weight are pronouns, adverbs, etc which are generally in stop words list, that has been shown to affect clustering results. In all the experiments, all models achieve their higher value at M = 20 which is the correct number of categories. Looking closely to macro-averaged F_1 we can clearly see that clustering was influenced by the fact that the categories are imbalanced. Accordingly, feature selection influences web clustering in off-line settings. In order to explore if this is the case for online settings in our next experiment we learnt web articles incrementally. As we proved in previous experiment that feature selection improves the clustering performance, we decided to remove stop words and rare words and apply stemming before clustering. Each article is then described a L_2-normalized frequency vector. We initialize $M_{min} = 1$ and $M_{max} = 30$ for all our experiments. We randomly select 1000 articles out of 2340 to initialize and we insert the remaining articles during the running time of the algorithm. The number of clusters found each time we insert 286 images based on movM and GMM is given in Figures 3.5 and 3.6, respectively.



(a) micro-averaged F1

(b) macro-averaged F1

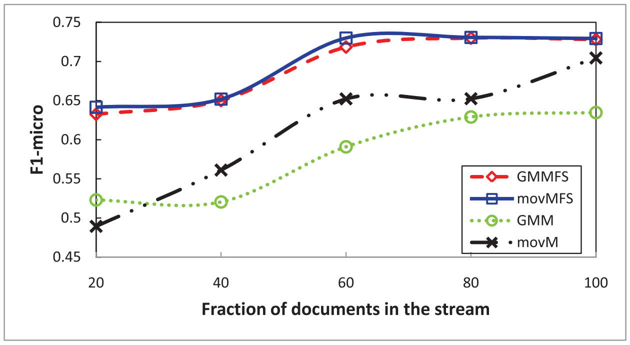
Figure 3.3: Micro-averaged F1 and macro-averaged F1 vs number of categories on Yahoo20 dataset with and without feature selection based on movM and GMM in off-line framework.

Table 3.4: Performance For Yahoo20 dataset in off-line and online settings for both movM and GMM using Feature selection.

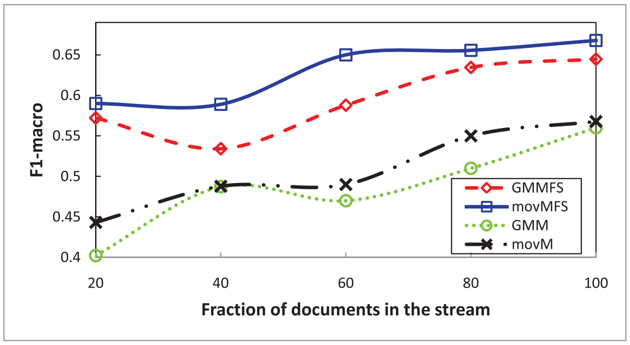
	Off-line		Online	
	movMFS	GMMFS	movMFS	GMMFS
Accuracy (%)	88.97	86.26	87.04	85.95
Run time (sec)	18.98	25.76	2.074	2.092

From these figures, we clearly observe that MML found the exact number of clusters (20 clusters) after the insertion of the remaining 1340 articles in case of movM. Fig. 3.4 shows the F1(micro-averaged) and F1(macro-averaged) over different number of documents fed to the system over time. Note that both movM and GMM achieve best F1(microaveraged) and F1(macro-averaged) when we insert the whole set of documents in both dataset. According to those figures we notice again that using feature selection as apart of online learning has improved the quality of the clusters.

Table 3.4 shows that the accuracy of online settings is comparable to its off-line counterpart when we insert the rest of articles.

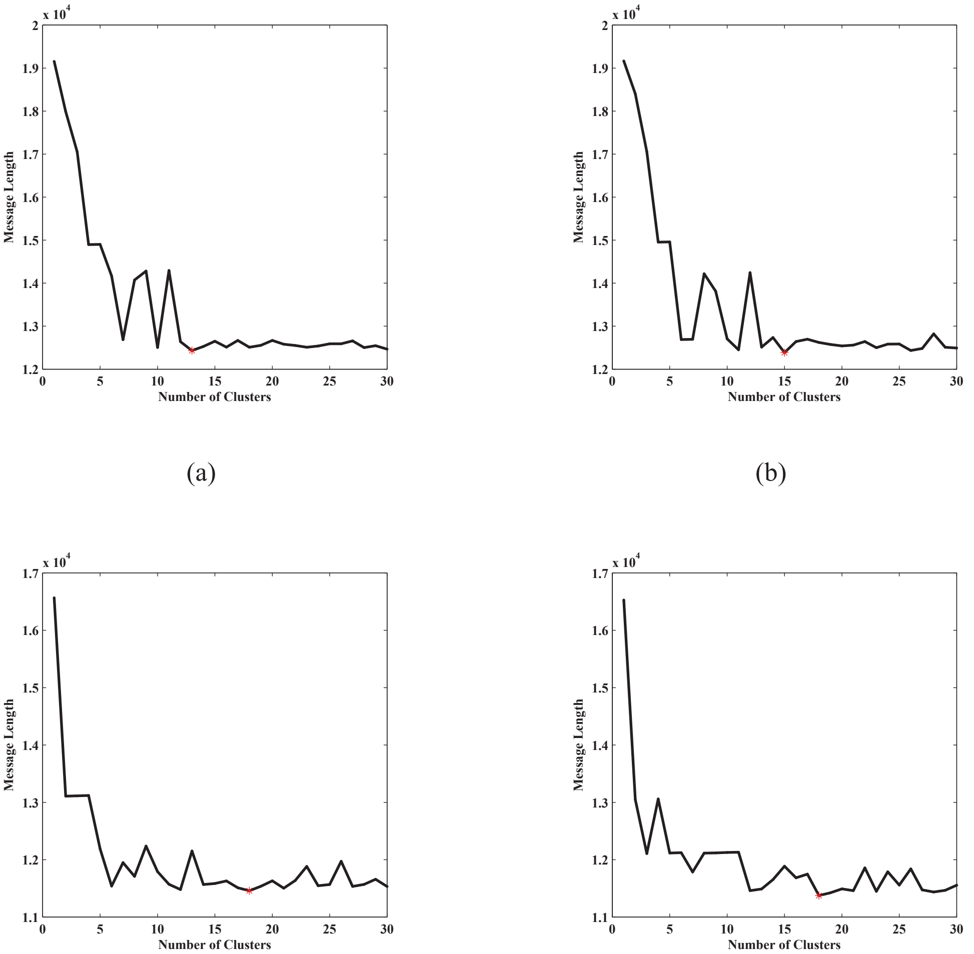


(a) micro-averaged F1



(b) macro-averaged F1

Figure 3.4: Micro-averaged F1 and macro-averaged F1 in Online Framework with and without feature selection based on movM and GMM.

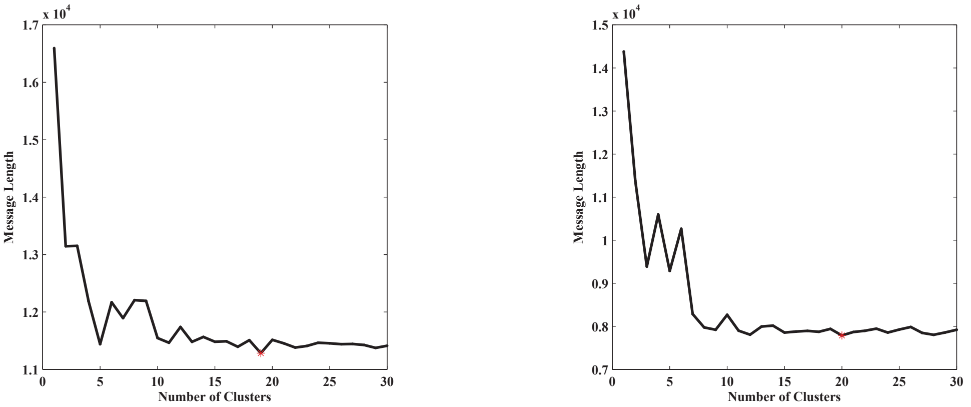

x 10^4
Message Length
Number of Clusters
0
5
10
15
20
25
30
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1

(a)

(b)

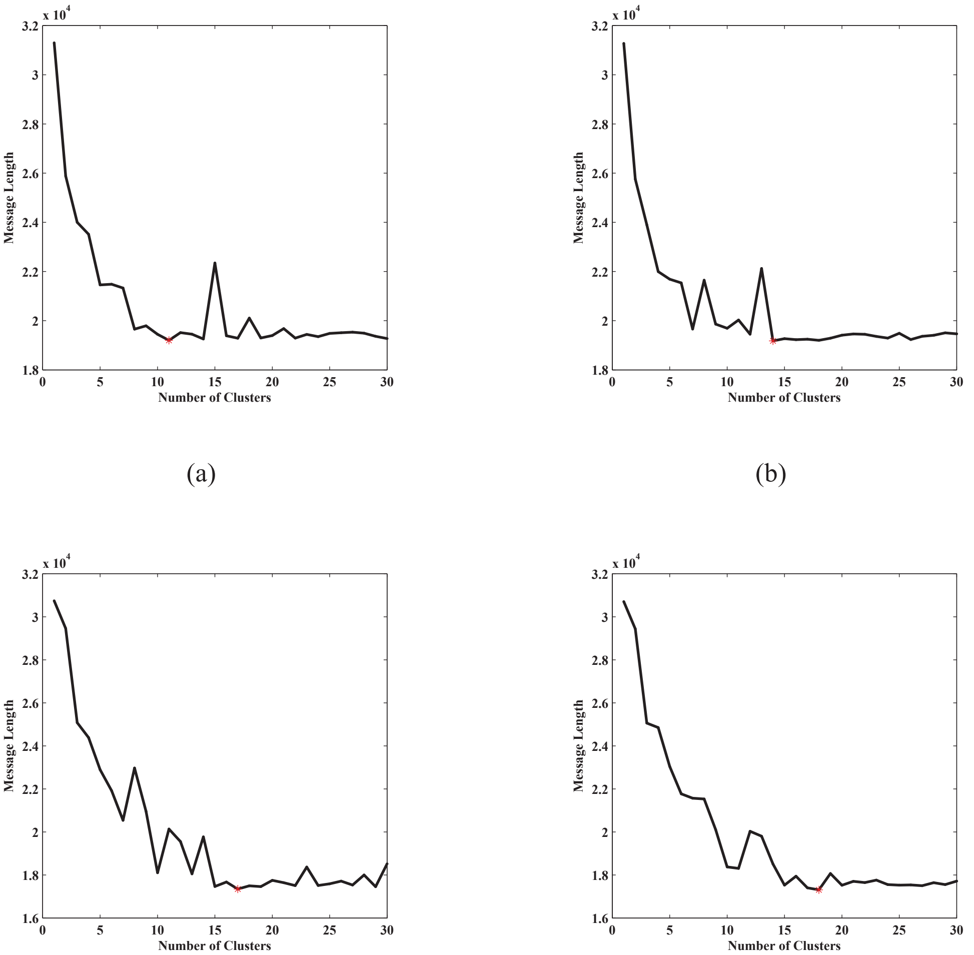
(c)

(d)



(e) (f)

Figure 3.5: Number of clusters determined to represent Yahoo20 categories based on online movM with feature selection (movMFS).

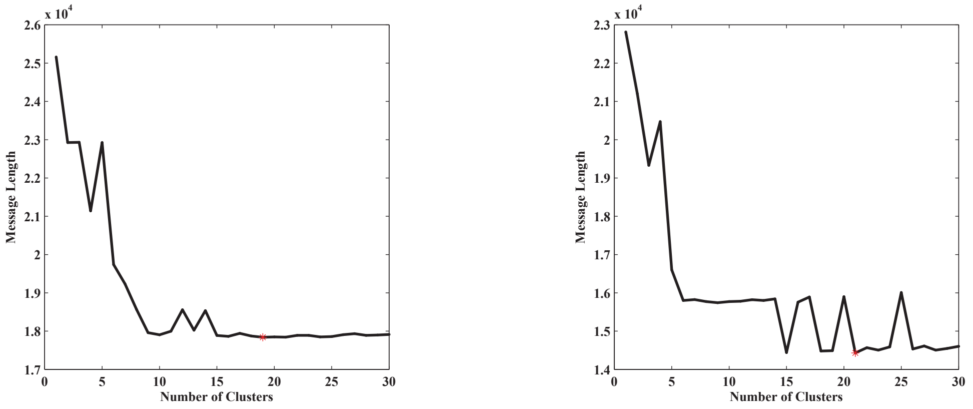

x 10^4
3.2
3
2.8
2.6
2.4
2.2
2
1.8
1.6
Message Length
0
5
10
15
20
25
30
Number of Clusters

(a)

(b)

(c)

(d)



(e)

(f)

Figure 3.6: Number of clusters determined to represent Yahoo20 categories based on online GMM with feature selection (GMMFS).

CHAPTER 4

A Bayesian Clustering and Feature Selection for Spherical Data

In this Chapter, we propose a new parameter estimation methods based on Bayesian inference. In particular, we first propose a pure Bayesian algorithm for the estimation and selection of Langevin mixture model. To extend Bayesian framework to consider feature selection, we propose a framework that combines clustering and feature selection based on movM learned using RJMCMC approach. Experimental results in vital and challenging problems, namely topic detection and tracking and image categorization, are presented.

4.1 Introduction

In this chapter, we shall consider finite Langevin mixtures. A key step in mixture-based modeling of data is parameter estimation. Many methods have been proposed in the literature in order to estimate mixture parameters, including frequentist (a.k.a deterministic) and Bayesian approaches [32]. In this chapter, we focus in developing parameter estimation and model selection from Bayesian perspective. We are mainly motivated by the fact that Bayesian learning has several desirable properties that make it widely used in several applications. For instance, it does not suffer over-fitting and prior knowledge is incorporated naturally in a principled way. In this chapter we shall not motivate further Bayesian learning which has been widely discussed in the past (interested reader may refer to [136–138] for further details and interesting discussions).

Rooted in the early work of [139] Bayesian inference for the Von Mises Fisher (vMF) distribution (3-dimensional case of the Langevin distribution) was proposed. This work was based on the development of a conjugate prior for the mean (Jeffreys prior was also developed for the polar coordinates) when concentration parameter is known. In the area of radio signals, authors in [140] applied Bayesian approach for finding the location of an emergency transmitter signal based on the von Mises (vM) distribution (2-dimensional case of the Langevin distribution) by developing conjugate priors using the canonical parameterizations. A Gibbs sampler for vM distribution was introduced in [141] by developing conjugate priors for the polar coordinates. In [142] authors provide a full Bayesian analysis of directional data using the vMF distribution, again using standard conjugate priors and obtaining samples from the posterior using a sampling-importance-resampling method. Compared to these methods, our work is not restricted to low dimensional data (i.e. Von Mises (2D) or Von Mises Fisher (3D)) which is a limited solution for many real-world problems. On contrary, we extend previous models to high dimensional data using Langevin mixture (for D>3) where both the concentration and mean parameters are unknown. In particular, we propose a Markov Chain Monte Carlo (MCMC) algorithm that relies on Gibbs sampler and Metropolis-Hastings (M-H) for the estimation of the parameters. To this end, we develop a conjugate prior for the Langevin distribution taking into account the fact that it belongs to the exponential family. As well as considering the estimation over model parameters, we also wish to consider the optimal number of components that best describe data at hand. One common approach is integrated likelihood [143] which we shall adopt for Langevin mixture in this chapter. Note that, despite various efforts to use Bayesian inference to learn mixtures [134, 144], to the best of our knowledge, none of the recent works has considered the case where the feature vectors to model are spherical so far.

Due to its high computational cost, these Bayesian models most often have be disregarded especially when one considers the necessity of feature selection which ironically provides superior performance. This motivates the need to find unified framework that combines the efficient models with feature selection. Subsequently, we propose an ambitious framework that simultaneously learn spherical clusters and identify relevant features. The learning of proposed framework is carried out using RJMCMC to estimate developed posterior distributions.

4.2 Bayesian Estimation

As we previously discussed EM provides an elegant and simple way to estimate the parameters of a given model, yet, EM algorithm is sensitive to the initialization and generally converges to local solution in the best case. To avoid this problem, an alternative way is to use Bayesian estimation for Langevin mixture model.

Bayesian estimation is based on finding the conditional distribution $p(\Theta|\mathcal{X}, \mathcal{Z})$ of parameters vector Θ which is brought by complete data $(\mathcal{X}, \mathcal{Z})$, where $\mathcal{Z} = \{\vec{Z}_1, \ldots, \vec{Z}_N\}$. We therefore select a prior distribution $p(\Theta)$ and then develop posterior distribution $p(\Theta|\mathcal{X}, \mathcal{Z})$ which is derived from the joint distribution $p(\mathcal{Z}, \Theta, \mathcal{X})$ via Bayes formula $p(\Theta|\mathcal{X}, \mathcal{Z}) \propto p(\mathcal{Z}, \Theta, \mathcal{X})$. The joint distribution of all variables can be written as:

$$p(\Theta|\mathcal{X}, \mathcal{Z}) = p(\vec{\theta}, \vec{P}|\mathcal{X}, \mathcal{Z}) \propto p(\vec{P})p(\vec{\theta})p(\mathcal{Z}|\vec{P}) \prod_{Z_{ij=1}} p(\vec{X}_i|\theta_j) \tag{1}$$

where $p(\vec{\theta})$ and $p(\vec{P})$ are the priors of θ and $\vec{P}$ which we will describe in what follows.

Priors and Posteriors

In order to derive our Bayesian algorithm we now turn to defining our priors over the parameters. Langevin distribution is a member of (curved)-exponential family of order D, whose shape is symmetric and unimodal. Thus, we can write it as the following [145]:

$$p(\vec{X}|\theta) = H(\vec{X}) \exp(G(\theta)^T T(\vec{X}) + \Phi(\theta)) \tag{2}$$

where $G(\theta) = (G_1(\theta), \dots, G_l(\theta))$, $T(\vec{X}) = (T_1(\vec{X}), \dots, T_l(\vec{X}))$ where l is the number of parameters of the distribution and tr denotes transpose. The conjugate prior [1] on θ, in this case, can be written as [138]:

$$p(\theta) \propto \exp(\sum_{l=1}^{S} \rho_l G_l(\theta) + \lambda \Phi(\theta)) \tag{3}$$

where $\rho = (\rho_1, \dots, \rho_S) \in \mathbb{R}^S$ and $\lambda > 0$ are referred as hyperparameters. To this end, Langevin distribution can be written as follows:

$$p_D(\vec{X}|\vec{\mu}, \kappa) = \exp\{\kappa \vec{\mu}^T \vec{X} - a_D(\kappa)\} \tag{4}$$

where $a_D(\kappa) = -\log\Big\{\dfrac{\kappa^{\frac{D}{2}-1}}{(2\pi)^{\frac{D}{2}} I_{\frac{D}{2}-1}(\kappa)}\Big\}$. Then, by letting $\Phi_\theta = -a_D(\kappa)$ and $G_\theta = \kappa\vec{\mu}$, the prior can be written as:

$$p(\theta) \propto (\exp(\sum_{d=1}^{D} \kappa_0 \vec{\mu}_0^T \vec{\mu}_j - \lambda a_D(\kappa_0))) \tag{5}$$

The prior hyperparameters are: $(\kappa_0, \vec{\mu}_0, \lambda)$, where $\vec{\mu}_0 \in S^D$ is the mean of the observations, κ_0 is the concentration parameter and λ is a non-negative integer. Having the prior at hand, the posterior is given as following:

$$\begin{aligned} p(\theta_j|\mathcal{Z}, \mathcal{X}) \propto & p(\theta_j) \prod_{Z_{ij}=1} p(\vec{X}_i|\theta_j) \propto \Big[\frac{\kappa_0^{\frac{D}{2}-1}}{(2\pi)^{\frac{D}{2}} I_{\frac{D}{2}-1}(\kappa_0)}\Big]^{\lambda} \exp(\kappa_0 \vec{\mu}_0^t \vec{\mu}_j) \\ & \times \Big[\frac{\kappa_j^{\frac{D}{2}-1}}{(2\pi)^{\frac{D}{2}} I_{\frac{D}{2}-1}(\kappa_j)}\Big]^{n_j} \exp(n_j \kappa_j \vec{\mu}_j^t \prod_{Z_{ij}=1} \vec{X}_i) \\ & \propto (a_D(\kappa_j))^{\lambda+n_j} \exp(R_j \beta_j^T \vec{\mu}_j) \end{aligned} \tag{6}$$

[1] [146] contains an interesting discussion about the characteristics of conjugate priors and their induced posteriors in Bayesian inference for von Mises Fisher distributions, using either the canonical natural exponential family or the more commonly employed polar coordinate parameterizations.

where

$$R_j = \parallel \kappa_0 \vec{\mu}_0 + n_j \kappa_j \prod_{Z_{ij}=1} \vec{X}_i \parallel \tag{7}$$

$$\beta_j = \frac{\kappa_0 \vec{\mu}_0 + n_j \kappa_j \prod_{Z_{ij}=1} \vec{X}_i}{R_j}$$

Next, we develop distribution $p(\vec{P}|\mathcal{Z})$ and according to Eq. 1, we have:

$$p(\vec{P}|\mathcal{Z}) \propto p(\vec{P}) p(\mathcal{Z}|\vec{P}) \tag{8}$$

We know that the vector $\vec{P}$ is defined on the simplex $\{(p_1, \ldots, p_M) : \sum_{j=1}^{M-1} p_j < 1\}$, and hence a natural choice, as a prior, for this vector is a Dirichlet distribution with parameters $\eta = (\eta_1, \ldots, \eta_M)$. Then,

$$p(\vec{P}|\mathcal{Z}) = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} p_j^{\eta_j - 1} \prod_{j=1}^{M} p_j^{\eta_j} \propto \mathcal{D}(\eta_1 + n_1, \ldots, \eta_M + n_M) \tag{9}$$

where $\mathcal{D}$ is a Dirichlet distribution with parameters $(\eta_1 + n_1, \ldots, \eta_M + n_M)$ and

$$p(\mathcal{Z}|\vec{P}) = \prod_{i=1}^{N} p(\vec{Z}_i|\vec{P}) = \prod_{i=1}^{N} p_1^{Z_{i1}} \ldots p_M^{Z_{iM}} = \prod_{i=1}^{N} \prod_{j=1}^{M} p_j^{Z_{ij}} = \prod_{j=1}^{M} p_j^{n_j} \tag{10}$$

where $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_{ij}=j}$. One of the most common approaches to conduct Bayesian inference is Gibbs sampler [137], which we adopt in this chapter. The standard Gibbs sampler for mixture models is based on the successive simulations of $\vec{\theta}$, Z, and $\vec{P}$ and is given as follows [137]:

Algorithm 1:

1. Initialization

2. Step t: For $t = 1, \ldots$

    - Generate $\vec{Z}_i^{(t)} \sim \mathcal{M}(1; \hat{Z}_{i1}^{(t-1)}, \ldots, \hat{Z}_{iM}^{(t-1)})$
    - Compute $n_j^{(t)} = \sum_{i=1}^{N} \mathbb{I}_{Z_{ij}^{(t)}=j}$
    - Generate $\vec{P}^{(t)}$ from Eq. 9.

- Generate $\theta_j^{(t)}$, $j = 1, \dots, M$, from Eq. 6 using M-H algorithm [147].

where $\mathcal{M}(1; \hat{Z}_{i1}^{(t-1)}, \dots, \hat{Z}_{iM}^{(t-1)})$ denotes a multinomial of order one with parameters $(\hat{Z}_{i1}^{(t-1)}, \dots, \hat{Z}_{iM}^{(t-1)})$, where $\hat{Z}_{ij}$ is calculated according to Eq. 1.

