VARIATIONAL APPROACHES FOR LEARNING FINITE SCALED DIRICHLET MIXTURE MODELS

DINH HIEU NGUYEN

A THESIS
IN
THE CONCORDIA INSTITUTE FOR INFORMATION SYSTEMS ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF APPLIED SCIENCE
(QUALITY SYSTEMS ENGINEERING)
CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

MAY 2019

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Dinh Hieu Nguyen**
Entitled: **Variational Approaches for Learning Finite Scaled Dirichlet Mixture Models**
and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science
(Quality Systems Engineering)

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

Signed by the final examining commitee:

Dr. Jun Yan ____________________ Chair

Dr. Nizar Bouguila ____________________ Supervisor

Dr. Walter Lucia ____________________ CIISE Examiner

Dr. Bruno Lee____________________ External Examiner

Approved ____________________

Dr. Chadi Assi Graduate Program Director

2019.05.27 ____________________

Dr. Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Variational Approaches for Learning Finite Scaled Dirichlet Mixture Models

Dinh Hieu Nguyen

With a massive amount of data created on a daily basis, the ubiquitous demand for data analysis is undisputed. Recent development of technology has made machine learning techniques applicable to various problems. Particularly, we emphasize on cluster analysis, an important aspect of data analysis. Recent works with excellent results on the aforementioned task using finite mixture models have motivated us to further explore their extents with different applications. In other words, the main idea of mixture model is that the observations are generated from a mixture of components, in each of which the probability distribution should provide strong flexibility in order to fit numerous types of data. Indeed, the Dirichlet family of distributions has been known to achieve better clustering performances than those of Gaussian when the data are clearly non-Gaussian, especially proportional data. Thus, we introduce several variational approaches for finite Scaled Dirichlet mixture models. The proposed algorithms guarantee reaching convergence while avoiding the computational complexity of conventional Bayesian inference. In summary, our contributions are threefold. First, we propose a variational Bayesian learning framework for finite Scaled Dirichlet mixture models, in which the parameters and complexity of the models are naturally estimated through the process of minimizing the Kullback-Leibler (KL) divergence between the approximated posterior distribution and the true one. Secondly, we integrate component splitting into the first model, a local model selection scheme, which gradually splits the components based on their mixing weights to obtain the optimal number of components. Finally, an online variational inference framework for finite Scaled Dirichlet mixture models is developed by employing a stochastic approximation method in order to improve the scalability of finite mixture models for handling large scale data in real time. The effectiveness of our models is validated with real-life challenging problems including object, texture, and scene categorization, text-based and image-based spam email detection.

Acknowledgments

I would like to express my profound gratitude to my supervisor Prof. Nizar Bouguila, without whom, I would not be able to pursue my machine learning research journey. During the course of two years working together, he has always been a wise and witty mentor. Despite my slow start, his constant motivation has inspired me to move forward to complete the milestones. I will always be grateful for his relentless support and guidance.

I am fortunate to have been working with Mr. Muhammad Azam, who has always motivated me with his support and passion. I would not be able to complete the research without his constructive advises. Therefore, I always respect and consider him as a big brother.

I would like to thank Kamal, Meeta, Jaspreet, Narges, Eddy, Omar, Basim, Shuai, Samr, Fatma, Walid, and other lab members, who have always shared their vast knowledge as well as taken the time and effort to explain various concepts in order to make sure I could thoroughly understand them.

I would like extend my gratitude to my sister for her love and support since my first day in Montreal. Last but not least, I am grateful that my parents and girlfriend, who despite the fact of being halfway around the world, have been sending endless motivating encouragement in our daily loving conversations.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Cluster Analysis via Finite Mixture Models

Cluster analysis can be understood as the process of detecting different groups within a considered dataset [1]. In other words, similar data points are naturally categorized into the same group without the prior knowledge of the true number of groups. Undoubtedly, the aforementioned exploratory problem has been frequently discussed due to its applications among various disciplines [2]. For instance, detecting spam emails is considered as a highly challenging task due to the fact that they are becoming more insidious as well as the need of identifying them in real time. With ubiquity of emails in both personal and professional environments, an efficient tool for finding spams is crucial, and recurrent spam emails have been known as the cause for the decline in productivity and additional financial cost among various organizations [3]. Image clustering is another task which has attracted many recent studies [4], [5], [6]. Indeed, it is the heterogeneous essence of the images that raises as a huge obstacle to any proposed method. In other words, all the pixels containing most important features should be identified and analyzed, in which the context and behavior of each pixel could be learned through its position and value, respectively. Therefore, an accurate mathematical representation of the images is a key step in order to efficiently analyzing them [7].

Probabilistic models have been widely chosen for their versatility in different applications [8], [9], [10], [11]. With the initial assumption that the data are originated from a mixture of components following a particular probabilistic distribution, the parameters are then updated within the Expectation Maximization (EM) framework [12] in order to find