The major problem in the M-H algorithm is the need to choose a proposal distribution. In order to tackle this problem, random walk Metropolis−Hastings algorithm is the most generic proposal where each unconstrained parameter is the mean of the proposal distribution for the new value. In our case, the distributions of parameters are considered as: Langevin $\tilde{\mu}_j \sim \mathcal{LM}(\mu_j^{t-1}|\tilde{\mu}_j, \kappa_j)$ for the mean μ, where μ_j is a constant unit vector, κ_j is concentration parameter. As we know the concentration parameter is non-negative integer $\kappa > 0$, we consider log-normal $\tilde{\kappa}_j \sim \mathcal{LN}(\log(\kappa_j^{t-1}), \sigma^2)$ for the concentration parameter κ with mean $\log(\kappa_j^{t-1})$ and variance σ^2. With these proposals[2] at hand the random walk M−H algorithm is given by:

M-H Algorithm:

- Generate $\tilde{\mu}_j \sim \mathcal{LM}(\mu_j^{t-1}|\tilde{\mu}_j, \kappa_j)$, $\tilde{\kappa}_j \sim \mathcal{LN}(\log(\kappa_j^{t-1}), \sigma^2)$ and $\mathbf{u} \sim \mathcal{U}_{[0,1]}$
- Compute $r = \frac{p(\tilde{\theta}_j|\mathcal{Z},\mathcal{X}) \prod_{d=1}^{D} \mathcal{LM}(\mu_j^{t-1}|\tilde{\mu}_j,\kappa_j)\mathcal{LN}(\kappa_j^{t-1}|\log(\tilde{\kappa}_j),\sigma^2)}{p(\theta_j^{t-1}|\mathcal{Z},\mathcal{X}) \prod_{d=1}^{D} \mathcal{LM}(\tilde{\mu}_j|\mu_j^{t-1},\kappa_j)\mathcal{LN}(\tilde{\kappa}_j|\log(\kappa_j^{t-1}),\sigma^2)}$
- if $r < u$ then $\theta_j^t = \tilde{\theta}_j$ else $\theta_j^t = \theta_j^{t-1}$.

Model Selection

This algorithm requires some further enhancement. Indeed, it is crucial to find also the optimal number of components. In this chapter we adopt integrated likelihood to determine the number of clusters M defined by [143]:

$$p(\mathcal{X}|M) = \int p(\Theta|\mathcal{X}, M)d\Theta = \int p(\mathcal{X}|\Theta, M)p(\Theta|M)d\Theta \tag{11}$$

where $p(\mathcal{X}|\Theta, M)$ is the likelihood function of finite mixture model taking into account the number of clusters, which is M in this case, Θ is the vector of parameters and $p(\Theta|M)$ is prior density.

[2] It is worth mentioning that other proposals are possible. For instance, in [142] authors chose Gamma distribution as a proposal for concentration parameter κ. But our experiments have proven better results with log-normal distribution selected as proposal.

This integral is not analytically tractable and is generally computed via Laplace approximation, on the logarithm scale, as follows [143]:

$$\log p(\mathcal{X}|M) = \log p(\mathcal{X}|\hat{\Theta}, M) + \log p(\hat{\Theta}|M) + \frac{N_p}{2}\log(2\pi) + \frac{1}{2}\log|H(\hat{\Theta})| \qquad (12)$$

where $|H(\hat{\Theta})|$ is the determinant of the Hessian matrix and $N_p = M(D+1) - 1$ is the number of parameters in the model. A simple and accurate solution to estimate $\hat{\Theta}$ is to choose Θ in the sample at which $p(\mathcal{X}|\hat{\Theta}, M)$ achieves its maximum [148]. Moreover, $H(\hat{\Theta})$ is asymptotically equal to the posterior variance matrix, and hence, we could estimate it by the sample covariance matrix of the posterior simulation output [148]. Thus, the complete Bayesian algorithm of finite Langevin mixture is as follows:

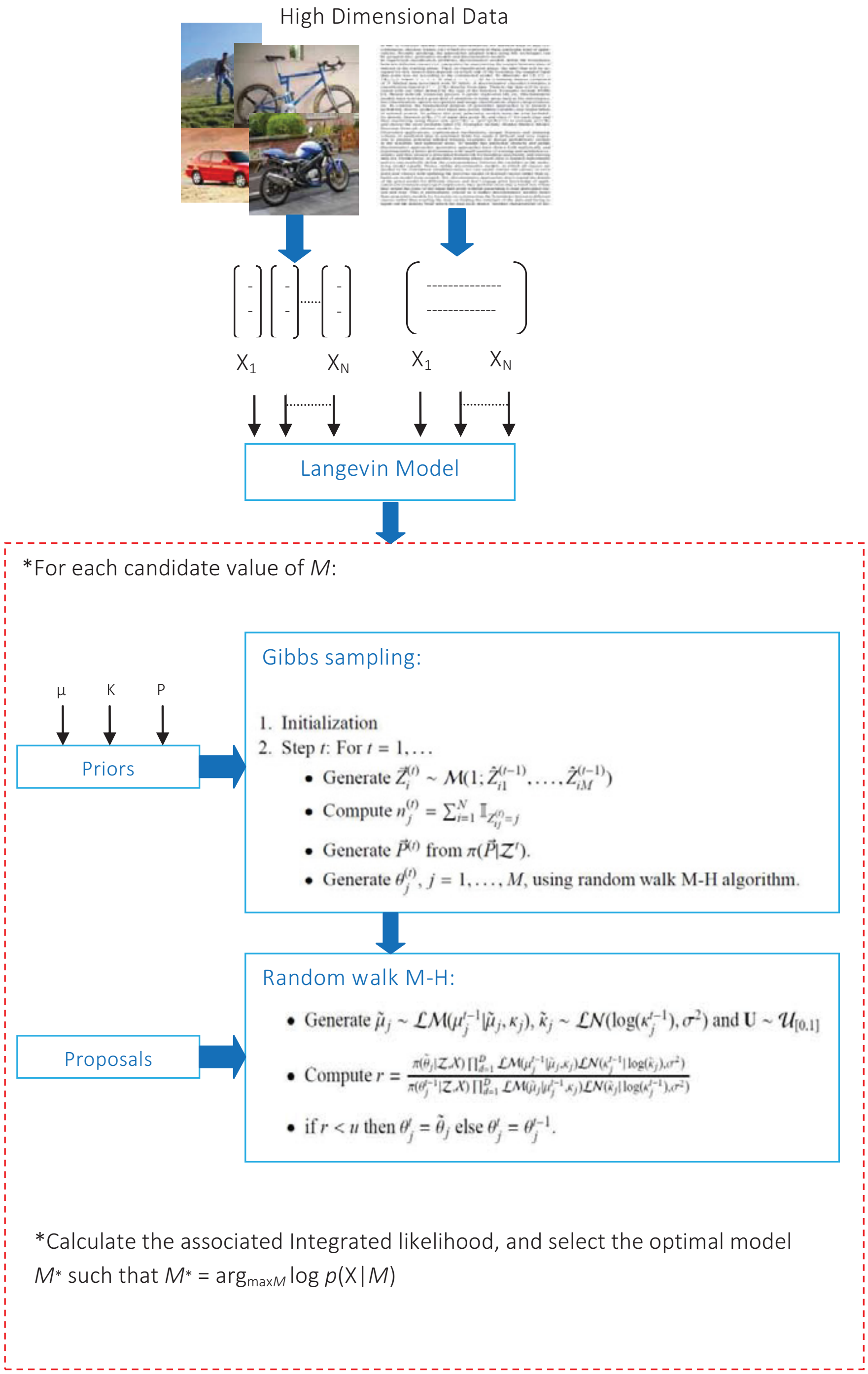
Algorithm 2:

1. Apply Algorithm 1
2. Select the optimal model M^* such that $M^* = \arg\max_M \log p(\mathcal{X}|M)$

One common argument when using MCMC is the convergence of the model. Many techniques have been proposed in the past to determine the convergence of the model, see for instance [149], however the discussion of those methods is beyond the scope of this chapter. Thus, in our chapter we used a diagnostic approach based on a single long run of the Gibbs sampler, proposed in [150] which has been shown to be sufficient for high dimensional data [134]. See Figure 4.1 for proposed learning model.

4.3 Mixture Density and Feature Selection

Given a dataset $\mathcal{X} = \{\vec{X}_1, \ldots, \vec{X}_N\}$ of N documents (or images). Therein, each document (image) $\vec{X}_i = (X_{i1}, \ldots, X_{iD})$ is described by a L_2-normalized D-dimensional feature vector, such that $(\vec{X}_i)^T\vec{X}_i = 1$. Each document is best modeled using vM distribution. In particular, we shall



*For each candidate value of M:

Gibbs sampling:

1. Initialization
2. Step t: For $t = 1, \dots$
   - Generate $\vec{Z}_i^{(t)} \sim \mathcal{M}(1; \hat{Z}_{i1}^{(t-1)}, \dots, \hat{Z}_{iM}^{(t-1)})$
   - Compute $n_j^{(t)} = \sum_{i=1}^{N} \mathbb{I}_{Z_{ij}^{(t)}=j}$
   - Generate $\vec{P}^{(t)}$ from $\pi(\vec{P}|\mathcal{Z}^t)$.
   - Generate $\theta_j^{(t)}$, $j = 1, \dots, M$, using random walk M-H algorithm.

Random walk M-H:

- Generate $\tilde{\mu}_j \sim \mathcal{LM}(\mu_j^{t-1}|\tilde{\mu}_j, \kappa_j)$, $\tilde{\kappa}_j \sim \mathcal{LN}(\log(\kappa_j^{t-1}), \sigma^2)$ and $\mathbf{U} \sim \mathcal{U}_{[0,1]}$
- Compute $r = \frac{\pi(\tilde{\theta}_j|\mathcal{Z},\mathcal{X}) \prod_{d=1}^{D} \mathcal{LM}(\mu_j^{t-1}|\tilde{\mu}_j,\kappa_j)\mathcal{LN}(\kappa_j^{t-1}|\log(\tilde{\kappa}_j),\sigma^2)}{\pi(\theta_j^{t-1}|\mathcal{Z},\mathcal{X}) \prod_{d=1}^{D} \mathcal{LM}(\tilde{\mu}_j|\mu_j^{t-1},\kappa_j)\mathcal{LN}(\tilde{\kappa}_j|\log(\kappa_j^{t-1}),\sigma^2)}$
- if $r < u$ then $\theta_j^t = \tilde{\theta}_j$ else $\theta_j^t = \theta_j^{t-1}$.

*Calculate the associated Integrated likelihood, and select the optimal model M^* such that $M^* = \text{arg}_{\text{max}M} \log p(X|M)$

Figure 4.1: Proposed Bayesian framework.

suppose that the features in each vector $\vec{X}_i$ are independent and follows a vM distribution which gives the following:

$$p(\vec{X}_i|\theta_j) = \prod_{d=1}^{D} p(Y_{id}|\theta_{jd}) = \prod_{d=1}^{D} \frac{1}{2\pi I_0(\kappa_{jd})} \exp\{\kappa_{jd}\mu_{jd}Y_{id}\} \tag{13}$$

where I_0 is the modified Bessel function of the first kind and order zero [24], $\theta_j = (\vec{\mu}_j, \kappa_j)$, $\theta_{jd} = (\mu_{jd}, \kappa_{jd})$, $\vec{\mu}_j = (\mu_{j1}, \ldots, \mu_{jD})$ is the mean direction, $\kappa = (\kappa_{j1}, \ldots, \kappa_{jD})$ is the concentration parameter and D is the number of features. let $p(\vec{X}_i|\Theta_M)$ be a mixture of M distributions represented by Eq. 1. The probability density function of a M-components movM is given by

$$p(\vec{X}_i|\Theta_M) = \sum_{j=1}^{M} p_j \prod_{d=1}^{D} p(Y_{id}|\theta_{jd}) \tag{14}$$

where $\Theta_M = \{\vec{P} = (p_1, \ldots, p_M), \theta_{jd}\}$ denotes all the parameters of the mixture model, p_j represents the weight of the j^{th} movM component and $\vec{P}$ is the vector of mixing parameters that are positive and sum to one.

Each d^{th} feature is represented by movM of two components $p(Y_{id}|\theta_{jd})$ and $p(Y_{id}|\theta_{jd}^{irr}))$ governed by ρ_{jd} that denotes the weight of the d^{th} feature on cluster j. In fact, if ρ_{jd} is very high, then there is no significant difference with the classical movM model $p(Y_{id}|\theta_{jd})$ without any saliency. Thus, our model, to take feature selection into account, can be written as:

$$p(\vec{X}_i|\Theta) = \sum_{j=1}^{M} p_j \prod_{d=1}^{D} (\rho_{jd} p(Y_{id}|\theta_{jd}) + (1 - \rho_{jd}) p(Y_{id}|\theta_{jd}^{irr})) \tag{15}$$

where $\Theta = \{\Theta_M, \{\rho_{jd}\}, \{\theta_{jd}^{irr}\}\}$, $\theta_{jd}^{irr} = (\vec{\mu}_{jd}^{irr}, \kappa_{jd}^{irr})$ are the parameters of vM from which the irrelevant feature is drawn.

4.3.1 Bayesian Learning using RJMCMC

Hierarchical model, Priors and Posteriors

In contrast to classical Bayesian model, which we proposed previously in Section 4.2, that relies on other model selection approaches (see Section 4.2) in order to find the optimal number of components, in this section we shall find the optimal number of components simultaneously with the estimation process. Thus, we shall adopt RJMCMC [151] approach, in which the number of components M is regarded as a model's parameter drawn from prior distributions. The joint distribution of our model is given by:

$$\begin{aligned} p(M, \vec{P}, Z, z, \vec{\rho}, \vec{\theta}, \vec{\theta}^{irr}, \mathcal{X}) = p(M)p(\mathcal{X}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}, M)p(\vec{\theta}^{irr}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, M) \\ \times p(\vec{\theta}|\vec{P}, Z, \vec{\rho}, z, M)p(\vec{P})p(Z|\vec{P}, M)p(\vec{\rho}|\vec{P}, Z, M) \quad (16) \\ \times p(z|\vec{\rho}, \vec{P}, Z, M) \quad (17) \end{aligned}$$

where $Z = (Z_1, \ldots, Z_N)$ denotes the missing allocation variables, such that Z_i shows the cluster that vector $\vec{X}_i$ was generated from, $z = (z_1, \ldots, z_N)$, such that $z_i = (\vec{z}_{i1}, \ldots, \vec{z}_{iM})$, where $\vec{z}_{ij} = (z_{ij1}, \ldots, z_{ijD})$ are the missing binary vectors that indicate if a given feature Y_{id} is relevant or not. Indeed, a proiri probability that the feature Y_{id} is relevant for component j is given by:

$$p(z_{ijd} = 1, Z_i = j|\vec{X}_i) = \frac{\rho_{jd}p(Y_{id}|\theta_{jd})}{\rho_{jd}p(Y_{id}|\theta_{jd}) + (1 - \rho_{jd})p(Y_{id}|\theta_{jd}^{irr})}p(Z_i = j|\vec{X}_i)$$

$$p(z_{ijd} = 0, Z_i = j|\vec{X}_i) = \frac{(1 - \rho_{jd})p(Y_{id}|\theta_{jd})}{\rho_{jd}p(Y_{id}|\theta_{jd}) + (1 - \rho_{jd})p(Y_{id}|\theta_{jd}^{irr})}p(Z_i = j|\vec{X}_i)$$

Following [151] we can impose some conditional independencies, such that:

$$\begin{aligned} &p(\vec{\theta}|\vec{P}, Z, \vec{\rho}, z, M) = p(\vec{\theta}|M), \quad p(\vec{\theta}^{irr}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, M) = p(\vec{\theta}^{irr}|M) \\ &p(\vec{\rho}|\vec{P}, Z, M) = p(\vec{\rho}|M), \quad p(z|\vec{\rho}, \vec{P}, Z, M) = p(z|\vec{\rho}), \\ &p(\mathcal{X}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}, M) = p(\mathcal{X}|Z, z, \vec{\theta}, \vec{\theta}^{irr}) = \prod_{i=1}^{N}\prod_{d=1}^{D}[(p(Y_{id}|\theta_{Z_{id}}))^{z_{id}}(p(Y_{id}|\theta_{Z_{id}}^{irr}))^{1-z_{id}}] \end{aligned}$$

which give us the following joint distribution:

$$\begin{aligned} p(\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}, \mathcal{X}, M) = p(\vec{P})p(Z|\vec{P})p(\vec{\rho}|M)p(z|\vec{\rho})p(\vec{\theta}|M)p(\vec{\theta}^{irr}|M) \\ \times \prod_{i=1}^{N}\prod_{d=1}^{D}[(p(Y_{id}|\theta_{Z_{id}}))^{z_{id}}(p(Y_{id}|\theta_{Z_{id}}^{irr}))^{1-z_{id}}] \quad (18) \end{aligned}$$

Furthermore, an extra layer can be introduced to the hierarchy in order to add more flexibility and hence we suppose that the model parameters ($\vec{\theta}$, $\vec{\theta}^{irr}$, $\vec{\rho}$, $\vec{P}$,M) follows priors depending on hyperparamters (Λ,Λ^{irr}, ξ, η,δ). Such that $\Lambda = (\Lambda_1, \ldots, \Lambda_M)$, $\Lambda_j = (\Lambda_{j|\mu}, \Lambda_{j|\kappa})$, and the same for the irrelevant model, where $\Lambda = (\Lambda_1^{irr}, \ldots, \Lambda_M^{irr})$ and $\Lambda_j = (\Lambda_{j|\mu}^{irr}, \Lambda_{j|\kappa}^{irr})$. Thus,

$$p(\vec{\theta}|\Lambda) = \prod_{j=1}^{M} p(\mu_j, \kappa_j|\Lambda_{j|\theta_j}) \quad p(\vec{\theta}^{irr}|\Lambda^{irr}) = \prod_{j=1}^{M} p(\mu_j^{irr}, \kappa_j^{irr}|\Lambda_{j|\theta_j^{irr}}^{irr})$$

Hence the joint distribution of our model is given by:

$$\begin{aligned} p(\vec{P}, Z, z, \vec{\rho}, \Lambda, \Lambda^{irr}, \eta, \xi, \delta, \theta, \theta^{irr}, \mathcal{X}) &= p(\Lambda)p(\Lambda^{irr})p(\xi)p(\eta)p(\delta) \\ &\times p(\vec{P}|\eta)p(Z|\vec{P})p(\vec{\rho}|\xi, M)p(z|\vec{\rho})p(\mathcal{X}|\vec{\theta}, \vec{\theta}^{irr}, Z, z) \\ &\times \prod_{j=1}^{M} \Big[p(\vec{\mu}_j, \kappa_j|\Lambda_{j|\theta_j})p(\vec{\mu}_j^{irr}, \kappa_j^{irr}|\Lambda_{j|\theta_j^{irr}}^{irr}) \Big] \end{aligned} \tag{19}$$

Now will define our priors, where we suppose that vM parameters μ, κ and μ^{irr}, κ^{irr} are drawn independently from the rest of parameters in our hierarchical model. Thus, our conjugate prior is given by:

$$p(\vec{\mu}_j, \kappa_j|\kappa_0, \mu_0) \propto (\exp(\sum_{d=1}^{D} \kappa_0 \mu_0 \vec{\mu}_j - \lambda a_2(\kappa_0))) \tag{20}$$

The prior hyperparameters are: $(\kappa_0, \mu_0, \lambda)$, where $\mu_0 \in S^D$ is the mean of the observations, κ_0 is the concentration parameter and λ is a non-negative integer. And $p(\vec{\mu}_j^{irr}, \kappa_j^{irr}|\kappa_0, \mu_0)$ has the same form as $p(\vec{\mu}_j, \kappa_j|\kappa_0, \mu_0)$. Using the above equation we have:

$$p(\vec{\theta}|M, \tau) = \prod_{j=1}^{M} p(\vec{\mu}_j, \kappa_j|\kappa_0, \mu_0) \tag{21}$$

Thus, the generic hyperparameters $\Lambda_{j|\theta_j}$ and $\Lambda_{j|\theta_j^{irr}}^{irr}$ become $\tau = (\kappa_0, \mu_0, \lambda)$ and hence the conditional posterior distributions for θ_j and θ_j^{irr}, giving the rest of the parameters, are:

$$p(\theta_j| \ldots) \propto p(\vec{\mu}_j, \kappa_j|\kappa_0, \mu_0)p(\mathcal{X}|\vec{\theta}, \vec{\theta}^{irr}, Z, z) \tag{22}$$

$$p(\theta_j^{irr}|\ldots) \propto p(\vec{\mu}_j^{irr}, \kappa_j^{irr}|\kappa_0, \mu_0)p(\mathcal{X}|\vec{\theta}, \vec{\theta}^{irr}, Z, z) \tag{23}$$

The hyperparameters μ_0 and κ_0 are given Von Mises and Gamma priors, respectively:

$$p(\mu_0|\mu_1, \kappa_1) = \prod_{d=1}^{D} \frac{1}{2\pi I_0(\kappa_1)} \exp\{\kappa_1\mu_1\mu_0\} \quad p(\kappa_0|a, b) = \prod_{d=1}^{D} \frac{\kappa_0^{a-1} b^a \exp(-b\kappa_0)}{\Gamma(a)} \tag{24}$$

Thus, according to Eqs. 19, 20 and 24 we obtain the following posteriors:

$$p(\mu_0|\ldots) \propto p(\mu_0|\mu_1, \kappa_1) \prod_{j=1}^{M} p(\vec{\mu}_j, \kappa_j|\kappa_0, \mu_0)p(\vec{\mu}_j^{irr}, \kappa_j^{irr}|\kappa_0, \mu_0) \tag{25}$$

$$p(\kappa_0|\ldots) \propto p(\kappa_0|a, b) \prod_{j=1}^{M} p(\vec{\mu}_j, \kappa_j|\kappa_0, \mu_0)p(\vec{\mu}_j^{irr}, \kappa_j^{irr}|\kappa_0, \mu_0) \tag{26}$$

We know that the vector $\vec{P}$ is defined on the simplex $\{(p_1, \ldots, p_M) : \sum_{j=1}^{M-1} p_j < 1\}$, and hence a natural choice, as a prior, for this vector is a Dirichlet distribution with parameters $\eta = (\eta_1, \ldots, \eta_M)$. Then,

$$p(\vec{P}|\eta, M) = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} p_j^{\eta_j - 1} \prod_{j=1}^{M} p_j^{\eta_j} \propto \mathcal{D}(\eta_1 + n_1, \ldots, \eta_M + n_M) \tag{27}$$

where $\mathcal{D}$ is a Dirichlet distribution with parameters $(\eta_1 + n_1, \ldots, \eta_M + n_M)$ and

$$p(\vec{P}|\ldots) \propto p(Z|P, \vec{M})p(\vec{P}|M, \eta) \propto \prod_{j=1}^{M} p_j^{n_j + \eta_j - 1} \tag{28}$$

where $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_{ij}=j}$. Now the posterior of the membership variables can be given by:

$$p(Z_i = j|\vec{X}_i) = \frac{p_j \prod_{d=1}^{D} \Big(\rho_{jd} p(Y_{id}|\theta_{jd}) + (1 - \rho_{jd}) p(Y_{id}|\theta_{jd}^{irr})\Big)}{\sum_{j=1}^{M} p_j \prod_{d=1}^{D} \Big(\rho_{jd} p(Y_{id}|\theta_{jd}) + (1 - \rho_{jd}) p(Y_{id}|\theta_{jd}^{irr})\Big)} \tag{29}$$

is the probability that vector i is in cluster j, conditional on having observing $\vec{X}_i$. For M we take as a prior a common choice which is uniform distribution $\{1, \ldots, \sigma\}$, where σ is a constant representing the maximum value allowed for M. As ρ_{jd} is defined in the compact support [0,1], we consider Beta distribution with parameters ϵ_1 and ϵ_2, common to all classes and dimensions, as prior:

$$p(\vec{\rho}|\epsilon) = \Big[\frac{\Gamma(\epsilon_1+\epsilon_2)}{\Gamma(\epsilon_1)\Gamma(\epsilon_2)}\Big]^{MD} \prod_{j=1}^{M}\prod_{d=1}^{D} \rho_{jd}^{\epsilon_1-1}(1-\rho_{jd})^{\epsilon_2-1} \tag{30}$$

Hence, the generic hyperparameter ϵ becomes (ϵ_1,ϵ_2). Recall that $\rho_{jd} = p(z_{jd} = 1)$ and $1 - \rho_{jd} = p(z_{jd} = 0)$, $d = 1, \dots, D$, $j = 1, \dots, M$ thus each z_{jd} follows a D-variate Bernoulli distribution and we have

$$p(z|\vec{\rho}) = \prod_{i=1}^{N}\prod_{j=1}^{M}\prod_{d=1}^{D} \rho_{jd}^{z_{ijd}}(1-\rho_{jd})^{1-z_{ijd}} = \prod_{j=1}^{M}\prod_{d=1}^{D} \rho_{jd}^{f_{jd}}(1-\rho_{jd})^{N-f_{jd}} \tag{31}$$

where $f_{jd} = \sum_{i=1}^{N} \mathbb{I}_{z_{ijd}=1}$. Then, according to Eqs. 19 30 31, we have

$$p(\vec{\rho}|\dots) \propto p(\vec{\rho}|\epsilon)p(z|\vec{\rho}) \propto \prod_{j=1}^{M}\prod_{d=1}^{D} \rho_{jd}^{f_{jd}+\epsilon_1-1}(1-\rho_{jd})^{N-f_{jd}+\epsilon_2-1} \tag{32}$$

Then, we suppose that the hyperparameters ϵ_1 and ϵ_2 are given Gamma priors with common hyperparameters (ε_ϵ, ϱ_ϵ) which gives us the following posteriors:

$$p(\epsilon_1|\dots) = p(\epsilon_1|\varepsilon_\epsilon, \varrho_\epsilon)p(\vec{\rho}|\epsilon) \quad p(\epsilon_2|\dots) = p(\epsilon_2|\varepsilon_\epsilon, \varrho_\epsilon)p(\vec{\rho}|\epsilon) \tag{33}$$

Model Learning using RJMCMC

Gibbs sampling moves

As for the first move, we start by updating the mixing parameters $\vec{P}$ which generated from Eq. 28. The second move is based on Update the model parameters $\vec{\mu}_j$, κ_j, $\vec{\mu}_j^{irr}$ and κ_j^{irr} from Eqs. 22 and 23. However, the conditional posteriors of model parameters do not have known forms. Thus, we have used the random walk MH algorithm with Langevin model proposal for $\vec{\mu}_j$ and $\vec{\mu}_j^{irr}$ and log-normal proposals for κ_j and κ_j^{irr} (See previous sections). In the next move we update the allocation (missing data) Z_i, $i = 1, \dots, N$ from standard uniform random variables r_n, where

$Z_{ij} = 1$ if $(p(Z_{i1} = 1|\ldots) + \ldots, p(Z_{ij-1} = 1|\ldots)) < r_n \leq (p(Z_{i1} = 1|\ldots) + \ldots, p(Z_{ij} = 1|\ldots))$ (see Eq. 29). In the fourth move we update the hyperparameters μ_0, κ_0, ϵ_1, ϵ_2 and η using Eqs. 25, 26 and 36. As we can clearly see those equations are hard to sample from, thus we use ARS [152].

Split and merge moves

In split and merge move, we have the choice to choose between splitting a given component or merging two components with probabilities g_M and k_M, respectively, where $g_\sigma = 0$ and $k_1 = 1$ and $g_M = k_M = 0.5$, otherwise. Following [151] we need to preserve the first two moments before and after the combine and split moves. In our case,the merging proposal works as follows: choose two components j_1 and j_2, where $\vec{\mu}_{j1} < \vec{\mu}_{j2}$ with no other $\vec{\mu}_j \in [\vec{\mu}_{j1}, \vec{\mu}_{j2}]$ (i.e adjacency condition). If these components are merged, we reduce M by 1, which forms a new components $j*$ containing all the observation previously allocated to j_1 and j_2 and then creates values for p_*, $\vec{\mu}_{j*}$ and κ_{j*} by preserving the first two moments, as follows:

$$p_{j*} = p_{j_1} + p_{j_2} \tag{34}$$

$$\rho_{j*} = 1 - \rho_{j_2}, \quad \rho_{j_1} = \rho_{j*} \tag{35}$$

$$\mu_{dj*} = \frac{p_{j_1}\mu_{dj_1} + p_{j_2}\mu_{dj_2}}{p_{j*}}, \quad d = 1, \ldots, D \tag{36}$$

$$\kappa_{dj*} = \frac{p_{j_1}(\mu_{dj_1}^2 + \kappa_{dj_1}) + p_{j_2}(\mu_{dj_2}^2 + \kappa_{dj_2})}{p_{j*}} - \mu_{dj*}^2 \tag{37}$$

When the decision is to split, we choose a component $j*$ at random to define two new components j_1 and j_2 having weights and parameters $(p_{j1}, \vec{\mu}_{j1}, \kappa_{j1}, \rho_{j1})$ and $(p_{j2}, \vec{\mu}_{j2}, \kappa_{j2}, \rho_{j2})$, respectively. Clearly, resolving these equations is an ill-posed problem for which we adopt the same solution as in:

$$p_{j_1} = u_1 p_{j*}, \quad p_{j_2} = (1 - u_1)p_{j*} \tag{38}$$

$$\mu_{dj_1} = \mu_{dj*} - u_2\sqrt{\kappa_{dj*}\frac{p_{j_2}}{p_{j_1}}} \tag{39}$$

$$\mu_{dj_2} = \mu_{dj*} + u_2\sqrt{\kappa_{dj*}\frac{p_{j_1}}{p_{j_2}}} \tag{40}$$

$$\kappa_{dj_1} = u_3(1-u_2^2)\kappa_{dj*}\frac{p_{j*}}{p_{j_1}} \tag{41}$$

$$\kappa_{dj_2} = (1-u_3)(1-u_2^2)\kappa_{dj*}\frac{p_{j*}}{p_{j_2}} \tag{42}$$

$$\rho_{j_1} = u_4\rho_{j*}, \quad \rho_{j_2} = (1-u_4)\rho_{j*} \tag{43}$$

where u_1, u_2, u_3 and u_4 are drawn from Beta distributions with parameters (2,2), (2,2), (1,1) and (1,1), respectively [151]. Then, we assign the different Y_{id} previously in $j*$ to j_1 or j_2 using Eq. 29. Next, we calculate the acceptance probabilities of split and combine moves: $\min\{1; R\}$ and $\min\{1; R^{-1}\}$, where according to $R = \frac{p(s'|\mathcal{X})r_m(s')}{p(s|\mathcal{X})r_m(s)q(u)}\left|\frac{\partial s'}{\partial(s,u)}\right|$, we have the following:

$$R = \frac{p(Z, \vec{P}, M+1, \vec{\theta}, \vec{\theta}^{irr}, \vec{\rho}, \Lambda, \Lambda^{irr}, z, \eta|\mathcal{X})k_{M+1}}{p(Z, \vec{P}, M, \vec{\theta}, \vec{\theta}^{irr}, \vec{\rho}, \Lambda, \Lambda^{irr}, z, \eta|\mathcal{X})g_M P_{alloc} q(u)}\left|\frac{\partial s'}{\partial(s,u)}\right| \tag{44}$$

where

$$P_{alloc} = \prod_{Z_i=j_1}\frac{\rho_{j_1}p_{j_1}p(X_i|\theta_{j_1})}{\rho_{j_1}p_{j_1}p(X_i|\theta_{j_1}) + \rho_{j_2}p_{j_2}p(X_i|\theta_{j_2})}\prod_{Z_i=j_2}\frac{\rho_{j_2}p_{j_2}p(X_i|\theta_{j_2})}{\rho_{j_1}p_{j_1}p(X_i|\theta_{j_1}) + \rho_{j_2}p_{j_2}p(X_i|\theta_{j_2})} \tag{45}$$

$$q(u) = p(u_1)p(u_2)p(u_3)p(u_4) \tag{46}$$

$$\left|\frac{\partial s'}{\partial(s,u)}\right| = \left|\frac{\partial(p_{j_1}, p_{j_2}, \vec{\mu}_{j_1}, \vec{\mu}_{j_2}, \kappa_{j_1}, \kappa_{j_2}, \vec{\rho_{j_1}}, \vec{\rho_{j_2}})}{\partial(p_{j*}, \vec{\rho_{j*}}, \kappa_{j*}, \vec{\mu}_{j*}, u_1, u_2, u_3, u_4)}\right| = \rho_{j*}p_{j*}\prod_{d=1}^{D}\frac{(\mu_{dj_2}-\mu_{dj_1})\kappa_{dj_1}\kappa_{dj_2}}{u_2(1-u_2^2)(1-u_3)\kappa_{dj*}} \tag{47}$$

Knowing that:

$$\frac{p(Z, \vec{P}, M+1, \vec{\theta}, \vec{\theta}^{irr}, \vec{\rho}, \Lambda, \Lambda^{irr}, z, \eta|\mathcal{X})}{p(Z, \vec{P}, M, \vec{\theta}, \vec{\theta}^{irr}, \vec{\rho}, \Lambda, \Lambda^{irr}, z, \eta|\mathcal{X})} = \text{(likelihood ratio)}(M+1)\frac{p(M+1)}{p(M)} \times \frac{p(\vec{P}|M+1,\eta)p(Z|\vec{P}, M+1)p(\vec{\theta}|M+1,\tau)p(\vec{\theta}^{irr}|M+1,\tau)}{p(\vec{P}|M,\eta)p(Z|\vec{P}, M)p(\vec{\theta}|M,\tau)p(\vec{\theta}^{irr}|M,\tau)} \tag{48}$$

where

$$\frac{p(\vec{P}|M+1,\eta)}{p(\vec{P}|M,\eta)} = \frac{\frac{\Gamma(\sum_{j=1}^{M+1}\eta_j)}{\sum_{j=1}^{M+1}\eta_j}\prod_{j=1}^{M+1} p_j^{\eta_j-1}}{\frac{\Gamma(\sum_{j=1}^{M}\eta_j)}{\sum_{j=1}^{M}\eta_j}\prod_{j=1}^{M} p_j^{\eta_j-1}}, \quad \frac{p(Z|\vec{P},M+1)}{p(Z|\vec{P},M)} = \frac{p_{j_1}^{n_{j_1}} p_{j_2}^{n_{j_2}} \prod_{j=1}^{M-1} p_j^{n_j}}{p_{j*}^{n_{j*}}\prod_{j=1}^{M-1} p_j^{n_j}}$$

where n_{j_1} and n_{j_2} are the number of observations to be assigned to j_1 and j_2 components, respectively, τ is the set of the hyperparameters.

Birth and death moves

In birth and death moves, we start by making a random choice between birth and death with probabilities g_M and k_M as above. For a birth, the parameters of the new component proposed a redrawn from the associated prior distributions. The weight of the new component, p_{j*}, is generated from the marginal distribution of p_{j*} derived from the distribution of $\vec{P} = (p_1, \ldots, p_M, p_{j*})$. The vector $\vec{P}$ follows a Dirichlet with parameters $(\eta_1, \ldots, \eta_M, \eta_{j*})$, thus the marginal of p_{j*} is a Beta distribution with parameters $\eta_{j*} \sum_{j=1}^{M} \eta_j$. Note that in order to keep the mixture constraint $\sum_{j=1}^{M} p_j + p_{j*} = 1$, the previous weights p_j, $j = 1, \ldots, M$ have to be rescaled and then all multiplied by $(1 - p_{j*})$. The Jacobian corresponding to the birth move is then $(1 - p_{j*})^M$. For the death move, we choose randomly an existing empty component to delete, then of course the remaining weights have to be rescaled to keep the unit-sum constraint. The acceptance probabilities of birth and death moves: $\min\{1; R\}$ and $\min\{1; R^{-1}\}$, are calculated according as the following

$$\begin{aligned} R = {} & \frac{p(M+1)}{p(M)} \frac{\Gamma(\eta_{j*} + \sum_{j=1}^{M}\eta_j)}{\Gamma(\eta_{j*})\Gamma(\sum_{j=1}^{M}\eta_j)} (M+1) p_{j*}^{\eta_{j*}-1} (1 - p_{j*})^{N + \sum_{j=1}^{M}\eta_j - M} \\ & \times \frac{k_{M+1}}{g_M(M_0+1)p(p_{j*})} (1 - p_{j*})^M \end{aligned} \tag{49}$$

where M_0 is the number of empty components before the birth.

4.4 Experimental Results

Experiments were conducted to assess the performance of the proposed framework by comparing it to other techniques mentioned previously in the literature. To achieve this goal, we used synthetic data and two challenging problems which are: Topic Detection and Tracking (TDT) and image categorization. In what follows, we used 5000 iteration in all for our algorithm where we discarded the first 500 iterations as burn-in. As for hyperparameters $(\mu_0, \kappa_0, \sigma^2, \lambda, \eta_1, \ldots, \eta_M) = (0, 1, 0.01, 0.12, 1, \ldots, 1)$ we have conducted a sensitivity test that showed the impact of the hyperparameters on the results. Thus, following [136] the Dirichlet parameters $(\eta_1, \ldots, \eta_M)$ are set to 1 as a common choice in the Bayesian case. As for σ^2 we set it to 0.01 as it has previously shown to increase the sensitivity of random walk sampler [134, 144]. Note that we tested different values for μ_0, λ in the range of [0,1] and the differences between results were not statistically significant. The remainder of this section is organized as follows. First, we present our experiments using generated data. Subsequently, we apply our approach on two challenging applications namely topic detection and tracking, and content-based image categorization.

4.4.1 Synthetic Data

We used several synthetic data to illustrate the performance of proposed framework. The first goal of this part is to compare Bayesian and EM learning of Langevin mixture model. In particular, we tested the performance of the two approaches for estimating the mixture's parameters and selecting the number of clusters by generating different datasets using different parameters. The results that we present were averaged over 20 runs. The real and estimated parameters of the generated datasets are given in Table 4.1. Figure 4.2 shows the time series plot of our Bayesian algorithm iterations for the first dataset. Figure 4.3 displays the number of clusters determined for generated datasets when using both Bayesian and EM approaches. Accordingly, we can clearly see that Bayesian approach provides more accurate estimates of the mixture parameters as compared to the EM. Moreover, Figure 4.3 represents the number of clusters found by both algorithms. It clearly shows that the

Table 4.1: True and estimated parameters for synthetic data generated from LMM. N and j denote the total number of elements and component number in each dataset. μ_j, κ_j and p_j are the true parameters. $\hat{\mu}_j^B$, $\hat{\kappa}_j^B$ and $\hat{p}_j^B$ are estimated parameters using Bayesian approach. $\hat{\mu}_j^{EM}$, $\hat{\kappa}_j^{EM}$ and $\hat{p}_j^{EM}$ are estimated parameters using the EM algorithm.

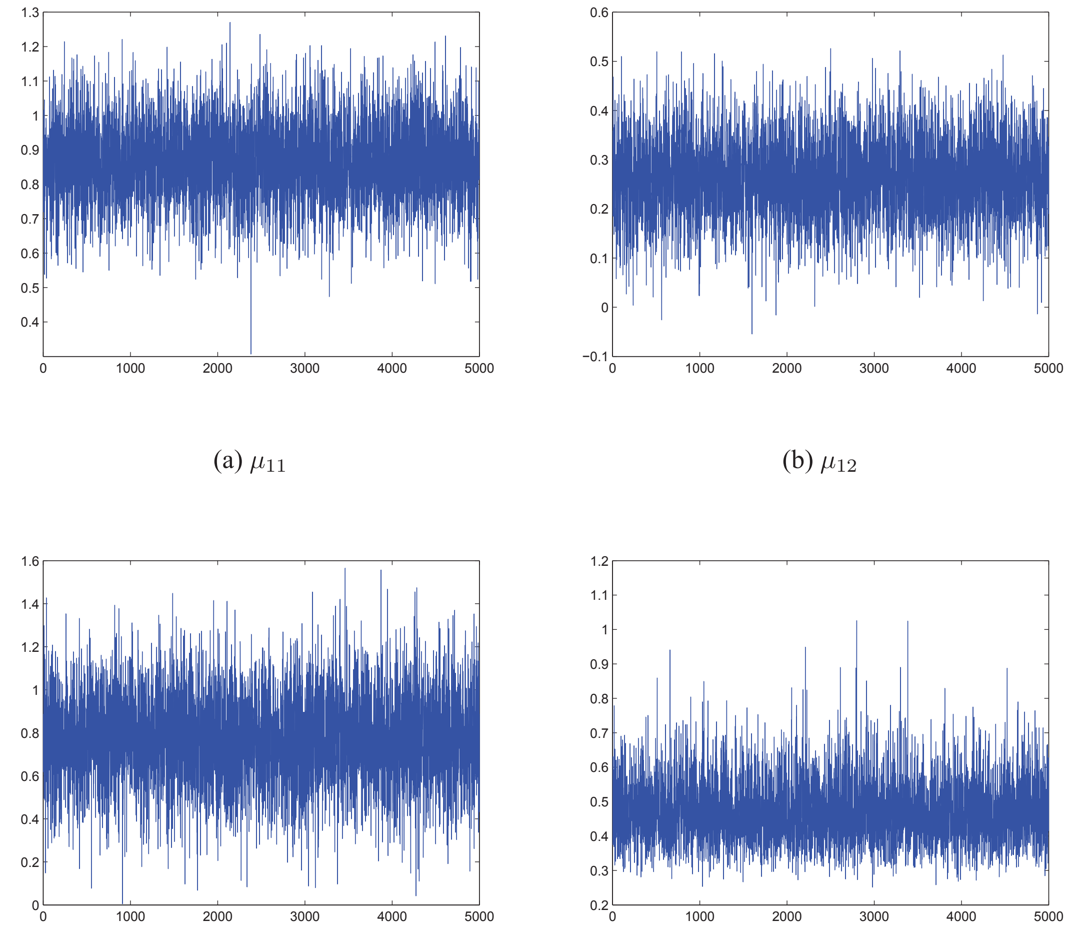
	j	μ_j	κ_j	p_j	$\hat{\mu}_j^B$	$\hat{\kappa}_j^B$	$\hat{p}_j^B$	$\hat{\mu}_j^{EM}$	$\hat{\kappa}_j^{EM}$	$\hat{p}_j^{EM}$
Dataset 1	1	(0.9547,0.2976)	10	0.5	(0.9500,0.2916)	10	0.5	(0.9499,0.2917)	9.97	0.5
(N =1000)	2	(0.8570,0.5153)	10	0.5	(0.8490,0.5150)	9.99	0.5	(0.8489,0.5149)	10	0.5
Dataset 2	1	(0.1997,0.0189,-0.3685)	10	0.3	(0.1996,0.0167,-0.3604)	10	0.3	(0.1901,0.0180,-0.3597)	8.79	0.34
(N =10000)	2	(-0.1588,-0.3477,-0.1244)	40	0.3	(-0.1534,-0.3397,-0.1213)	38	0.29	(-0.1510,-0.3400,-0.1200)	34.00	0.28
	3	(0.1011,-0.0485,0.0471)	30	0.2	(0.1010,-0.0476,0.0430)	29	0.21	(0.1001,-0.0474,0.0431)	30.40	0.22
	4	(0.1242,0.3738,-0.0043)	10	0.2	(0.1239,0.3721,-0.0041)	10	0.2	(0.1240,0.3721,-0.0040)	9.60	0.16
Dataset 3	1	(0.1997,0.0189,-0.3685,0.9077)	100	0.2	(0.1999,0.0180,-0.3605,0.9097)	97	0.18	(0.1900,0.0192,-0.3680,0.9060)	90	0.24
(N =15000)	2	(-0.1588,-0.3477,-0.1244,0.9156)	20	0.2	(-0.1590,-0.3475,-0.1254,0.9120)	19.9	0.19	(-0.1580,-0.3479,-0.1200,0.9100)	15.67	0.22
	3	(0.1011,-0.0485,0.0471,0.9926)	20	0.2	(0.1005,-0.04931,0.0470,0.9931)	21.54	0.22	(0.0987,-0.0480,0.0380,0.9901)	23.1	0.29
	4	(0.1242,0.3738,-0.0043,0.9191)	10	0.2	(0.1266,0.3711,-0.0040,0.9096)	9.80	0.21	(0.1120,0.3650,-0.0039,0.9020)	8.2	0.15
	5	(-0.2645,0.1520,0.0227,0.9521)	100	0.2	(-0.2590,0.1518,0.0221,0.9500)	99.10	0.2	(-0.2599,0.1460,0.0200,0.9502)	89.05	0.10
Dataset 4	1	(0.1997,0.0189,-0.3685,0.9077)	23	0.2	(0.2001,0.0190,-0.3680,0.9072)	21.57	0.19	(0.1990,0.0196,-0.3675,0.9057)	19.02	0.28
(N =20000)	2	(-0.1588,-0.3477,-0.1244,0.9156)	27	0.2	(-0.1570,-0.3479,-0.1243,0.9160)	27.88	0.22	(-0.1597,-0.3502,-0.1220,0.9140)	28.59	0.22
	3	(0.1011,-0.0485,0.0471,0.9926)	15	0.2	(0.1020,-0.0489,0.0460,0.9921)	14.98	0.28	(0.1005,-0.0474,0.0479,0.9929)	15.88	0.17
	4	(0.1242,0.3738,-0.0043,0.9191)	75	0.2	(0.1240,0.3740,-0.0037,0.9199)	75.01	0.23	(0.1333,0.3701,-0.0025,0.9200)	70.09	0.18
	5	(-0.2645,0.1520,0.0227,0.9521)	90	0.1	(-0.2649,0.1534,0.0217,0.9518)	87.25	0.13	(-0.2658,0.1566,0.0210,0.9546)	90.64	0.15
	6	(-0.0526,-0.1399,0.5361,0.8308)	100	0.1	(-0.0524,-0.1402,0.5359,0.8300)	110.01	0.11	-	-	-

correct number of clusters has been favored for all datasets using Bayesian algorithm, while EM failed to select the correct number of clusters for the last dataset.

4.4.2 Image Categorization

Image categorization problem is classical problem in computer vision and essential prerequisite for many important applications such as face and car detection, and video surveillance [153, 154]. Considerable attention on this topic in the past decades has produced diverse and rich collection of algorithms [154–157]. Many of these are based on exploiting contextual information [158]. However, the main goal of this section is not the comparison of all these approaches, which is actually beyond the scope of this chapter, but the investigation of our Bayesian when applied to image categorization.

The experiments were conducted on PASCAL Visual Object Classes (VOC) 2005 challenge dataset [159] that contains high variation objects and consists of 2232 images grouped into four categories: car, motorbike, people and bicycle. Figure 4.4 shows examples of images from all categories. An important step in our application is the extraction of local features to describe the visual objects.



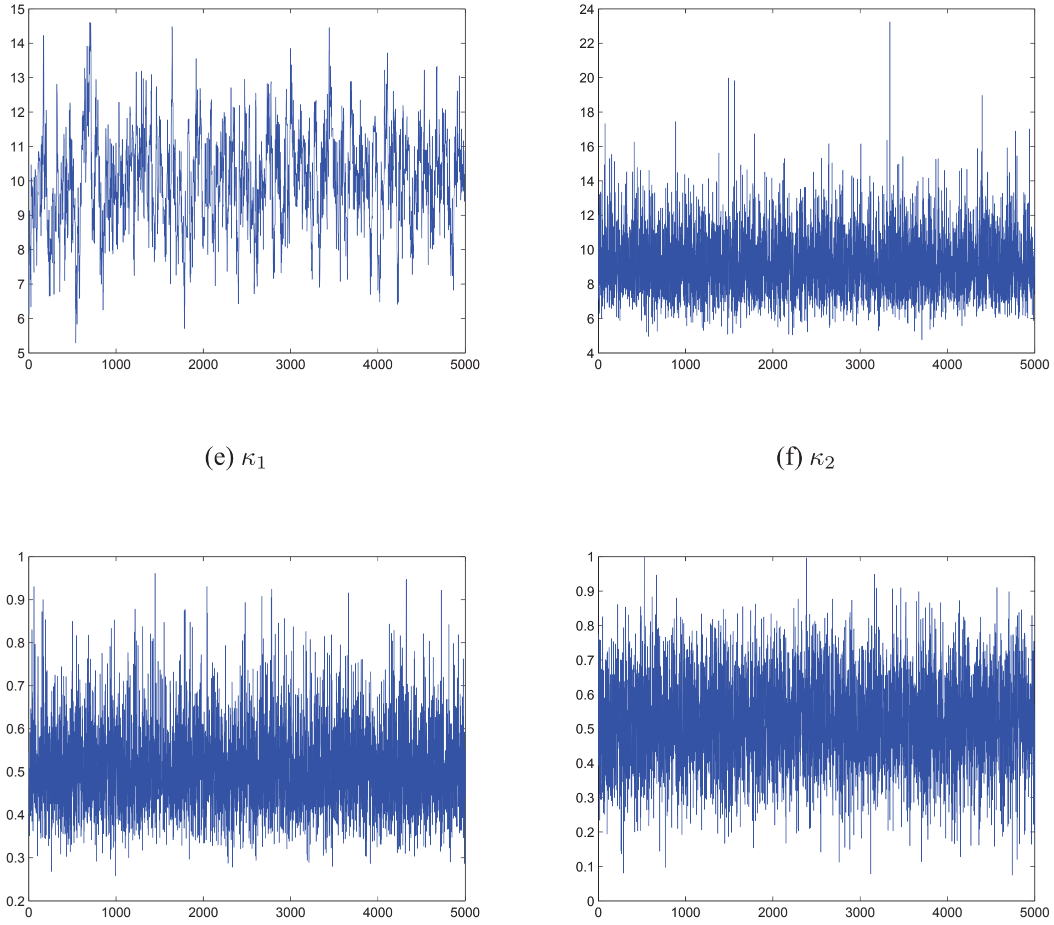
(a) μ_{11}

(b) μ_{12}

(c) μ_{21}

(d) μ_{22}

To this sake, we adopt the Bag-of-visual-words (BoVW) approach; thereby each image is represented by a single vector of frequencies. We start by detecting local regions on each image, using difference-of-Gaussian (DoG) detector, which we describe using their SIFT descriptors [85], giving 128 dimensional vector for each local region. Extracted vectors are clustered using the K-Means algorithm providing 700 visual-words vocabulary. We have tested several vocabulary sizes and the best classification results were obtained with 700 visual words, as illustrated in Fig.4.5(b).



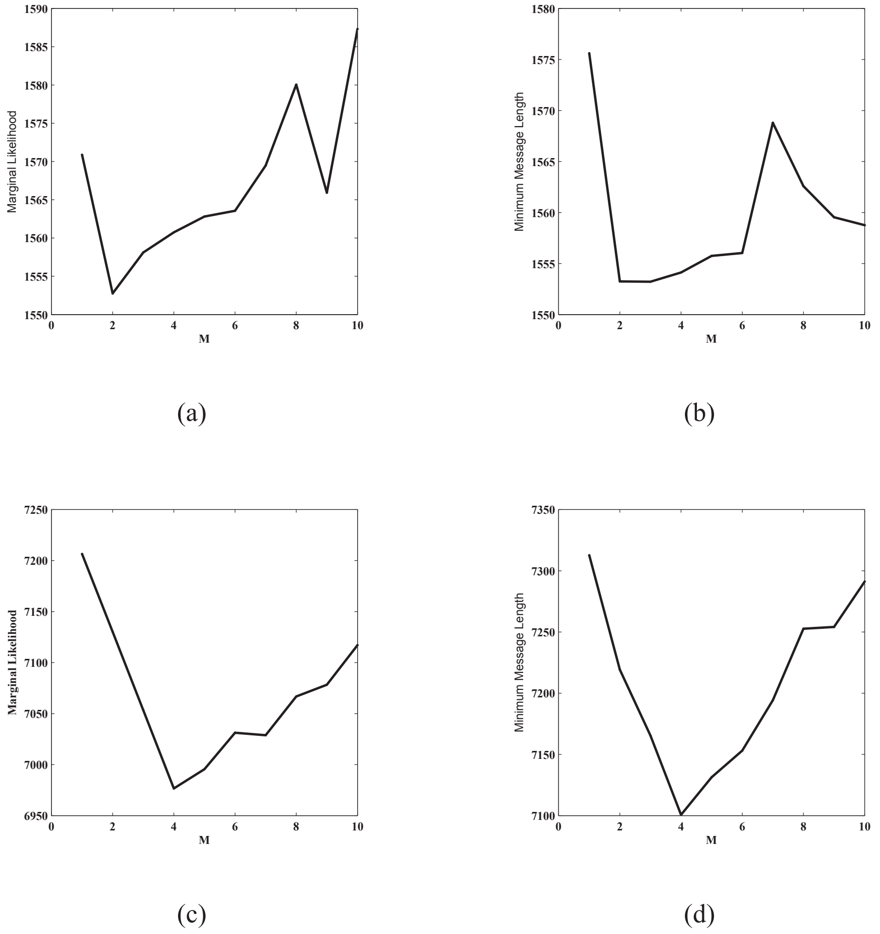
(e) κ_1

(f) κ_2

(g) p_1

(h) p_2

Figure 4.2: Time series plot of Gibbs-within-Metropolis for the first dataset.



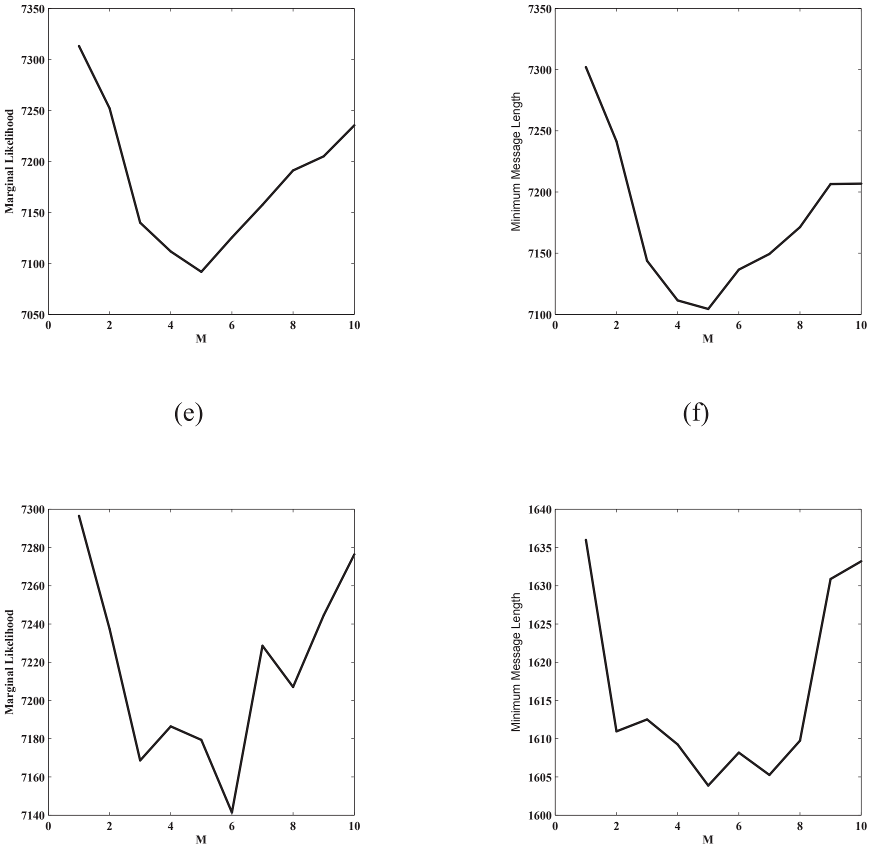
(a)

(b)

(c)

(d)

Each image in the dataset is then represented by a 700-dimensional vector describing the frequencies of a set of visual words, provided from the constructed visual vocabulary. Having these feature vectors, the Probabilistic Latent Semantic Analysis (pLSA) model is applied by considering 32 topics, for dimensionality reduction which has been shown to improve classification performance. Figure 4.5(a) shows that the choice of the number of aspects has a real impact on the accuracy of filtering and the optimal accuracy obtained when the number of aspects is set to 32. Finally, we employ our Langevin mixture model (LMM) model as a classifier to categorize images by assigning the testing image to the group which has the highest posterior probability according to Baye's



(e) (f)

(g) (h)

Figure 4.3: Number of clusters found for the different generated datasets using both the Bayesian (a,c,e,g) and EM (b,d,f,h) approaches.

decision rule. After we prepared our dataset, we have used 684 images (which are included in ”train+val” set) for training and 859 images (in ”test2” set) for testing. We evaluated the categorization performance of the proposed algorithm by running it 20 times.

We plot the receiver operating characteristic (ROC) curve using the code provided in the challenge toolkit. In particular, ROC curve is a visualization tool which illustrates the performance of our classifier by plotting the fraction of true positives out of the total actual positives vs. the fraction

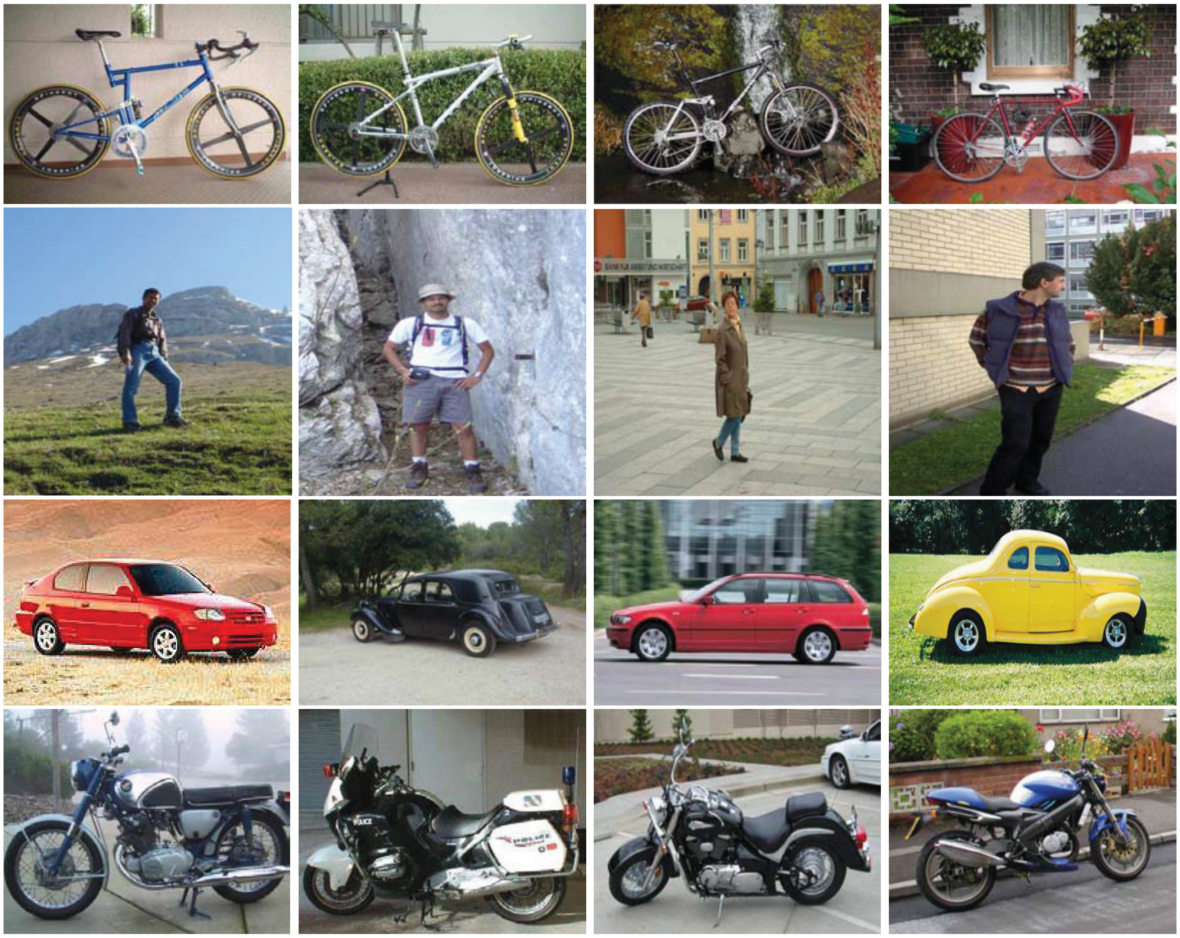
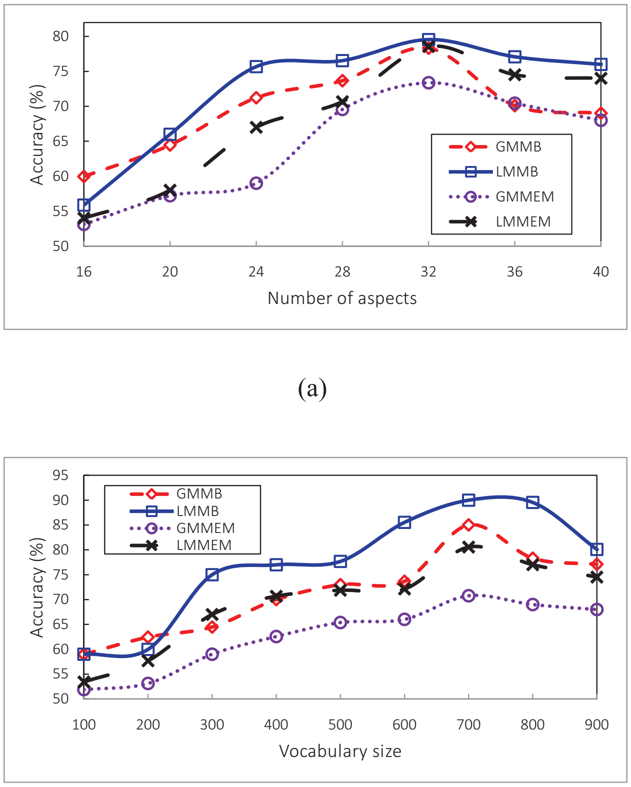


Figure 4.4: Sample images from each category in the VOC data set.

of false positives out of the total actual negatives. Figure 4.6 shows the ROC curve for the categorization of four categories. From these curves, it is evident that categorization using Bayesian inference based on LMM improves the categorization performance. However, for person category the lack of improvement is due to the wide variation of backgrounds in these images. A further enhancement can be considered by applying HOG (histogram of oriented gradients) detector which was originally optimized to detect people.



(a)

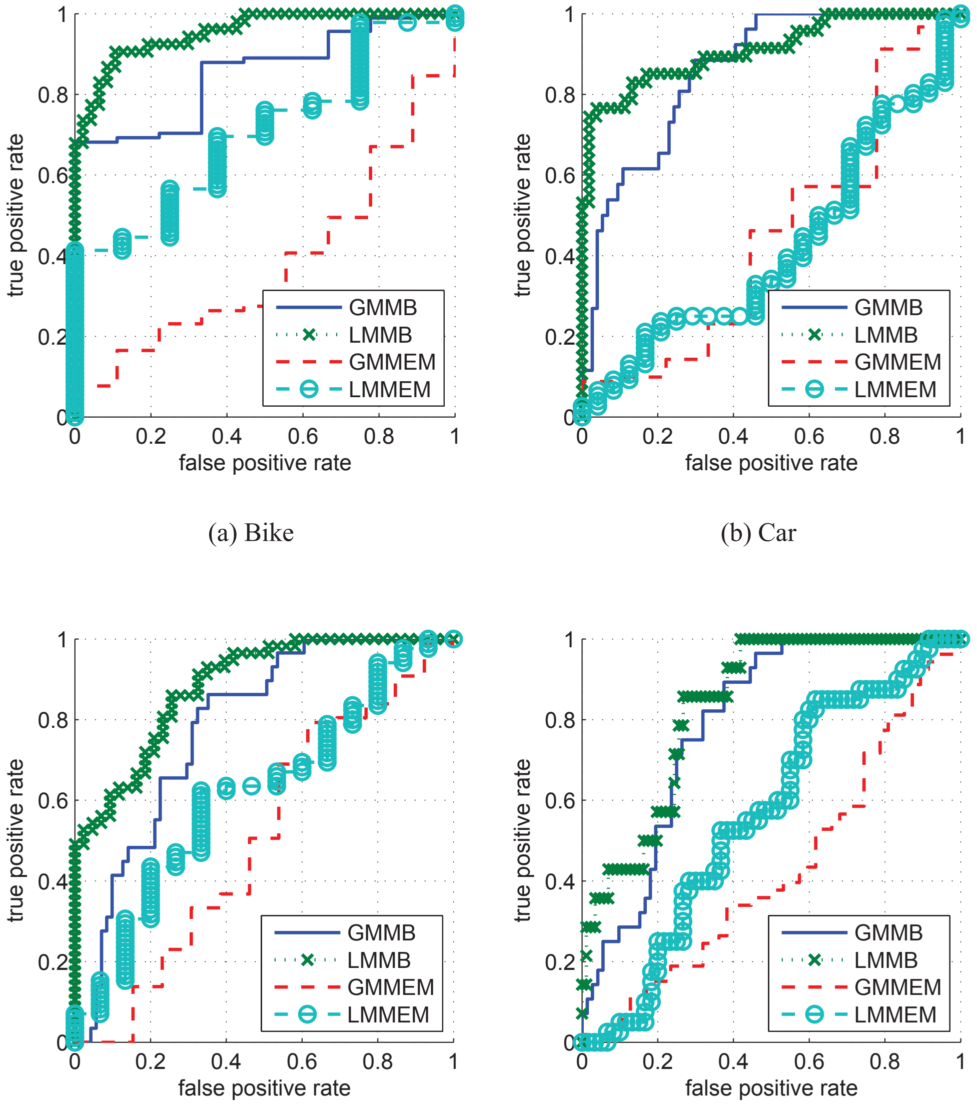
(b)

Figure 4.5: Classification accuracy for the PASCAL VOC2005 dataset as a function of (a) the number of aspects, (b) the vocabulary size.

4.4.3 Topic Detection and Tracking

Topic detection and tracking (TDT) is a challenging problem which has been the subject of extensive research in the past [160, 161]. The original research for TDT was initiated in Defense Advanced Research Projects Agency (DARPA) [162] and driven by the demand of deep insight into tremendous amounts of news. In this chapter we formulate TDT task as a text clustering [163] problem of partitioning stories (from different topics) distributed on unit sphere.

Datasets: TDT results are presented on two public news datasets, namely, The Topic Detection



(a) Bike

(b) Car

(c) Motorcycle

(d) Person

Figure 4.6: ROC curves for the categorization of four object categories in PASCAL VOC2005.

and Tracking (TDT-2) dataset [164] and 20 Newsgroup dataset[3]. TDT-2 dataset contains news stories classified into 96 topics and has been collected in 1998 from six sources: two newswires (Associated Presss World Stream and New York Times), two radio programs (Voice of America and Public Radio International's The World) and two television programs (CNN and ABC). The TDT-2 corpus is subdivided into three two-month sets: a training set (Jan-Feb), a development test set (Mar-Apr), and an evaluation set (May-Jun). In preprocessing step, we removed the documents that belong to several topics, and hence, only 30 topics were left, resulting in 9394 patterns over 36771 dimensions. On the other hand, 20 Newsgroup was collected from UseNet postings over several months in 1993. In our experiments, we find 26214 patterns over 24876 dimensions classified into 20 categories. Figure 4.7 shows the distribution of documents over topics in both datasets. It is clear that TDT-2 is unbalanced where some topics have less than 60 documents while others have more than 18000 document. On contrary, in 20 Newsgroup dataset documents are fairly distributed over different topics.

Evaluation Criteria: We evaluated the proposed framework for TDT problem using typical evaluation criteria that have been used, for instance, in the context of text clustering. We reported the execution time of the batch framework on an Intel(R) Core(TM) 64 Processor PC with the Windows XP Service Pack 3 operating system and a 4 GB main memory. Moreover, we calculated F_1(micro-averaged) measure as follows:

$$\begin{array}{c} F_1(\text{micro-averaged}) = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \\ \text{where} \\ \text{Precision} = \frac{\text{number of documents correctly predicted in class}i}{\text{number of documents in class}i} \\ \text{Recall} = \frac{\text{number of documents correctly predicted in class}i}{\text{number of correct prediction of class}i} \end{array} \tag{50}$$

It is worth mentioning that the larger values of $F_1 \in (0, 1)$ represent higher classification quality. In addition, we calculated normalized mutual information (NMI) [7] criterion as an external measure

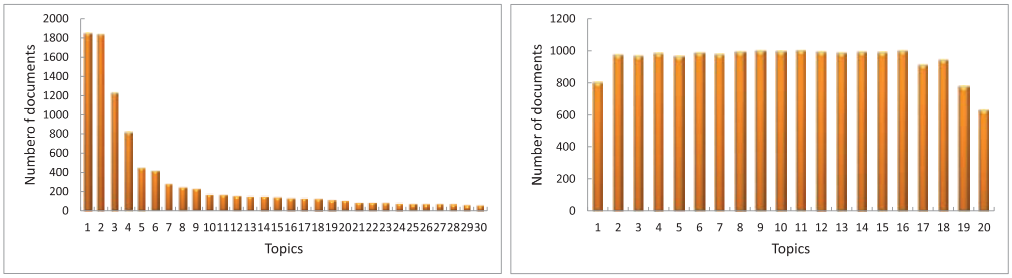
[3] http://kdd.ics.uci.edu/databases/20newsgroups

of how well the clustering results conform to existing class labels:

$$NMI = \frac{\sum_{j,c} N_{j,c} \log \frac{N_{j,c}}{N_j N_c}}{\sqrt{(\sum_j N_j \log \frac{N_j}{N})(\sum_c N_c \log \frac{N_c}{N})}}$$

where N_j is the number of stories in cluster j, N_c is the number of stories in true classes labels c, and $N_{j,c}$ is the number of stories that are in cluster j and class c. The larger value of NMI reflects better clustering. It is noteworthy that NMI, is unbiased towards high number of clusters M as purity and entropy criteria.

Results: We start by preparing our data by extracting the text of the news articles. Next, we applied stemming and removed stop words. This gives us vocabulary of words for each dataset. Next, each article is described as a L_2-normalized frequency vector, then each topic can be modeled accurately using Langevin distributions. Figure 4.8 shows F1(micro-averaged) vs. number of topics for both Bayesian and EM approaches based on Langevin mixture model (LMM) and Gaussian mixture model (GMM). In all the experiments, both mixtures achieve their higher value at $M = 20$ and $M = 30$ for 20 Newsgroup and TDT-2 datasets, respectively, which is the correct number of topics. Tables 4.2 and 4.3 illustrate the average normalized mutual information (NMI), the estimated number of components (M^*) and runtime results averaged over 5 runs. We compared Langevin mixture model learned using EM algorithm with (LMM^{EM}+FS)and without feature selection (LMM^{EM}), Langevin mixture model learned using Bayesian algorithm with (LMM^B+FS) and without feature selection (LMM^B), Gaussian mixture model learned using EM algorithm with (GMM^{EM}+FS) and without feature selection (GMM^{EM}), Gaussian mixture model learned using Bayesian algorithm with (GMM^B+FS) and without feature selection (GMM^B), spherical k-means, k-means, and Latent Dirichlet Allocation (LDA). All results in these tables are shown in the format of (average± standard deviation). According to these tables, LMM^B shows a better detection over the rest of the models in all experiments. Indeed, we clearly see slight improvement of LMM detection over GMM which has been extensively used in the past. This result is expected since we are modeling vector of stories that are defined on the unit hypershpere (i.e. non-Gaussian



(a) TDT-2

(b) 20 Newsgroup

Figure 4.7: Analysis of datasets.

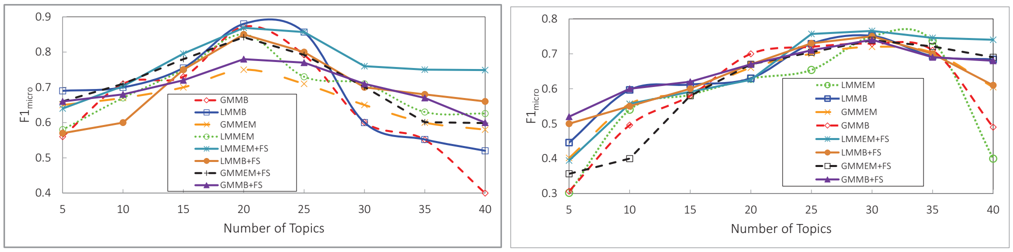
Table 4.2: Performance of TDT framework for TDT-2 dataset based on different models averaged over 5 runs

	Evaluation Criteria		
	NMI	M^*	Runtime(sec)
LMM^B	0.74	30±1.00	888
LMM^B+FS	0.91	30±0.08	521
LMM^{EM}	0.71	30±1.90	390
LMM^{EM}+FS	0.89	30±0.14	230
GMM^B	0.71	30±2.05	897
GMM^B+FS	0.79	30±1.14	530
GMM^{EM}	0.70	30±2.10	410
GMM^{EM}+FS	0.79	30±1.36	294
Spherical k-means	0.67	29±1.56	488
k-means	0.66	27±0.20	491
LDA	0.60	30±4.01	400

vectors) and hence Langevin mixture (spherical distribution) provides a better clustering. It is noteworthy that the improvement on the performance came on the cost of time as clearly shown in the tables. However, one can also clearly observe that the results are further improved when feature selection is considered while processing time has somehow decreased in all models. Moreover, we can clearly see that we obtained better results in terms of NMI than [6], where vMF, LDA and

Table 4.3: Performance of TDT framework for 20 Newsgroup dataset based on different models averaged over 5 runs.

	Evaluation Criteria		
	NMI	M^*	Runtime(sec)
LMM^B	0.70	20±1.79	720
LMM^B+FS	0.77	20±1.31	650
LMM^{EM}	0.68	20±2.56	320
LMM^{EM}+FS	0.76	20±0.49	197
GMM^B	0.69	20±2.01	725
GMM^B+FS	0.75	20±0.01	668
GMM^{EM}	0.67	20±2.09	356
GMM^{EM}+FS	0.74	20±0.01	211
Spherical k-means	0.58	21±2.00	500
k-means	0.54	18±5.41	510
LDA	0.60	20±0.11	350



(a) 20 Newsgroup

(b) TDT-2

Figure 4.8: F1(micro-averaged) vs number of topics in TDT framework for both Bayesian and EM approaches based on Langevin mixture model (LMM)& Gaussian mixture model (GMM).

EDCM were used for the 20 Newsgroup dataset. In another interesting work authors in [165] have evaluated TDT-2 dataset using five approaches, namely, Canonical K-means, K-means clustering in the principle components subspace, Normalized Cut, Graph Regularized Nonnegative Matrix Factorization (GNMF), Nonnegative Matrix Factorization (NMF). It is noteworthy though that our proposed TDT framework has a comparable performance with GNMF framework, in terms of NMI, which achieved the best results in all their experiments.

In spite of the promising results achieved in this chapter, further enhancement can be done. A crucial factor, for instance, that could be the subject of future investigation is the extension of the proposed model to the infinite cases using Dirichlet processes.

Chapter 5

On Nonparametric Bayesian Spherical Data Clustering and Feature Selection

In this chapter we develop an Infinite model tailed to spherical data, specified hierarchically within the Bayesian paradigm, and we discuss the construction of its priors. We then develop the complete posteriors from which the model's parameters are simulated. Moreover, we propose an infinite framework that allows simultaneous feature selection selection and parameter estimation. We end this chapter by presenting some experimental results and discuss the merits of proposed model over others.

5.1 Introduction

The dramatic growth in data collection techniques has resulted in large dynamically growing multimedia datasets. As a result, the huge amount of everyday generated data is pushing used static models to their limits. To this end, data mining and machine learning techniques are widely used to understand and model the content of these datasets. In particular, in this Chapter, we approach this issue using Bayesian nonparametric approaches for modeling and selection using mixture of Dirichlet processes [166] which has been shown to be a powerful alternative to select the number of clusters. In contrast to classic Bayesian approaches (we proposed in Chapter 4) which suppose an unknown finite number of mixture components, nonparametric Bayesian approaches assume an infinite number of components. Indeed, nonparametric Bayesian approaches allow the increasing of the number of mixture components to infinity, which removes the problems underlying the selection of the number of clusters which can increase or decrease as new data arrive. Because of their simplicity and thanks to the development of MCMC techniques, infinite mixture models based on Dirichlet processes are now widely used in different domains and variety of applications. To this end, clustering analysis using a set of given features is often used to identify clusters that are distinct from one another. However, the data at certain domains present unique challenges. For instance, the majority of research on TDT have stressed the importance of words in identifying the topic of the story [167] and hence have applied feature selection as a preprocessing step [6, 162, 167]. Since different features will yield different clustering results, a feature selection (weighting) process is desirable to reach optimal clustering results. Previously, we proposed [34, 36] a feature selection that allow simultaneous feature selection and parameter estimation. Although computationally efficient, these methods would fail when the number of clusters varies according to the dynamic changes of data over time. To address these limitations, we propose a nonparametric Bayesian approach that takes into account selection of informative features at the same time.

5.2 Infinite Langevin Mixture Model

Consider the problem of clustering N multimedia objects, let $\mathcal{X} = (\vec{X}_1, \ldots, \vec{X}_N)$ be a set of independent vectors representing N objects (e.g. text, video, etc), where $\vec{X}_i = (X_{i1}, \ldots, X_{id})$ follows a Langevin distribution if its probability density function is given by [24]:

$$p_D(\vec{X}|\vec{\mu}, \kappa) = \exp\{\kappa \vec{\mu}^T \vec{X} - a_D(\kappa)\} \tag{1}$$

on the $(D-1)$-dimensional unit sphere $\mathbb{S}^{D-1} = \{\vec{X}|\vec{X} \in \mathbb{R}^D : ||\vec{X}|| = \sqrt{\vec{X}^T\vec{X}} = 1\}$, with mean direction unit vector $\vec{\mu} \in \mathbb{S}^{D-1}$, where $\vec{\mu}^T$ denotes the transpose of $\vec{\mu}$ and non-negative real concentration parameter $\kappa \geq 0$. And $a_D(\kappa) = -\log\left\{\frac{\kappa^{\frac{D}{2}-1}}{(2\pi)^{\frac{D}{2}} I_{\frac{D}{2}-1}(\kappa)}\right\}$ is the normalizing constant function, where $I_D(\kappa)$ denotes the modified Bessel function of first kind [24]. By considering a Dirichlet process, $\mathcal{X}$ can be modeled using a set of latent parameters $\{\theta_1, \ldots, \theta_N\}$ where each $\vec{X}_i$ has distribution $F(\theta_i)$ and each θ_i is drawn independently and identically from a mixing distribution G on which a Dirichlet process prior is placed:

$$\vec{X}_i|\theta_i \sim F(\theta_i)$$

$$\theta_i|G \sim G$$

$$G \sim DP(G_0, \eta)$$

where G_0 and η define a baseline distribution for the Dirichlet process prior and the concentration parameter, respectively.

In infinite mixture model, we need to specify the prior distribution of the mixing proportions p_j. Thus, we know that $(0 < p_j < 1$ and $\sum_{j=1}^{M} p_j = 1)$, then the typical choice, as a prior is symmetric Dirichlet distribution with the positive parameters $\frac{\eta}{M}$:

$$p(\vec{P}|\eta) = \frac{\Gamma(\eta)}{\prod_{j=1}^{M}\Gamma(\frac{\eta}{M})}\prod_{j=1}^{M} p_j^{\frac{\eta}{M}-1} \tag{2}$$

where

$$p(Z|\vec{P}) = \prod_{j=1}^{M} p_j^{n_j} \tag{3}$$

which can be easily determined using $p(Z_i = j) = p_j, j = 1, \ldots, M$. Then using the standard Dirichlet integral, we may integrate out the mixing proportions and write the prior directly in terms of the indicators from which we can show:

$$p(Z|\eta) = \int_{\bar{P}} p(Z|\vec{P})p(\vec{P}|\eta)d\vec{P} = \frac{\Gamma(\eta)}{\Gamma(\eta+N)}\prod_{j=1}^{M}\frac{\Gamma(\frac{\eta}{M}) + n_j}{\Gamma(\frac{\eta}{M})} \tag{4}$$

which can be considered as a prior on Z. In order to be able to use Gibbs sampling for the missing vector, Z, we need the conditional prior for a single indicator given all the others; this can be easily obtained from Eq. 4 by keeping all but a single indicator fixed, we can show that [166, 168]

$$p(Z_i = j|\eta, Z_{-i}) = \frac{n_{-i,j} + \frac{\eta}{M}}{N - 1 + \eta} \tag{5}$$

where $Z_{-i} = \{Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_n\}$, where $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_{ij=1}}$ is the number of vectors previously affected to cluster j and $n_{-i,j}$ is the number of vectors, excluding $\vec{X}_i$, in cluster j. Letting $M \longrightarrow \infty$ in Eq. 5, the conditional prior gives the following limits [166, 168]

$$p(Z_i = j|\eta, Z_{-i}) \begin{cases} \frac{n_{-i,j}}{N-1+\eta}, & \text{if } n_{-i,j} > 0 (\text{cluster} j \in \mathcal{R}) \\ \frac{\eta}{N-1+\eta}, & \text{if } n_{-i,j} = 0 (\text{ cluster} j \in \mathcal{U}) \end{cases} \tag{6}$$

where $\mathcal{R}$ and $\mathcal{U}$ are the sets of nonempty and empty clusters, respectively. We notice that $p(Z_i = j|\eta, Z_{-i}) = p(Z_i \neq Z_{i'} \forall i \neq i'|\eta, Z_{-i})$ when $n_{-i,j} = 0$.

5.2.1 Priors and Posteriors

The main goal here is to obtain the conditional posterior distributions of our infinite models parameters given the data to cluster. This requests the choice of prior distributions. First, we need to choose priors for mixture's parameters. In this chapter, we suppose that the mixture parameters are independent realizations from appropriately selected distributions. Thus, for the mean $\vec{\mu}_j$ we choose Langevin distribution with μ_0, κ_0 as the mean and concentration parameters. For concentration parameter κ_j we choose Gamma distribution with a, b as the shape and rate parameters, given by:

$$\mu_{jd} \sim \mathcal{LM}(\mu_0, \kappa_0) \quad \kappa_j \sim \mathcal{G}(a, b) \tag{7}$$

where $a > 0$, $b > 0$, μ_0 and $\kappa_0 > 0$ are the hyperparameters chosen common to all components. Having these priors in hand, the conditional posteriors of $\vec{\mu}_j$ and κ_j can now be determined as follows:

$$p(\vec{\mu}_j|\ldots) = \exp(\kappa_0\mu_0\mu_{jd}) \times \prod_{Z_i=j}\prod_{d=1}^{D}\exp(\kappa_j\mu_{jd}X_{id}) \tag{8}$$

$$p(\kappa_j|\ldots) = \frac{\kappa_j^{a-1}b^a\exp(-b\kappa_j)}{\Gamma(a)} \times \Big[\frac{\kappa^{\frac{D}{2}-1}}{(2\pi)^{\frac{D}{2}}I_{\frac{D}{2}-1}(\kappa)}\Big]^{n_j}\prod_{Z_i=j}\exp(\kappa_j\mu_{jd}X_{id}) \tag{9}$$

In order to add more flexibility, we add another layer to the Bayesian hierarchy as we develop prior distribution on the hyperparameters: a, b, μ_0 and κ_0. Thus, we select the following priors for the hyperparameters:

$$\mu_0 \sim \mathcal{LM}(\mu_1,\kappa_1) \quad \kappa_0 \sim \mathcal{G}(a_1,b_1) \quad a \sim \mathcal{IG}(\alpha,\beta) \quad b \sim \mathcal{G}(\nu,\omega)$$

Having these priors, the conditional posteriors of the hyperparameters are given by:

$$p(\mu_0|\ldots) \propto p(\mu_0|\mu_1,\kappa_1)\prod_{j=1}^{M}p(\vec{\mu}_j|\mu_0,\kappa_0) \propto \exp(\kappa_1\mu_1\mu_0)\prod_{j=1}^{M}\prod_{d=1}^{D}\exp(\kappa_0\mu_0\mu_{jd}) \tag{10}$$

$$p(\kappa_0|\ldots) \propto p(\kappa_0|a_1,b_1)\prod_{j=1}^{M}p(\vec{\mu}_j|\mu_0,\kappa_0) \propto \kappa_0^{a_1-1}\exp(-\kappa_0 b_1)\prod_{j=1}^{M}\exp(\kappa_0\mu_0\mu_{jd}) \tag{11}$$

$$p(a|\ldots) \propto p(a|\alpha,\beta)\prod_{j=1}^{M}p(\kappa_j|a,b) \propto \frac{\exp(\frac{-\beta}{a})}{a^{\alpha+1}}\Big[\frac{b^a}{\Gamma(a)}\Big]^{Md}\prod_{j=1}^{M}\kappa_j^{a-1}\exp(-b\kappa_j) \tag{12}$$

$$p(b|\ldots) \propto p(b|\nu,\omega)\prod_{j=1}^{M}p(\kappa_j|a,b) \propto b^{\nu-1}\exp(-\omega b)\Big[\frac{b^a}{\Gamma(a)}\Big]^{Md}\prod_{j=1}^{M}\kappa_j^{a-1}\exp(-b\kappa_j) \tag{13}$$

Having the conditional priors in Eq. 6, the conditional posteriors are obtained by combining these priors with the likelihood of the data [169]

$$p(Z_i = j|\ldots)\begin{cases}\frac{n_{-i,j}}{N-1+\eta}p(\vec{X}_i|\theta_i), & \text{if } j \in \mathcal{R}\\ \int\frac{\eta p(\vec{X}_i|\theta_i)p(\theta_i|\mu_1,\kappa_1,a_1,b_1,\alpha,\beta,\nu,\omega)}{N-1+\eta}d\theta_i, & \text{if } j \in \mathcal{U}\end{cases} \tag{14}$$

where

$$p(\theta_i|\mu_1,\kappa_1,a_1,b_1,\alpha,\beta,\nu,\omega) = p(\kappa_j|a,b)\prod_{d=1}^{D}p(\mu_{jd}|\mu_0,\kappa_0) \tag{15}$$

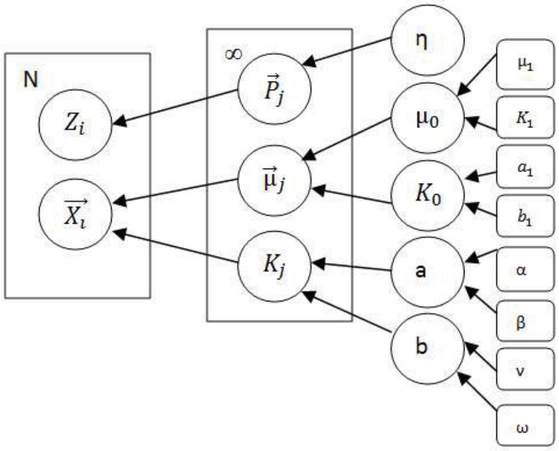


Figure 5.1: Graphical Model representation of the Bayesian hierarchical Langevin mixture model. Nodes in this graph represent random variables, rounded boxes are fixed hyperparameters, boxes indicate repetition (with the number of repetitions in the upper left) and arcs describe conditional dependencies between variables.

5.2.2 The Complete Algorithm

Our complete algorithm can be summarized as follows:

1. Generate Z_i from Eq. 14 then update $n_j, j = 1, \ldots, M, i = 1, \ldots, N$.

2. Update the number of represented components M.

3. Update the mixing parameters for the represented components by $P_j = \frac{n_j}{N+\eta}$ for $j = 1, \ldots, M$ and for the unrepresented components by $P_U = \frac{\eta}{\eta+N}$.

4. Generate the mixture parameters μ_j and κ_j from Eqs. 8 and 9.

5. Update the hyperparameters: generate μ_0, κ_0, a and b from Eqs. 10, 11, 12 and 13, respectively.

Note that in the initialization step, the algorithm started by assuming that all the vectors are in the same cluster and the initial parameters are generated as random samples from their prior distribution. The distributions given by Eqs. 10, 11, 12 and 13 are not of standard forms. However, it is possible to show that they are log-concave (i.e it is straightforward to show that the second derivatives of the logarithms of these functions are negative), then the samples generation is based on the adaptive rejection sampling (ARS) [152]. The sampling of the vectors Z_i requires the evaluation of the integral in Eq. 14 which is not analytically tractable. Thus, we have used an approach, originally proposed in [166] which consists on approximating this integral by generating a Monte Carlo estimate by sampling from the priors of μ_j and κ_j. The sampling of μ_j and κ_j is more complex, since the posteriors given by Eqs. 8 and 9 do not have known forms. Thus, we have used the random walk Metropolis-Hastings algorithm. At iteration t, the steps of the M-H algorithm to generate μ_j and κ_j can be described as follows:

- Generate $\tilde{\mu}_j \sim \mathcal{LM}(\mu_j^{t-1}|\tilde{\mu}_j, \kappa_j)$, $\tilde{\kappa}_j \sim \mathcal{LN}(\log(\kappa_j^{t-1}), \sigma^2)$ and $\mathbf{u} \sim \mathcal{U}_{[0,1]}$
- Compute:
  - $r_\mu = \frac{p(\tilde{\mu}_j|...)\mathcal{LM}(\mu_j^{t-1}|\tilde{\mu}_j,\kappa_j)}{p(\mu_j^{(t-1)}|...)\mathcal{LM}(\tilde{\mu}_j|\mu_j^{(t-1)},\kappa_j)}$
  - $r_\kappa = \frac{p(\tilde{\kappa}_j|...)\mathcal{LN}(\log(\kappa_j^{t-1}|\tilde{\kappa}_j))}{p(\kappa_j^{t-1}|...)\mathcal{LN}(\log(\tilde{\kappa}_j|\kappa_j^{t-1}))}$
- if
  - $r_\mu < u$ then $\mu_j^t = \tilde{\mu}_j$ else $\mu_j^t = \mu_j^{t-1}$.
  - $r_\kappa < u$ then $\kappa_j^t = \tilde{\kappa}_j$ else $\kappa_j^t = \kappa_j^{t-1}$.

The convergence of MCMC is based on a single long-run of the Gibbs sampler.

5.3 Infinite Langevin Clustering and Feature Selection

As we previously proved (See Chapter 3 and Chapter 4) that selecting the informative features has improved the performance of the classifier and enhanced the quality of cluster. Thus, in this section we will combine infinite clustering and feature selection based on movM learned using nonparametric Bayesian approaches. Thus, let $p(\vec{X}_i|\Theta_M)$ be a mixture of M vM distributions represented by Eq. 1. Therein, considering feature selection into account, we assume that a given feature is relevant if it follows vM distribution $p(Y_{id}|\theta_{jd})$ across clusters and irrelevant if it follows

$p(Y_{id}|\theta_{jd}^{irr})$ a vM distribution also, can be written as:

$$p(\vec{X}_i|\Theta) = \sum_{j=1}^{M} p_j \prod_{d=1}^{D} (\rho_{jd} p(Y_{id}|\theta_{jd}) + (1-\rho_{jd}) p(Y_{id}|\theta_{jd}^{irr})) \tag{16}$$

where , $\Theta = \{\Theta_M, \vec{\rho}, \vec{\theta}^{irr}\}$, $\vec{\theta}^{irr} = (\theta_1^{irr}, \dots, \theta_M^{irr})$, as $\theta_{jd}^{irr} = (\mu_{jd}^{irr}, \kappa_{jd}^{irr})$ are the parameters of vM from which the irrelevant feature is drawn, and $\theta_{jd} = (\mu_{jd}, \kappa_{jd})$ re the parameters of vM from which the relevant feature is drawn. $\vec{\rho} = (\vec{\rho}_1, \dots, \vec{\rho}_M)$ such that $\vec{\rho}_j = (\rho_{j1}, \dots, \rho_{jD})$ where each $0 \leq \rho_{jd} \leq 1$ denotes the weight of the d^{th} feature on cluster j.

5.3.1 Bayesian Hierarchial Model

The joint distribution of our model is given by:

$$\begin{aligned} p(\vec{P}, Z, z, \vec{\rho}, \vec{\theta}, \vec{\theta}^{irr}, \mathcal{X}) &= p(\mathcal{X}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}) p(\vec{\theta}^{irr}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta}) \\ &\times p(\vec{\theta}|\vec{P}, Z, \vec{\rho}, z) p(\vec{P}) p(Z|\vec{P}) p(\vec{\rho}|\vec{P}, Z) \qquad (17) \\ &\times p(z|\vec{\rho}, \vec{P}, Z) \qquad (18) \end{aligned}$$

where $Z = (Z_1, \dots, Z_N)$ denotes the missing allocation variables, such that Z_i shows the cluster that vector $\vec{X}_i$ was generated from, $z = (z_1, \dots, z_N)$, such that $z_i = (\vec{z}_{i1}, \dots, \vec{z}_{iM})$, where $\vec{z}_{ij} = (z_{ij1}, \dots, z_{ijD})$ are the missing binary vectors that indicate if a given feature Y_{id} is relevant or not. Moreover,

$$p(Z_i = j|\vec{X}_i) \propto p_j \prod_{d=1}^{D} (\rho_{jd} p(Y_{id}|\theta_{jd}) + (1-\rho_{jd}) p(Y_{id}|\theta_{jd}^{irr}))$$

is the probability that vector i is in cluster j, conditional on having observing $\vec{X}_i$. Indeed, proir probability that the feature Y_{id} is relevant for component j is given by:

$$p(z_{ijd} = 1, Z_i = j|\vec{X}_i) \propto \rho_{jd} p(Y_{id}|\theta_{jd}) p(Z_i = j|\vec{X}_i)$$

Note that we can easily deduce $p(z_{ijd} = 0, Z_i = j|\vec{X}_i)$ which is the case in which feature Y_{id} is irrelevant from Eq. 19. Next, we can impose some conditional independencies, such that:

$$p(\vec{\theta}|\vec{P}, Z, \vec{\rho}, z) = p(\vec{\theta}), \quad p(\vec{\theta}^{irr}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta}) = p(\vec{\theta}^{irr})$$

$$p(\vec{\rho}|\vec{P}, Z) = p(\vec{\rho}), \quad p(z|\vec{\rho}, \vec{P}, Z) = p(z|\vec{\rho}), \quad p(\vec{\theta}|Z, P) = p(\vec{\theta})$$

$$p(\mathcal{X}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}) = p(\mathcal{X}|Z, z, \vec{\theta}, \vec{\theta}^{irr}) = \prod_{i=1}^{N}\prod_{d=1}^{D}[(p(Y_{id}|\theta_{Z_{id}}))^{z_{id}}(p(Y_{id}|\theta^{irr}_{Z_{id}}))^{1-z_{id}}]$$

which give us the following joint distribution:

$$p(\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}, \mathcal{X}) = p(\vec{P})p(Z|\vec{P})p(\vec{\rho})p(z|\vec{\rho})p(\vec{\theta})p(\vec{\theta}^{irr})p(\mathcal{X}|\vec{P}, Z, z, \vec{\theta}, \vec{\theta}^{irr}) \qquad (19)$$

To add more flexibility we suppose that the model parameters ($\vec{\theta}$, $\vec{\theta}^{irr}$, $\vec{\rho}$, $\vec{P}$) follows priors depending on hyperparamters (Λ,Λ^{irr}, ξ, η), which are in turn drawn from independent hyperpriors: $p(\Lambda)$,$p(\Lambda^{irr})$,$p(\xi)$, and $p(\eta)$, respectively. Such that $\Lambda = (\Lambda_1, \ldots, \Lambda_M)$, $\Lambda_j = (\Lambda_{j|\mu}, \Lambda_{j|\kappa})$, and the same for the irrelevant model, where $\Lambda = (\Lambda_1^{irr}, \ldots, \Lambda_M^{irr})$ and $\Lambda_j = (\Lambda_{j|\mu}^{irr}, \Lambda_{j|\kappa}^{irr})$. Thus,

$$p(\vec{\theta}|\Lambda) = \prod_{j=1}^{M} p(\vec{\mu}_j|\Lambda_{j|\mu_j})p(\kappa_j|\Lambda_{j|\kappa_j}) \quad p(\vec{\theta}^{irr}|\Lambda^{irr}) = \prod_{j=1}^{M} p(\vec{\mu}_j^{irr}|\Lambda^{irr}_{j|\mu_j^{irr}})p(\kappa_j^{irr}|\Lambda^{irr}_{j|\kappa_j^{irr}})$$

Hence the joint distribution of our model is given by:

$$\begin{aligned} p(\vec{P}, Z, z, \vec{\rho}, \Lambda, \Lambda^{irr}, \eta, \xi, \theta, \theta^{irr}, \mathcal{X}) = {} & p(\Lambda)p(\Lambda^{irr})p(\xi)p(\eta) \\ & \times p(\vec{P}|\eta)p(Z|\vec{P})p(\vec{\rho}|\xi)p(z|\vec{\rho})p(\mathcal{X}|\vec{\theta}, \vec{\theta}^{irr}, Z, z) \\ & \prod_{j=1}^{M}\Big[p(\vec{\mu}_j|\Lambda_{j|\mu_j})p(\kappa_j|\Lambda_{j|\kappa_j})p(\vec{\mu}_j^{irr}|\Lambda^{irr}_{j|\mu_j^{irr}})p(\kappa_j^{irr}|\Lambda^{irr}_{j|\kappa_j^{irr}})\Big] \end{aligned} \qquad (20)$$

5.3.2 Nonparametric Bayesian Learning

In this section, we start by developing priors for our model parameters. Next, based on these priors, we develop posterior distributions accordingly. Later, we show how we extend our model to infinite case. Finally, we present the MCMC algorithm and complete learning algorithm.

priors and posteriors

For the mean $\vec{\mu}_j$ we consider a Von Mises prior with hyperparameters $\mu_0 \in S^D$ and κ_0 as the mean of the observations and the concentration parameters, respectively, and it is given by:

$$p(\vec{\mu}_j|\mu_0, \kappa_0) = \prod_{d=1}^{D} \frac{1}{2\pi I_0(\kappa_0)} \exp\{\kappa_0 \mu_0 \mu_{jd}\} \tag{21}$$

And $p(\vec{\mu}_j^{irr}|\mu_0, \kappa_0)$ has the same form as $p(\vec{\mu}_j|\mu_0, \kappa_0)$. Thus, the generic hyperparameters $\Lambda_{j|\theta_j}$ and $\Lambda^{irr}_{j|\theta_j^{irr}}$ become (κ_0, μ_0) and hence the conditional posterior distributions for θ_j and θ_j^{irr}, giving the rest of the parameters, are:

$$p(\vec{\mu}_j| \ldots) \propto p(\vec{\mu}_j|\kappa_0, \mu_0) p(\mathcal{X}|\vec{\theta}, \vec{\theta}^{irr}, Z, z) \tag{22}$$

$$p(\vec{\mu}_j^{irr}| \ldots) \propto p(\vec{\mu}_j^{irr}|\kappa_0, \mu_0) p(\mathcal{X}|\vec{\theta}, \vec{\theta}^{irr}, Z, z) \tag{23}$$

The hyperparameters μ_0 and κ_0 are given Von Mises and Gamma priors, respectively:

$$p(\mu_0|\mu_1, \kappa_1) = \frac{1}{2\pi I_0(\kappa_1)} \exp\{\kappa_1 \mu_1 \mu_0\} \quad p(\kappa_0|a_1, b_1) = \frac{\kappa_0^{a_1-1} b_1^{a_1} \exp(-b_1 \kappa_0)}{\Gamma(a_1)} \tag{24}$$

Thus, according to Eqs. 20, 21 and 24 we obtain the following posteriors:

$$p(\mu_0| \ldots) \propto p(\mu_0|\mu_1, \kappa_1) \prod_{j=1}^{M} p(\vec{\mu}_j|\kappa_0, \mu_0) p(\vec{\mu}_j^{irr}|\kappa_0, \mu_0) \tag{25}$$

$$p(\kappa_0| \ldots) \propto p(\kappa_0|a_1, b_1) \prod_{j=1}^{M} p(\vec{\mu}|\kappa_0, \mu_0) p(\vec{\mu}_j^{irr}|\kappa_0, \mu_0) \tag{26}$$

For the concentration parameter κ we choose Gamma distribution with a, b as the shape and rate parameters, given by:

$$p(\kappa_j|a, b) = \frac{\kappa_j^{a-1} b^a \exp(-b\kappa_j)}{\Gamma(a)} \tag{27}$$

And so $p(\kappa_j^{irr}|a,b)$ has the same form as $p(\kappa_j|a,b)$. Thus, the generic hyperparameters $\Lambda_{j|\theta_j}$ and $\Lambda^{irr}_{j|\theta_j^{irr}}$ become (a,b) and hence the conditional posterior distributions for θ_j and θ_j^{irr}, giving the rest of the parameters, are:

$$p(\kappa_j|\ldots) \propto p(\kappa|a,b)p(\mathcal{X}|\vec{\theta},\vec{\theta}^{irr},Z,z) \tag{28}$$

$$p(\kappa_j^{irr}|\ldots) \propto p(\kappa_j^{irr}|a,b)p(\mathcal{X}|\vec{\theta},\vec{\theta}^{irr},Z,z) \tag{29}$$

The hyperparameters a and b are given inverse Gamma and Gamma priors, respectively:

$$p(a|\alpha,\beta) = \frac{\beta^{\alpha}\exp(-\beta/a)}{\Gamma(\alpha)a^{\alpha+1}} \quad p(b|\nu,\omega) = \frac{b^{\nu-1}\omega^{\nu}\exp(-\omega b)}{\Gamma(\nu)} \tag{30}$$

Thus, according to Eqs. 20, 27 and 30 we obtain the following posteriors:

$$p(a|\ldots) \propto p(a|\alpha,\beta)\prod_{j=1}^{M} p(\kappa_j|a,b)p(\kappa_j^{irr}|a,b) \tag{31}$$

$$p(b|\ldots) \propto p(b|\nu,\omega)\prod_{j=1}^{M} p(\kappa_j|a,b)p(\kappa_j^{irr}|a,b) \tag{32}$$

As ρ_{jd} is defined in the compact support [0,1], we consider Beta distribution with parameters ϵ_1 and ϵ_2, common to all classes and dimensions, as prior:

$$p(\vec{\rho}|\epsilon) = \left[\frac{\Gamma(\epsilon_1+\epsilon_2)}{\Gamma(\epsilon_1)\Gamma(\epsilon_2)}\right]^{MD}\prod_{j=1}^{M}\prod_{d=1}^{D}\rho_{jd}^{\epsilon_1-1}(1-\rho_{jd})^{\epsilon_2-1} \tag{33}$$

Hence, the generic hyperparameter ϵ becomes (ϵ_1,ϵ_2). Recall that $\rho_{jd} = p(z_{jd}=1)$ and $1-\rho_{jd} = p(z_{jd}=0)$, $d=1,\ldots,D$, $j=1,\ldots,M$ thus each z_{jd} follows a D-variate Bernoulli distribution and we have

$$p(z|\vec{\rho}) = \prod_{i=1}^{N}\prod_{j=1}^{M}\prod_{d=1}^{D}\rho_{jd}^{z_{ijd}}(1-\rho_{jd})^{1-z_{ijd}} = \prod_{j=1}^{M}\prod_{d=1}^{D}\rho_{jd}^{f_{jd}}(1-\rho_{jd})^{N-f_{jd}} \tag{34}$$

where $f_{jd} = \sum_{i=1}^{N} \mathbb{I}_{z_{ijd}=1}$. Then, according to Eqs. 20, 33 and 34, we have

$$p(\vec{\rho}| \dots) \propto p(\vec{\rho}|\epsilon)p(z|\vec{\rho}) \propto \prod_{j=1}^{M}\prod_{d=1}^{D} \rho_{jd}^{f_{jd}+\epsilon_1-1}(1-\rho_{jd})^{N-f_{jd}+\epsilon_2-1} \tag{35}$$

Then, we suppose that the hyperparameters ϵ_1 and ϵ_2 are given Gamma priors with common hyperparameters $(\varepsilon_\epsilon, \varrho_\epsilon)$ which gives us the following posteriors:

$$p(\epsilon_1| \dots) = p(\epsilon_1|\varepsilon_\epsilon, \varrho_\epsilon)p(\vec{\rho}|\epsilon) \quad p(\epsilon_2| \dots) = p(\epsilon_2|\varepsilon_\epsilon, \varrho_\epsilon)p(\vec{\rho}|\epsilon) \tag{36}$$

The infinite Langevin Mixture Model with Feature selection

In order to allow an infinite number

$$p(Z_i = j| \dots) \begin{cases} \frac{n_{-i,j}}{N-1+\eta} p(\vec{X}_i|\Theta), & \text{if } j \text{is represented} \\ \int \frac{\eta p(\vec{X}_i|\Theta)p(\vec{\rho}_j|\epsilon)p(\theta_j|\Lambda_j)p(\theta_j^{irr}|\Lambda_j^{irr})}{N-1+\eta} d\theta_j d\theta_j^{irr} d\vec{\rho}_j, & \text{if } j \text{is not represented} \end{cases} \tag{37}$$

For η hyperparameter we chose an inverse Gamma prior with the parameters (χ_η, ψ_η) which is given by:

$$p(\eta|\chi_\eta, \psi_\eta) \propto \frac{\psi_\eta^{\chi_\eta} \exp(-\psi_\eta/\eta)}{\Gamma(\chi_\eta)\eta^{\chi_\eta+1}} \tag{38}$$

and hence the posterior can be written as [168]:

$$p(\eta| \dots) \propto p(\eta|\chi_\eta, \psi_\eta) \frac{\eta^M \Gamma(\eta)}{\Gamma(N+\eta)} \tag{39}$$

Complete Algorithm

Having all the posteriors in hand, we can employ a Gibbs sampler and each iteration will be based on the following steps:

- Generate Z_i from Eq. 37 and then update n_j, $j = 1, \dots, M$, $i = 1, \dots, N$.
- Update the number of represented components M.
- Update the mixing parameters for the represented components by $P_j = \frac{n_j}{N+\eta}$ for $j = 1, \dots, M$ and for the unrepresented components by $P_U = \frac{\eta}{\eta+N}$.

- Generate $\vec{z}_{ij}$ from a D-variate Bernoulli distribution with parameters $p(z_{ijd} = 1, Z_i = j|\vec{X}_i)$.
- Generate ρ_d, $\vec{\mu}_j$, $\vec{\mu}_j^{irr}$, κ_j and κ_j^{irr} from Eqs. 35, 22, 23, 28 and 29, $j = 1, \dots, M$, respectively.
- Update the hyperparameters: generate μ_0, κ_0, a, b, ϵ_1, ϵ_2 and η from Eqs. 25, 26, 31, 32, 36 and 39, respectively.

Note that distributions given by Eqs. 25, 26, 31, 32, 36 and 39 are not of standard forms. Thus one common solution is to sample using ARS approach [152]. Then we sample Z_i using the approach proposed in [166]. We have used the random walk Metropolis-Hastings algorithm (See Chapter 4 for further details) to sample μ_j, $\vec{\mu}_j^{irr}$, κ_j^{irr} and κ_j. The convergence of MCMC is based on a single long-run of the Gibbs sampler.

5.4 Experimental Results

In this section, we present the experimental results of applying proposed framework on high dimensional data extracted from challenging applications, namely, text categorization, and object detection. In these experiments we compare Infinite Langevin mixture model with ($ILMM$+FS) and without feature selection ($ILMM$) with other that have been used in the literature, namely, Infinite Gaussian mixture model with ($IGMM$+FS) and without feature selection ($IGMM$), Langevin mixture model learned using EM algorithm (LMM^{EM}), Langevin mixture model learned using Bayesian algorithm (LMM^{B}), Gaussian mixture model learned using EM algorithm (GMM^{EM}), Gaussian mixture model learned using Bayesian algorithm (GMM^{B}). In these applications, the values of the hyperparameters have been set experimentally as following $\varepsilon_\epsilon \in [1.8, 2.2]$, $\varrho_\epsilon \in [0.3, 0.7]$, $\chi_\eta \in [1.8, 2.2]$, $\psi_\eta \in [0.8, 1.2]$, $\nu \in [1.0, 2.0]$, $\omega \in [0.3, 0.8]$, $\alpha \in [1, 2.5]$, $\beta \in [0.1, 0.9]$, $\kappa_1 \in [1, 10]$, $a_1 \in\in [1.0, 2.0]$, $b_1 \in [0.3, 0.8]$ and $\mu_1 \in [0, 1]$. These choices have been found reasonable according to our experimental results. the feature saliency values are initialized at ($\rho = 0.5$).

Table 5.1: Classification F1 results for the Yahoo20 and WebKB datasets.

	$ILMM$	LMM^{EM}	LMM^{B}	$IGMM$	GMM^{EM}	GMM^{B}
Yahoo20	82.34	80.10	80.52	77.25	69.01	85.22
WebKB	79.88	73.12	73.59	78.15	66.93	71.04

5.4.1 Text Categorization

In the first experiment, we test our model for the classification of two well-known data sets consisting of a set of documents and which were used,namely, Yahoo20 and WebKB. Yahoo20[1] dataset which contains 2340 articles belonging to 20 categories (See Chapter 3). The WebKB are webpages collected by the World Wide Knowledge Base project of the CMU text learning group[2]. These pages were manually classified into seven different classes: student, faculty, staff, department, course, project, and other. However, we used only four categories [170] which are: student, faculty (1641), staff (1124), course (930) and project (504).

Among preprocessing approaches one common used approach to represent text documents is the BoW scheme. This scheme is based on representing each text document as a feature vector containing the frequencies of distinct words (after tokenisation, stemming, and stop-words removal) observed in the text. This gives us a vocabulary of words, next in the second step we normalized document's vectors using L2 normalization. The vectors in the different training sets were then modeled by our infinite mixture using the algorithm in the previous sections. After this stage, each class in the training set was represented by a mixture. Finally, in the classification stage each test vector was affected to a given class according to the Bayes classification rule. In order to evaluate our results we have used the F1 measure which combines the precision and recall measures.

Table 5.1 shows the classification results using $ILMM$, LMM^{EM}, LMM^{B}, $IGMM$, GMM^{EM} and GMM^{B} models. According to this table it is clear that infinite model produces better results than the finite one estimated for both LMM and GMM.

[1] fttp://fttp.cs.umn.edu/dept/users/boley

[2] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

5.4.2 Topic Detection and Tracking

TDT is a multifaceted issue which requires in-depth analysis, and hence, clustering algorithms should be able to construct flexible TDT models to adapt to real-time demands. For example, the dynamic anatomy of topics makes it difficult to keep pace; while new stories constantly continue to appear, outdated stories may disappear [6]. This in turn stresses the need to adequately update the detection model (clusters) through online learning. Indeed, it also highlights the deficiency of supervised TDT and directs the detection to unsupervised fashion [7]. Thus, in contrast to Chapter 4, which assumes a fix number of components, in the following experiments we model TDT problem by infinite Langevin mixture models.

The majority of research on TDT have stressed the importance of words in identifying the topic of the story and hence have applied feature selection as a preprocessing step. Thus, in our second experiment we enhance further our infinte TDT model by engaging feature selection in the detection process.

We use two public news datasets, namely, The Topic Detection and Tracking (TDT-2) dataset and 20 Newsgroup dataset (See Chapter 4 for further details). We start by preparing our data by extracting the text of the news articles. Next, we applied stemming. This gives us vocabulary of words for each dataset. Each article is then described as a L_2-normalized frequency vector. We run all the tested algorithms 5 times for evaluation.

Evaluation results for the two data sets generated by the $ILMM$, $ILMM$+FS, $IGMM$, $IGMM$+FS and are summarized in Tables 5.2 and 5.3. Clearly, the $ILMM$+FS and the $IGMM$+FS outperform the $ILMM$ and $IGMM$ models. Indeed, the clustering time has also decreased using feature selection approach. This can be explained by the fact that not all features have the same discriminative power during the clustering process, which is actually expected and confirms the conclusion we reached in previous Chapters. Finally, comparing these results with the one we previously achieved in Chapter 4, we clearly see that infinite model has boost the overall performance for involving distributions.

Table 5.2: Performance of TDT framework for TDT-2 dataset based on different models averaged over 5 runs

	Evaluation Criteria		
	NMI	M^*	Runtime(sec)
$ILMM$	0.75	30±0.90	796
$ILMM$+FS	0.91	30±0.81	490
$IGMM$	0.73	30±2.00	817
$IGMM$+FS	0.82	30±1.09	507

Table 5.3: Performance of TDT framework for 20 Newsgroup dataset based on different models averaged over 5 runs.

	Evaluation Criteria		
	NMI	M^*	Runtime(sec)
$ILMM$	0.78	20±1.00	647
$ILMM$+FS	0.79	20±0.09	649
$IGMM$	0.71	20±2.63	700
$IGMM$+FS	0.74	20±1.00	654

CHAPTER 6

Conclusion

In this digital era, when wide-scoped simulation studies have become of incredible potential, good representation of random inputs is crucial and a poor choice of the model may hurt the experiments. Many approaches have been proposed to model and analyze digital data. In this thesis, we have focused on spherical data clustering as we have introduced several framework to learn this kind of data.

Our first framework is based on combining generative and discriminative models which consists on developing probabilistic SVMs kernels from Langevin mixture. In particular, parameter estimation was carried out using EM approach and meanwhile we have proposed an algorithm to automatically select the number of components of Langevin mixture models using MML criterion. We also have developed SVM kernels as we have shown the existence of closed form expressions of probabilistic product kernels, Kullback-Leibler kernels, Rényi kernel and Jensen-Shannon kernel between two Langevin distributions. The validation was based on synthetic data, email categorization and spam filtering where we justified our choice of Langevin mixture model. Interesting application of proposed framework is the ability to learn bag of descriptors, instead of single bag of word (BoW), usually performed to classify multimedia objects. We explored this through application on spam filtering. Indeed, we have proposed a spam filtering framework adapted to the enriched multimedia content of emails. The main motivation was the fact that spammers have recently adopted image spam to defeat widely used text categorization based approaches. We have empirically proved that the simultaneous exploitation of both textual and visual information achieves better filtering results.

Next, we have proposed a principled statistical framework that simultaneously determines relevant features, number of clusters while being able to incrementally update model's parameters. To this aim, we have developed a MML objective that was minimized using EM in off-line scenario and RSEM for online scenario. Empirical experiments on spam image filtering and web documents clustering have proven that the proposed algorithm yields to good performance in terms of accuracy. In addition of being an efficient feature selection approach, the proposed algorithm has been shown to improve the quality of clustering in most cases. Besides, we have further improved our work to show how feature selection might influence modeling capabilities.

Moreover, we have proposed a novel Bayesian algorithm based on finite Langevin mixture. We therefore developed a conjugate prior distribution for Langevin mixture, exploiting the fact that it belongs to the exponential family of distributions. The idea of our approach is based on Monte Carlo simulation technique of Gibbs sampling mixed with a M-H step. Furthermore, we handle the estimation of the number of components using marginal likelihood with Laplace approximation. Later, we extended our Bayesian model to simultaneously handle the issue of feature selection. Experiments are carried out to investigate the performance of proposed algorithm on two challenging problems which are topic detection and tracking and image categorization. Reported results have shown that our algorithm is robust and outperforms several existing comparable methods. Indeed, our proposed methodology is flexible and can be easily generalized to deal with other applications.

Finally, an approach to generalize previous approaches for analyzing high-dimensional spherical data was developed in this thesis. In particular, we have described and illustrated a nonparametric Bayesian approach based on infinite Langevin mixture. Therein, we have shown that the problem of determining the number of clusters can be avoided by using infinite mixtures which model the structure of the data well. Indeed, the resulting optimal clustering is obtained by averaging over all number of clusters of different possible models. We also considered the problem of feature selection as we proposed a framework that allows simultaneous clustering and feature selection in infinite framework. Empirical experiments on text categorization and topic detection and tracking have proven that the proposed algorithms have provided good performance. Our model has several natural extensions such as the development of variational estimation approach which shall save a lot of computational time over Bayesian methods and later we could extend this work to the infinite case.

Appendix A

Proof of Equation 22

In the case of Langevin model, we can show that

$$\begin{aligned}
&\int_{\Omega} p(\vec{X}|\Theta)^{\rho} q(\vec{X}|\acute{\Theta})^{\rho} d\vec{X} \\
&= \Big[\Big(\frac{\kappa}{2}\Big)^{\frac{p}{2}-1} \frac{1}{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\kappa)}\Big]^{\rho} \Big[\Big(\frac{\acute{\kappa}}{2}\Big)^{\frac{p}{2}-1} \frac{1}{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\acute{\kappa})}\Big]^{\rho} \int_{\Omega} \Big(e^{\kappa\vec{\mu}^T\vec{X}}\Big)^{\rho} \Big(e^{\acute{\kappa}\vec{\acute{\mu}}^T\vec{X}}\Big)^{\rho} d\vec{X} \\
&= \Big[\Big(\frac{\kappa}{2}\Big)^{\frac{p}{2}-1} \frac{1}{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\kappa)}\Big]^{\rho} \Big[\Big(\frac{\acute{\kappa}}{2}\Big)^{\frac{p}{2}-1} \frac{1}{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\acute{\kappa})}\Big]^{\rho} \int_{\Omega} e^{(\kappa\vec{\mu}+\acute{\kappa}\vec{\acute{\mu}})^T\vec{X}\rho} d\vec{X}
\end{aligned} \tag{1}$$

The product of two Langevin can be written as

$$M_p(\vec{X}|\vec{\mu},\kappa) M_p(\vec{X}|\vec{\acute{\mu}},\acute{\kappa}) \propto M_p(\vec{X}|\tau_{\vec{\mu},\vec{\acute{\mu}}}, \xi_{\kappa,\acute{\kappa}})$$

where

$$\begin{aligned}
\xi_{\kappa,\acute{\kappa}} &= \sqrt{\kappa^2 + \acute{\kappa}^2 + 2\kappa\acute{\kappa}(\vec{\mu}\vec{\acute{\mu}})} \\
\tau_{\vec{\mu},\acute{\vec{\mu}}} &= \frac{\kappa\vec{\mu} + \acute{\kappa}\acute{\vec{\mu}}}{\xi_{\kappa,\acute{\kappa}}}
\end{aligned} \tag{2}$$

Using 2 and Langevin integral, we obtain:

$$\int_{\Omega} p(\vec{X}|\Theta)^{\rho} q(\vec{X}|\acute{\Theta})^{\rho} d\vec{X} = \Big[\Big(\frac{\kappa\acute{\kappa}}{4}\Big)^{\frac{p}{2}-1} \frac{1}{(2\pi)^{p} I_{\frac{p}{2}-1}(\kappa) I_{\frac{p}{2}-1}(\acute{\kappa})}\Big]^{\rho} \Big[\frac{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\xi_{\kappa,\acute{\kappa}}\rho)}{(\xi_{\kappa,\acute{\kappa}}\rho)^{\frac{p}{2}-1}}\Big] \tag{3}$$

APPENDIX B

Proof of Equation 28

The KL divergence between two exponential distributions is presented by [171]

$$KL(p(\vec{X}|\Theta), q(\vec{X}|\acute{\Theta})) = \Phi(\theta) - \Phi(\acute{\theta}) + [G(\theta) - G(\acute{\theta})]^T E_\theta[T(\vec{X})] \tag{1}$$

where E_θ is the expectation with respect to $p(\vec{X}|\Theta), G(\theta) = (G_1(\theta), \ldots, G_l(\theta)), T(\vec{X}) = (T_1(\vec{X}), \ldots, T_l(\vec{X}))$ where l is the number of parameters of the distribution and T denotes transpose. Furthermore, we have the following:

$$E_\theta[T(\vec{X})] = -\acute{\Phi}(\theta) \tag{2}$$

Then by letting $\Phi_\theta = -a_p(\kappa)$ and $G_\theta = \kappa\vec{\mu}$. Thus, the KL divergence between two Langevin distributions

$$KL(p(\vec{X}|\Theta), q(\vec{X}|\acute{\Theta})) = -\log \frac{\kappa^{\frac{p}{2}-1}}{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\kappa)} + \log \frac{\acute{\kappa}^{\frac{p}{2}-1}}{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\acute{\kappa})} + [\kappa\vec{\mu} - \acute{\kappa}\acute{\vec{\mu}}]^T \acute{a}_p(\kappa)\vec{\mu} \tag{3}$$

Appendix C

Proof of Equation 34

In the case of Langevin model, we can show the Shannon entropy is given by

$$\begin{aligned} H[p(\vec{X}|\theta)] &= -\int_{\Omega} p(\vec{X}|\theta) \log p(\vec{X}|\theta) d\vec{X} \\ &= -\int_{\Omega} p(\vec{X}|\theta) \Big[\sum_{p=1}^{P} \kappa \vec{\mu}^T X - a_p(\kappa)\Big] d\vec{X} \\ &= -\Big[E_\theta[\sum_{p=1}^{P} \kappa \vec{\mu}^T \vec{X}] - a_p(\kappa)\Big] \\ &= -\kappa a_p'(\kappa) \vec{\mu}^T \vec{\mu} + a_p(\kappa) \end{aligned} \tag{1}$$

Substitute Eq.1 into Eq. 33 we obtain the Jensen-Shannon divergence for Langevin model.

APPENDIX D

Proof of Equation 36

We can show that the Rényi divergence between two Langevin distribution, is given by

$$\begin{aligned}
&\int_{\Omega} p(\vec{X}|\Theta)^{\sigma} q(\vec{X}|\acute{\Theta})^{1-\sigma} d\vec{X} \qquad (1)\\
&= \left[\left(\frac{\kappa}{2}\right)^{\frac{p}{2}-1} \frac{1}{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\kappa)}\right]^{\sigma} \left[\left(\frac{\acute{\kappa}}{2}\right)^{\frac{p}{2}-1} \frac{1}{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\acute{\kappa})}\right]^{1-\sigma} \int_{\Omega} \left(e^{\kappa\vec{\mu}^{T}\vec{X}}\right)^{\sigma} \left(e^{\acute{\kappa}\acute{\vec{\mu}}^{T}\vec{X}}\right)^{1-\sigma} d\vec{X}\\
&= \left[\left(\frac{\kappa}{2}\right)^{\frac{p}{2}-1} \frac{1}{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\kappa)}\right]^{\sigma} \left[\left(\frac{\acute{\kappa}}{2}\right)^{\frac{p}{2}-1} \frac{1}{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\acute{\kappa})}\right]^{1-\sigma} \int_{\Omega} e^{(\kappa\vec{\mu}^{T}\vec{X}\sigma+\acute{\kappa}\acute{\vec{\mu}}^{T}\vec{X}(1-\sigma))} d\vec{X}
\end{aligned}$$

Assume that $\zeta_{\kappa,\acute{\kappa}} = \sqrt{(\sigma\kappa)^2 + ((1-\sigma)\acute{\kappa})^2 + 2\sigma\kappa(1-\sigma)\acute{\kappa}(\vec{\mu}.\acute{\vec{\mu}})}$ and $\psi_{\vec{\mu},\acute{\vec{\mu}}} = \frac{\sigma\kappa\vec{\mu}+(1-\sigma)\acute{\kappa}\acute{\vec{\mu}}}{\zeta_{\kappa,\acute{\kappa}}}$, and hence

$$\int_{\Omega} p(\vec{X})^{\sigma} q(\vec{X})^{1-\sigma} d\vec{X} = \left[\left(\frac{\kappa}{2}\right)^{\frac{p}{2}-1} \frac{1}{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\kappa)}\right]^{\sigma} \left[\left(\frac{\acute{\kappa}}{2}\right)^{\frac{p}{2}-1} \frac{1}{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\acute{\kappa})}\right]^{1-\sigma} \left[\frac{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\zeta_{\kappa,\acute{\kappa}})}{(\zeta_{\kappa,\acute{\kappa}})^{\frac{p}{2}-1}}\right] \qquad (2)$$

By substituting Eq. 2 in Eq. 35 we obtain the symmetric Rényi divergence for two Langevin distributions.

APPENDIX E

Proof of equation 15

In order to compute p_j, we introduce the Lagrange multiplier ξ to incorporate the constraint $\sum_{j=1}^{M} p_j = 1$. Assigning the derivative of 13 w.r.t p_j to zero, we obtain:

$$\frac{\partial S(\Theta, \mathcal{X})}{\partial p_j} = 0 \tag{1}$$

$$\frac{\partial S(\Theta, \mathcal{X})}{\partial p_j} = \Big[\sum_{i=1}^{N} \frac{\prod_{d=1}^{D}\Big(\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})\Big)}{\sum_{j=1}^{M} p_j \prod_{d=1}^{D}\Big(\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})\Big)}\Big] - \frac{D}{p_j} - \xi$$

Multiplying by p_j

$$\sum_{i=1}^{N} \hat{Z}_{ij} - D - p_j \xi = 0 \tag{2}$$

$$p_j = \frac{\sum_{i=1}^{N} \hat{Z}_{ij} - D}{\xi}$$

where

$$\hat{Z}_{ij} = \frac{p_j \prod_{d=1}^{D}\Big(\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})\Big)}{\sum_{j=1}^{M} p_j \prod_{d=1}^{D}\Big(\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})\Big)} \tag{3}$$

Computing the derivative w.r.t ξ,we obtain $1 - \sum_{j=1}^{M} p_j = 0$. Thus,

$$1 - \sum_{j=1}^{M} p_j = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \hat{Z}_{ij} - D}{\xi} \tag{4}$$

we obtain $\xi = \sum_{i=1}^{N} \sum_{j=1}^{M} \hat{Z}_{ij} - D$, since p_j are positive we can writs it:

$$p_j = \frac{\max(\sum_{i=1}^{N} \hat{Z}_{ij} - D, 0)}{\sum_{j=1}^{M} \max(\sum_{i=1}^{N} \hat{Z}_{ij} - D, 0)} \tag{5}$$

APPENDIX F

Proof of equation 16

Let $\rho_{jd} = \rho_{jd1}$ and $\rho_{jd2} = 1 - \rho_{jd}$. We introduce Lagrange multiplier ν_{jd} to incorporate the constraint $\rho_{jd1} + \rho_{jd2} = 1$, then we compute the derivative of Eq. 13 w.r.t ρ_{jd1}

$$\frac{\partial S(\Theta, \mathcal{X})}{\partial \rho_{jd1}} = 0 \tag{1}$$

$$\frac{\partial S(\Theta, \mathcal{X})}{\partial \rho_{jd1}} = \sum_{i=1}^{N}\sum_{j=1}^{M} \hat{Z}_{ij}\Big[\frac{p(\vec{Y}_{id}|\theta_{jd})}{\rho_{jd1}p(\vec{Y}_{id}|\theta_{jd}) + \rho_{jd2}p(\vec{Y}_{id}|\lambda_{jd})}\Big] - \frac{M}{\rho_{jd1}} - \nu_{jd}$$

Multiplying by ρ_{jd1}

$$\sum_{i=1}^{N} \hat{Z}_{ij}\Big[\frac{\rho_{jd1}p(\vec{Y}_{id}|\theta_{jd})}{\rho_{jd1}p(\vec{Y}_{id}|\theta_{jd}) + \rho_{jd2}p(\vec{Y}_{id}|\lambda_{jd})}\Big] - M - \rho_{jd1}\nu_{jd} = 0 \tag{2}$$

Now, we compute the derivative of Eq. 13 w.r.t ρ_{jd2}, we obtain

$$\frac{\partial S(\Theta, \mathcal{X})}{\partial \rho_{jd2}} = 0 \tag{3}$$

$$\frac{\partial S(\Theta, \mathcal{X})}{\partial \rho_{jd2}} = \sum_{i=1}^{N} \hat{Z}_{ij}\Big[\frac{p(\vec{Y}_{id}|\lambda_{jd})}{\rho_{jd1}p(\vec{Y}_{id}|\theta_{jd}) + \rho_{jd2}p(\vec{Y}_{id}|\lambda_{jd})}\Big] - \frac{M}{\rho_{jd2}} - \nu_{jd}$$

Multiplying by ρ_{jd2}

$$\sum_{i=1}^{N} \hat{Z}_{ij}\Big[\frac{\rho_{jd2}p(\vec{Y}_{id}|\lambda_{jd})}{\rho_{jd1}p(\vec{Y}_{id}|\theta_{jd}) + \rho_{jd2}p(\vec{Y}_{id}|\lambda_{jd})}\Big] - M - \rho_{jd2}\nu_{jd} = 0 \tag{4}$$

Similarly, we compute the derivative of Eq. 13 w.r.t ν_{jd}, we obtain

$$1 - \rho_{jd1} - \rho_{jd2} \;=\; 0 \tag{5}$$

Summing Eqs. 2 and 4, we obtain

$$\begin{aligned}\nu_{jd}(\rho_{jd2} + \rho_{jd1}) = \max\Big(&\sum_{i=1}^{N} \hat{Z}_{ij}\Big[\frac{\rho_{jd1} p(\vec{Y}_{id}|\theta_{jd})}{\rho_{jd1} p(\vec{Y}_{id}|\theta_{jd}) + \rho_{jd2} p(\vec{Y}_{id}|\lambda_{jd})}\Big] - M, 0\Big) \\ + \max\Big(&\sum_{i=1}^{N} \hat{Z}_{ij}\Big[\frac{\rho_{jd2} p(\vec{Y}_{id}|\lambda_{jd})}{\rho_{jd1} p(\vec{Y}_{id}|\theta_{jd}) + \rho_{jd2} p(\vec{Y}_{id}|\lambda_{jd})}\Big] - M, 0\Big)\end{aligned}$$

Thus, according to Eq. 2

$$\rho_{jd1} = \frac{\max\Big(\sum_{i=1}^{N} \hat{Z}_{ij}\Big[\frac{\rho_{jd1} p(\vec{Y}_{id}|\theta_{jd})}{\rho_{jd1} p(\vec{Y}_{id}|\theta_{jd}) + \rho_{jd2} p(\vec{Y}_{id}|\lambda_{jd})}\Big] - M, 0\Big)}{\nu_{jd}} \tag{6}$$

then, according to Eq. 4

$$\rho_{jd2} = \frac{\max\Big(\sum_{i=1}^{N} \hat{Z}_{ij}\Big[\frac{\rho_{jd2} p(\vec{Y}_{id}|\lambda_{jd})}{\rho_{jd1} p(\vec{Y}_{id}|\theta_{jd}) + \rho_{jd2} p(\vec{Y}_{id}|\lambda_{jd})}\Big] - M, 0\Big)}{\nu_{jd}} \tag{7}$$

APPENDIX G

Proof of equation 17

Compute the derivative of Eq. 13 w.r.t μ_{jd}

$$\begin{aligned}
\frac{\partial S(\Theta,\mathcal{X})}{\partial \mu_{jd}} &= 0 \\
\frac{\partial S(\Theta,\mathcal{X})}{\partial \mu_{jd}} &= \sum_{i=1}^{N} \hat{Z}_{ij} \frac{\partial}{\partial \mu_{jd}} \log\Big[\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})\Big] \\
&= \sum_{i=1}^{N} \hat{Z}_{ij} \frac{\frac{\partial(\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd}))}{\partial \mu_{jd}}}{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})}
\end{aligned}$$

We have

$$\begin{aligned}
\frac{\partial}{\partial \mu_{jd}} p(\vec{Y}_{id}|\theta_{jd}) &= \frac{\partial}{\partial \mu_{jd}} \Big[\kappa_{jd} \vec{\mu}_{jd} \vec{Y}_{id} - \log(2\pi I_0(\kappa_{jd})) + \alpha_{jd}(1 - \vec{\mu}_{jd}^T \vec{\mu}_{jd})\Big] \\
&= \kappa_{jd} \vec{Y}_{id} - 2\alpha_{jd} \vec{\mu}_{jd}
\end{aligned}$$

where α_{jd} is a Lagrange multiplier to correspond the constraint $\vec{\mu}_{jd}^T \vec{\mu}_{jd} = 1$. Derive w.r.t α_{jd}

$$\begin{aligned}
\frac{\partial}{\partial \alpha_{jd}} p(\vec{Y}_{id}|\theta_{jd}) &= \frac{\partial}{\partial \alpha_{jd}} \Big[\kappa_{jd} \vec{\mu}_{jd} \vec{Y}_{id} - \log(2\pi I_0(\kappa_{jd})) + \alpha_{jd}(1 - \vec{\mu}_{jd}^T \vec{\mu}_{jd})\Big] \\
&= 1 - \vec{\mu}_{jd}^T \vec{\mu}_{jd}
\end{aligned}$$

Thus,

$$\vec{\mu}_{jd} = \frac{\sum_{i=1}^{N} \hat{Z}_{ij} \frac{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) \vec{Y}_{id}}{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})}}{\sum_{i=1}^{N} \hat{Z}_{ij} \frac{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd})}{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})}} \tag{1}$$

And

$$\vec{\mu}_{jd} = \frac{\vec{\mu}_{jd}}{\|\vec{\mu}_{jd}\|} \tag{2}$$

APPENDIX H

Proof of equation 20

Compute the derivative of Eq. 13 w.r.t κ_{jd} given that $\kappa_{jd} \geq 0$

$$\begin{aligned}
\frac{\partial S(\Theta, \mathcal{X})}{\partial \kappa_{jd}} &= 0 \\
\frac{\partial S(\Theta, \mathcal{X})}{\partial \kappa_{jd}} &= \sum_{i=1}^{N} \hat{Z}_{ij} \frac{\partial}{\partial \kappa_{jd}} \log\Big[\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})\Big] \\
&= \sum_{i=1}^{N} \hat{Z}_{ij} \frac{\frac{\partial(\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd}))}{\partial \kappa_{jd}}}{\rho_{jd} p(\vec{Y}_{id}|\theta_{jd}) + (1-\rho_{jd}) p(\vec{Y}_{id}|\lambda_{jd})}
\end{aligned}$$

We have

$$\begin{aligned}
\frac{\partial}{\partial \kappa_{jd}} p(\vec{Y}_{id}|\theta_{jd}) &= \frac{\partial}{\partial \kappa_{jd}} \Big[-\log(2\pi I_0(\kappa_{jd})) + \kappa_{jd} \vec{\mu}_{jd} \vec{Y}_{id}\Big] \\
&= \vec{\mu}_{jd}^T \vec{Y}_{id} - \frac{I_1(\kappa_{jd})}{I_0(\kappa_{jd})}
\end{aligned} \tag{1}$$

Assume that $A(\kappa_{jd}) = \frac{I_1(\kappa_{jd})}{I_0(\kappa_{jd})}$. Since it is hard to find tractable forum of $A^{-1}(\kappa_{jd})$ so we need to do some approximation. Note that we can use Newton-Raphson iterations to find κ_{jd}, where:

$$\kappa_{jd}^{new} = \kappa_{jd}^{old} - \frac{\partial S(\mathcal{X}|\Theta)}{\partial \kappa_{jd}} \Big(\frac{\partial^2 S(\mathcal{X}|\Theta)}{\partial^2 \kappa_{jd}}\Big)^{-1} \tag{2}$$

List of References

[1] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, “Clustering on the unit hypersphere using von mises-fisher distributions,” *Journal of Machine Learning Research*, vol. 6, pp. 1345 – 1382, Septmber 2005.

[2] O. Amayri and N. Bouguila, “Probabilistic clustering based on langevin mixture,” in *Proceedings of the 10th International Conference on Machine Learning and Applications*, vol. 2, pp. 388–391, IEEE Computer Society, 2011.

[3] O. Amayri and N. Bouguila, “Beyond hybrid generative discriminative learning: Spherical data classification,” *Pattern Analysis and Applications*, pp. 1–21, 2013.

[4] G. Salton and M. J. McGill, *Introduction to Modern Retrieval*. McGraw-Hill Book Company, 1983.

[5] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, “Generative model-based clustering of directional data,” in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 19–28, 2003.

[6] A. Banerjee and S. Basu, “Topic models over text streams: A study of batch and online unsupervised learning,” in *Proceedings of the SIAM International Conference on Data Mining*, pp. 437–442, 2007.

[7] Q. He, K. Chang, E.-P. Lim, and A. Banerjee, “Keep it simple with time: A re-examination of probabilistic topic detection models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1795–1808, 2010.

[8] O. Amayri and N. Bouguila, “Online news topic detection and tracking via localized feature selection,” in *Proceedings of the 2013 International Joint Conference on Neural Networks*, pp. 1–8, 2013.

[9] G. S. Watson and E. J. Williams, “On the construction of significance tests on the circle and the sphere,” *Biometrika*, vol. 43, no. 3/4, pp. 344–352, 1956.

[10] G. S. Watson, “Analysis of dispersion on a sphere,” *International Geophysical Journal*, vol. 7, no. 4, pp. 153–159, 1956.

[11] G. S. Watson, “More significance tests on the sphere,” *Biometrika*, vol. 47, no. 1/2, pp. 87–91, 1960.

[12] M. A. Stephens, “Exact and approximate tests for directions. i,” *Biometrika*, vol. 49, no. 3/4, pp. 463–477, 1962.

[13] M. A. Stephens, “Exact and approximate tests for directions. ii,” *Biometrika*, vol. 49, no. 3/4, pp. 547–552, 1962.

[14] M. A. Stephens, “Tests for the von mises distribution,” *Biometrika*, vol. 56, no. 1, pp. 149–160, 1969.

[15] S. Coles, “Inference for circular distributions and processes,” *Statistics and Computing*, vol. 8, no. 2, pp. 105–113, 1998.

[16] A. L. Rukhin, “Some statistical decisions about distributions on a circle for large samples,” *Sankhya: The Indian Journal of Statistics, Series A*, vol. 34, no. 3, pp. 243–250, 1972.

[17] K. V. Mardia and B. D. Spurr, "Multisample tests for multimodal and axial circular populations," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 35, no. 3, pp. 422–436, 1973.

[18] J. E. Morris and P. J. Laycock, "Discriminant analysis of directional data," *Biometrika*, vol. 61, no. 2, pp. 335–341, 1974.

[19] P. E. Jupp and K. V. Mardia, "A general correlation coefficient for directional data and related regression problems," *Biometrika*, vol. 67, no. 1, pp. 163–173, 1980.

[20] E. Leopold and J. Kindermann, "Text categorization with support vector machines. how to represent texts in input space?," *Machine Learning*, vol. 46, no. 13, pp. 423 – 444, 2002.

[21] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proceedings of 10th European Conference on Machine Learning* (C. Nédellec and C. Rouveirol, eds.), no. 1398, (Chemnitz, DE), pp. 137 – 142, Springer-Verlag, 1998.

[22] G. Wittel and S. Wu, "On attacking statistical spam filters," in *Proceedings of the First Conference on Email and Anti-Spam*, (California, USA), 2004.

[23] K. V. Mardia, "Statistics of directional data (with discussions)," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 37, no. 3, pp. 349–393, 1975.

[24] K. V. Mardia, *Statistics of directional data*. Academic Press, 1972.

[25] G. S. Watson, *Statistics on Spheres*. Wiley: New York, 1983a.

[26] N. I. Fisher, *Statistical analysis of circular data*. Cambridge, United Kingdom: Cambridge University Press, 1st ed., 1993.

[27] N. I. Fisher, B. J. J. Embleton, and T. Lewis, *Statistical analysis of spherical data*. Cambridge University Press, 1993.

[28] T. McGraw, B. Vemuri, B. Yezierski, and T. Mareci, "von Mises-Fisher mixture model of the diffusion ODF," in *Proceedings of 3rd IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, (Arlington, VA), pp. 65 – 68, 2006.

[29] H. Tang, S. M. Chu, and T. S. Huang, "Generative model-based speaker clustering via mixture of von mises-fisher distributions," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Los Alamitos, CA, USA), pp. 4101– 4104, IEEE Computer Society, 2009.

[30] K. V. Mardia and P. J. Zemroch, "Algorithm as 81: Circular statistics," *Applied Statistics*, vol. 24, no. 1, pp. 147–150, 1975.

[31] K. V. Mardia and P. J. Zemroch, "Algorithm as 80: Spherical statistics," *Applied Statistics*, vol. 24, no. 1, pp. 144–146, 1975.

[32] G.J. McLachlan and D. Peel, *Finite Mixture Models*. New York: Wiley, 2000.

[33] O. Amayri and N. Bouguila, "Unsupervised feature selection for spherical data modeling: Application to image-based spam filtering," in *Proceedings of Multimedia Communications, Services and Security*, pp. 13–23, Springer, Heidelberg, 2012.

[34] O. Amayri and N. Bouguila, "On online high-dimensional spherical data clustering and feature selection," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 4, pp. 1386 – 1398, 2013.

[35] O. Amayri and N. Bouguila, "A bayesian analysis of spherical pattern based on langevin mixture," *Journal*, 2014. submitted.

[36] O. Amayri and N. Bouguila, "Simultaneous bayesian analysis and feature selection of langevin mixture," *Journal*, 2014. submitted.

[37] O. Amayri and N. Bouguila, “On nonparametric bayesian clustering and feature selection for high dimensional data and applications,” *Journal*, 2014. submitted.

[38] I. T. Podolak and A. Roman, “Cores: fusion of supervised and unsupervised training methods for a multi-class classification problem,” *Pattern Analysis and Applications*, vol. 14, no. 4, pp. 395–413, 2011.

[39] B. Yang, S. Chen and X. Wu, “A structurally motivated framework for discriminant analysis,” *Pattern Analysis and Applications*, vol. 14, no. 4, pp. 349–367, 2011.

[40] V. N. Vapnik, *The nature of statistical learning theory*. Springer-Verlag, 2000.

[41] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[42] T. Jebara, R. Kondor, and A. Howard, “Probability product kernels,” *Journal of Machine Learning Research*, vol. 5, pp. 819–844, 2004.

[43] A. Y. Ng and M. I. Jordan, “On discriminative vs generative classifiers: A comparison of logistic regression and naive bayes,” in *Proceedings of 4th Conference on Advances in Neural Information Processing Systems*, pp. 841 – 848, MIT Press, December 2001.

[44] R. Raina, Y. Shen, A. Y. Ng, and A. McCallum, “Classification with hybrid generative/discriminative models,” in *Proceedings of 16th Conference on Advances in Neural Information Processing Systems*, MIT Press, December 2003.

[45] A. Bosch, A. Zisserman, and X. M. Noz, “Scene classification using a hybrid generative/ discriminative approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 712 – 727, 2008.

[46] L. Prevost, L. Oudot, A. Moises, C. Michel-Sendis, and M. Milgram, “Hybrid generative/discriminative classifier for unconstrained character recognition,” *Pattern Recognition Letters*, vol. 26, pp. 1840 – 1848, September 2005.

[47] R. Herbrich and T. Graepel, “A PAC-Bayesian Margin Bound for Linear Classifiers: Why SVMs Work,” in *Proceedings of Advances in Neural Information Processing Systems*, pp. 224–230, 2000.

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

[49] A. B. Graf, A. J. Smola, and S. Borer, “Classification in a normalized feature space uisng support vector machines,” *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 597–605, 2003.

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

[51] C. S. Wallace and D. L. Dowe, “MML clustering of multi-state, Poisson, von Mises circular and Gaussian distributions,” *Statistics and Computing*, vol. 10, no. 1, pp. 73–83, 2000.

[52] D. Dowe, J. Oliver, and C. Wallace, “MML estimation of the parameters of the spherical fisher distribution,” in *Proceedings of the conference on Algorithmic Learning Theory* (S. Arikawa and A. Sharma, eds.), vol. 1160 of *Lecture Notes in Computer Science*, pp. 213–227, Springer Berlin / Heidelberg, 1996.

[53] I. S. Dhillon and D. S. Modha, “Concept Decompositions for Large Sparse Text Data Using Clustering,” *Machine Learning*, vol. 42, no. 1-2, pp. 143–175, 2001.

[54] I. Dhillon, J. Fan, and Y. Guan, *Efficient Clustering of Very Large Document Collections*. Kluwer Academic Publishers, 2001.

[55] H. Akaike, “A new look at the statistical model identification,” *IEEE Transaction on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[56] G. Schwarz, "Estimating dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1978.

[57] J. Rissanen, "Modeling by shortest data discription," *Automatica*, vol. 14, pp. 465–471, 1987.

[58] J. A. Mooney, P. J. Helms, and I. T. Jolliffe, "Fitting mixtures of von mises distributions: A case study involving sudden infant death syndrome," *Computational Statistics and Data Analysis*, vol. 41, pp. 505–513, 2003.

[59] N. Bouguila and D. Ziou, "Unsupervised Selection of a Finite Dirichlet Mixture Model: An MML-Based Approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 8, pp. 993–1009, 2006.

[60] N. Bouguila and D. Ziou, "High-dimensional unsupervised selection and estimation of a finite generalized dirichlet mixture model based on minimum message length," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1716–1731, 2007.

[61] K. V. Mardia, "Distribution theory for the von mises-fisher distribution and its application," in *Statistical Distributions for Scientific Work* (S. Kotz, G. P. Patial, and J. K. ord, eds.), vol. 1, pp. 113–130, 1975.

[62] A. Agarwal and H. Daumé, "Generative kernels for exponential families," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2011.

[63] T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Proceedings of Advances in Neural Information Systems*, pp. 487 – 493, MIT Press, 1998.

[64] A. B. Chan, N. Vasconcelos, and P. J. Moreno, "A family of probabilistic kernels based on information divergence," TechnicalReport SVCL-TR2004/01, University of California, SanDiego, 2004.

[65] N. Bouguila, "Hybrid generative/discriminative approaches for proportional data modeling and classification," *IEEE Transactions on Knowledge and Data Engineering*, 2011.

[66] S. Kullback, *Information Theory and Statistics*. Wiley, 1959.

[67] P. J. Moreno, P. P. Ho, and N. Vasconcelos, "A kullback-leibler divergence based kernel for svm classification in multimedia applications," in *Proceedings of Advances in Neural Information Processing Systems*, MITPress, 2003.

[68] J. Hershey and P. Olsen, "Approximating the kullback leibler divergence between gaussian mixture models," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. 317–320, April 2007.

[69] J. Lin, "Divergence measure based on shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 14, pp. 145–151, 1991.

[70] A. Rényi, "On measures of entropy and information," in *Proceedings of Berkeley Symposium Mathmetical Statistics and Probability*, pp. 547–561, 1960.

[71] G. Ulrich, "Computer generation of distributions on the m-sphere," *Journal of the Royal Statistical Society*, vol. 33, no. 2, pp. 158–163, 1984.

[72] A. T. A. Wood, "Simulation of the von mises fisher distribution," *Communications in Statistics Simulation and Computation*, vol. 23, no. 1, pp. 157–164, 1994.

[73] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artif. Intell. Rev.*, vol. 29, pp. 63–92, 2008.

[74] Y. Zhu and Y. Tan, "A local-concentration-based feature extraction approach for spam filtering," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 486 –497, 2011.

[75] L. Özgür and T. Güngör, “Optimization of dependency and pruning usage in text classification,” *Pattern Analysis and Applications*, vol. 15, no. 1, pp. 45–58, 2012.

[76] G. V. Cormack and T. R. Lynam, “Online supervised spam filter evaluation,” *ACM Transactions on Information Systems*, vol. 25, pp. 1–29, 2007.

[77] S. Hershkop and S. J. Stolfo, “Combining email models for false positive reduction,” in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 98–107, 2005.

[78] M. Chang, W. Yih, and C. Meek, “Partitioned logistic regression for spam filtering,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 97–105, 2008.

[79] K. Yoshida, F. Adachi, T. Washio, H. Motoda, T. Homma, A. Nakashima, H. Fujikawa, and K. Yamazaki, “Density-based spam detector,” in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 486–493, 2004.

[80] P. Chirita, J. Diederich, and W. Nejdl, “Mailrank: using ranking for spam detection,” in *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 373–380, 2005.

[81] C. Tseng, J. Huang, and M. Chen, “Promail: Using progressive email social network for spam detection,” in *PAKDD* (Z. Zhou, H. Li, and Q. Yang, eds.), vol. 4426 of *Lecture Notes in Computer Science*, pp. 833–840, Springer, 2007.

[82] C. H. Wu, “Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks,” *Expert System Application*, vol. 36, no. 3, pp. 4321 – 4330, 2009.

[83] G. Fumera, I. Pillai, and F. Roli, “Spam filtering based on the analysis of text information embedded into images,” *Journal of Machine Learning Research*, vol. 7, pp. 2699–2720, 2006.

[84] K. Konstantinidis, V. Vonikakis, G. Panitsidis and I. Andreadis, “A Center-Surround Histogram for content-based image retrieval,” *Pattern Analysis and Applications*, vol. 14, no. 3, pp. 251–260, 2011.

[85] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[86] I. Androutsopoulos, J. Koutsias, K. V. Cb, and C. D. Spyropoulos, “An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages,” in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 160–167, ACM Press, 2000.

[87] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, “A bayesian approach to filtering junk e-mail,” in *Proceedings of National Conference on Artificial Intelligence*, 1998.

[88] G. V. Cormack and T. R. Lynam, “Trec 2005 spam track overview,” in *Proceedings of the Fourteenth Text REtrieval Conference*, (Gaithersburg MD), 2005.

[89] B. Mehta, S. Nangia, M. Gupta, and W. Nejdl, “Detecting image spam using visual features and near duplicate detection,” in *Proceedings of the 17th international conference on World Wide Web*, pp. 497–506, 2008.

[90] M. Dredze, R. Gevaryahu, and A. Elias-Bachrach, “Learning fast classifiers for image spam,” in *Proceedings of the 4th Conference on Email and Anti-Spam*, pp. 487–493, 2007.

[91] M. C. Burl, L. Asker, P. Smyth, U. Fayyad, P. Perona, L. Crumpler and J. Aubele, “Learning to Recognize Volcanoes on Venus,” *Machine Learning*, vol. 30, no. 2-3, pp. 165–194, 1998.

[92] H. Liu and H. Motoda, *Computational Methods of Feature Selection*. Data Mining and Knowledge Discovery, Chapman & Hall/CRC, 2008.

[93] N. Bouguila, “On multivariate binary data clustering and feature weighting,” *Computational Statistics and Data Analysis*, vol. 54, no. 1, pp. 120 – 134, 2010.

[94] M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain, “Simultaneous feature selection and clustering using mixture models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1154–1166, 2004.

[95] P. Mitra, C. Murthy, and S. Pal, “Unsupervised feature selection using feature similarity,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 301 –312, march 2002.

[96] A. Jain and D. Zongker, “Feature selection: evaluation, application, and small sample performance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 153–158, february 1997.

[97] A. Lemos, W. Caminhas, and F. Gomide, “Evolving fuzzy linear regression trees with feature selection,” in *IEEE Workshop on Evolving and Adaptive Intelligent Systems*, pp. 31–38, IEEE Press, 2011.

[98] Y. Saeys, I. n. Inza, and P. Larrañaga, “A review of feature selection techniques in bioinformatics,” *Bioinformatics*, vol. 23, pp. 2507–2517, october 2007.

[99] J. Zhang and K. W. Chau, “Multilayer ensemble pruning via novel multi-sub-swarm particle swarm optimization,” *Journal of Universal Computer Science*, vol. 15, no. 4, pp. 840–858, 2009.

[100] M. Allili, D. Ziou, N. Bouguila, and S. Boutemedjet, “Unsupervised feature selection and learning for image segmentation,” in *Proceedings of the Canadian Conference on Computer and Robot Vision*, pp. 285–292, IEEE Press, 2010.

[101] 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*, vol. 31, pp. 1429 –1443, August 2009.

[102] N. Bouguila, “A model-based approach for discrete data clustering and feature weighting using map and stochastic complexity,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, pp. 1649 –1664, december 2009.

[103] K. V. Mardia and T. W. Sutton, “On the Modes of a Mixture of two von Mises Distributions,” *Biometrika*, vol. 62, no. 3, pp. 699–701, 1975.

[104] J. T. Kent, “Identifiability of Finite Mixtures for Directional Data,” *The Annals of Statistics*, vol. 11, no. 3, pp. 984–988, 1983.

[105] Y. Fu, J. Chen, and P. Li, “Modified likelihood ratio test for homogeneity in a mixture of von mises distributions,” *Journal of Statistical Planning and Inference*, vol. 138, no. 3, pp. 667 – 681, 2008.

[106] S. Calderara, A. Prati, and R. Cucchiara, “Mixtures of von mises distributions for people trajectory shape analysis,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, pp. 457 –471, April 2011.

[107] C. Grana, D. Borghesani, and R. Cucchiara, “Describing texture directions with von mises distributions,” in *Proceedings Of the 19th International Conference on Pattern Recognition*, pp. 1–4, IEEE Press, 2008.

[108] Y. Agiomyrgiannakis and Y. Stylianou, “Stochastic modeling and quantization of harmonic phases in speech using wrapped gaussian mixture models,” in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. 1121–1124, 2007.

[109] S. Perkins and J. Theiler, "Online feature selection using grafting," in *Proceedings Of 20th International Conference on Machine Learning*, pp. 592–599, AAAI Press, 2003.

[110] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks* (D. Saad, ed.), Cambridge, UK: Cambridge University Press, 1998.

[111] N. Bouguila and D. Ziou, "Online clustering via finite mixtures of dirichlet and minimum message length," *Engineering Applications of Artificial Intelligence*, vol. 19, pp. 371–379, June 2006.

[112] J. Zhang, Z. Ghahramani, and Y. Yang, "A probabilistic model for online document clustering with application to novelty detection," in *Advances in Neural Information Processing Systems 17* (L. K. Saul, Y. Weiss, and L. Bottou, eds.), pp. 1617–1624, Cambridge, MA: MIT Press, 2005.

[113] J. Beringer and E. Hullermeier, "Online clustering of parallel data streams," *Data & Knowledge Engineering*, vol. 58, no. 2, pp. 180 – 204, 2006.

[114] O. Amayri and N. Bouguila, "Online spam filtering using support vector machines," in *Proceedings of IEEE Symposium on Computers and Communications*, pp. 337–340, IEEE Press, 2009.

[115] M-A. Sato and S. Ishii, "On-line EM Algorithm for the Normalized Gaussian Network," *Neural Computation*, vol. 12, no. 2, pp. 407–432, 2000.

[116] A. Banerjee and J. Ghosh, "Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres," *IEEE Transactions on Neural Networks*, vol. 15, pp. 702–719, may 2004.

[117] J. F. Yao, "On recursive estimation in incomplete data models," *Statistics*, vol. 34, no. 1, pp. 27–51, 2000.

[118] J. Kivinen, A. J. Smola, and R. C. Williamson, “Online learning with kernels,” *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2165–2176, 2004.

[119] J. Ratsaby, “A stochastic gradient descent algorithm for structural risk minimisation,” in *Algorithmic Learning Theory* (R. Gavalda, K. Jantke, and E. Takimoto, eds.), vol. 2842 of *Lecture Notes in Computer Science*, pp. 205–220, Springer Berlin, Heidelberg, 2003.

[120] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proceedings of the 19th International Conference on Computational Statistics* (Y. Lechevallier and G. Saporta, eds.), (Paris, France), pp. 177–187, Springer, August 2010.

[121] A. Samé, G. Govaert, and C. Ambroise, “A mixture model-based on-line cem algorithm,” in *Proceedings of the 6th international conference on Advances in Intelligent Data Analysis*, (Berlin, Heidelberg), pp. 373–384, Springer-Verlag, 2005.

[122] E. J. Gumbel, J. A. Greenwood and D. Durand, “The Circular Normal Distribution: Theory and Tables,” *Journal of the American Statistical Association*, vol. 48, no. 261, pp. 131–152, 1953.

[123] D. L. Dowe, J. J. Oliver, R. A. Baxter, and C. S. Wallace, “Bayesian estimation of the von mises concentration parameter,” in *Proceedings Of the 15th International Workshop On Maximum Entropy And Bayesian Methods*, pp. 51–59, Kluwer Academic, 1995.

[124] R. Fisher, “Dispersion on a Sphere,” *Proceedings of the Royal Society of London. Series A, Mathematical and Physical*, vol. 217, no. 1130, pp. 295–305, 1953.

[125] E. Breitenberger, “Analogues of the Normal Distribution on the Circle and the Sphere,” *Biometrika*, vol. 50, no. 1/2, pp. 81–88, 1963.

[126] R. Baxter and J. Oliver, “Finding Overlapping Components with MML,” *Statistics and Computing*, vol. 10, no. 1, pp. 5–16, 2000.

[127] C. S. Wallace and D. L. Dowe, “MML Estimation of the von mises Concentration Parameter,” Technical Report TR 193, Monash University, 1993.

[128] T. S. Guzella and W. M. Caminhas, “A review of machine learning approaches to spam filtering,” *Expert Systems with Applications*, vol. 36, no. 7, pp. 10206 – 10222, 2009.

[129] Q. Liu, Z. Qin, H. Cheng, and M. Wan, “Efficient modeling of spam images,” in *Proceedings of the 3rd International Symposium on Intelligent Information Technology and Security Informatics*, pp. 663–666, 2010.

[130] J.-H. Hsia and M.-S. Chen, “Language-model-based detection cascade for efficient classification of image-based spam e-mail,” in *Proceedings of the 2009 IEEE international conference on Multimedia and Expo*, pp. 1182–1185, IEEE Press, 2009.

[131] F. Perronnin and C. R. Dance, “Fisher kernels on visual vocabularies for image categorization,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE Computer Society, 2007.

[132] O. Amayri and N. Bouguila, “Improved online support vector machines spam filtering using string kernels,” in *Proceedings of the 14th Iberoamerican Congress on Pattern Recognition*, vol. 5856 of *Lecture Notes in Computer Science*, pp. 621–628, Springer, 2009.

[133] K. F. Chen, *Offline and Online SVM performance Analysis*. Thesis, Massachusetts Institute of Technology, Febreuary 2007.

[134] N. Bouguila, “Bayesian hybrid generative discriminative learning based on finite liouville mixture models,” *Pattern Recognition*, vol. 44, no. 6, pp. 1183–1200, 2011.

[135] A. Strehl, E. Strehl, J. Ghosh, and R. Mooney, “Impact of similarity measures on web-page clustering,” in *Proceedings of the 17th National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web Search*, pp. 58–64, AAAI Press, 2000.

[136] C. P. Robert and G. Casella, *Monte Carlo statistical methods*, vol. 319. Citeseer, 2004.

[137] J.-M. Marin and C. P. Robert, *Bayesian core: a practical approach to computational Bayesian statistics*. Springer, 2007.

[138] C. Robert, *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer, 2007.

[139] K. Mardia and S. El-Atoum, "Bayesian inference for the von mises-fisher distribution," *Biometrika*, vol. 63, no. 1, pp. 203–206, 1976.

[140] P. Guttorp and R. A. Lockhart, "Finding the location of a signal: A bayesian analysis," *Journal of the American Statistical Association*, vol. 83, no. 402, pp. 322–330, 1988.

[141] P. Damien and S. Walker, "A full bayesian analysis of circular data using the von mises distribution," *Canadian Journal of Statistics*, vol. 27, no. 2, pp. 291–298, 1999.

[142] G. Nunez-Antonio and E. Gutiérrez-Pena, "A bayesian analysis of directional data using the von mises–fisher distribution," *Communications in StatisticsSimulation and Computation*, vol. 34, no. 4, pp. 989–999, 2005.

[143] R. E. Kass and A. E. Raftery, "Bayes factors," *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 773–795, 1995.

[144] N. Bouguila, D. Ziou, and R. I. Hammoud, "On bayesian analysis of a finite generalized dirichlet mixture via a metropolis-within-gibbs sampling," *Pattern Analysis Application*, vol. 12, no. 2, pp. 151–166, 2009.

[145] L. D. Brown, *Fundamentals of statistical exponential families: with applications in statistical decision theory*. Ims, 1986.

[146] K. Hornik and B. Grün, "On conjugate families and jeffreys priors for von mises-fisher distributions," *Journal of Statistical Planning and Inference*, vol. 143, no. 5, pp. 992 – 999, 2013.

[147] S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *The American Statistician*, vol. 49, no. 4, pp. 327–335, 1995.

[148] S. M. Lewis and A. E. Raftery, "Estimating bayes factors via posterior simulation with the laplacemetropolis estimator," *Journal of the American Statistical Association*, vol. 92, no. 438, pp. 648–655, 1997.

[149] C. P. Robert, "Convergence control methods for markov chain monte carlo algorithms," *Statistical Science*, pp. 231–253, 1995.

[150] A. E. Raftery and S. M. Lewis, "[practical markov chain monte carlo]: Comment: One long run with diagnostics: Implementation strategies for markov chain monte carlo," *Statistical Science*, vol. 7, no. 4, pp. 493–497, 1992.

[151] S. Richardson and P. J. Green, "On bayesian analysis of mixtures with an unknown number of components (with discussion)," *Journal of the Royal Statistical Society: series B (statistical methodology)*, vol. 59, no. 4, pp. 731–792, 1997.

[152] W. R. Gilks and P. Wild, "Adaptive rejection sampling for gibbs sampling," *Applied Statistics*, vol. 42, no. 4, pp. 701–709, 1993.

[153] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1475–1490, November 2004.

[154] W.-S. Zheng, S. Gong, and T. Xiang, "Quantifying and transferring contextual information in object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 762–777, 2012.

[155] B. W. Mel, "Seemore: combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition," *Neural Computation*, vol. 9, no. 4, pp. 777–804, 1997.

[156] S. Maji and A. C. Berg, "Max-margin additive classifiers for detection," in *Proceedings of the IEEE 12th International Conference on Computer Vision*, pp. 40–47, 2009.

[157] F. Perronnin, J. Snchez, and Y. Liu, "Large-scale image categorization with explicit data embedding," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2297–2304, 2010.

[158] G. Heitz and D. Koller, "Learning spatial context: Using stuff to find things," in *Computer Vision–ECCV 2008*, pp. 30–43, Springer, 2008.

[159] M. Everingham, A. Zisserman, C. K. Williams, L. Van Gool, M. Allan, C. M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorkó, *et al.*, "The 2005 pascal visual object classes challenge," in *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pp. 117–176, Springer, 2006.

[160] J. Allan, V. Lavrenko, and H. Jin, "First story detection in tdt is hard," in *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 374–381, 2000.

[161] J. Allan, S. Harding, D. Fisher, A. Bolivar, S. Guzman-Lara, and P. Amstutz, "Taking topic detection from evaluation to practice," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pp. 101–110, IEEE Computer Society, 2005.

[162] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic detection and tracking pilot study final report," in *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[163] J. M. Kleinberg, “Bursty and hierarchical structure in streams,” *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 373–397, 2003.

[164] D. Cai, X. He, and J. Han, “Document clustering using locality preserving indexing,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1624–1637, 2005.

[165] D. Cai, X. He, J. Han, and T. S. Huang, “Graph regularized nonnegative matrix factorization for data representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.

[166] R. M. Neal, “Markov chain sampling methods for dirichlet process mixture models,” *Journal of computational and graphical statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[167] Q. He, K. Chang, and E.-P. Lim, “Using burstiness to improve clustering of topics in news streams,” in *Proc. of the 7th IEEE International Conference on Data Mining (ICDM)*, pp. 493–498, 2007.

[168] C. E. Rasmussen, “The infinite gaussian mixture model.,” in *NIPS*, vol. 12, pp. 554–560, 1999.

[169] J. J. Kivinen, E. B. Sudderth, and M. I. Jordan, “Learning multiscale representations of natural scenes using dirichlet processes,” in *Proceedings of the 11th International Conference on Computer Vision*, pp. 1–8, IEEE, 2007.

[170] A. Cardoso-Cachopo, “Improving Methods for Single-label Text Categorization.” PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2007.

[171] T. Kailath, “The divergence and bhattacharyya distance measures in signal selection,” *IEEE Transactions on Communications Technology*, vol. 15, no. 1, pp. 52–60, 1967.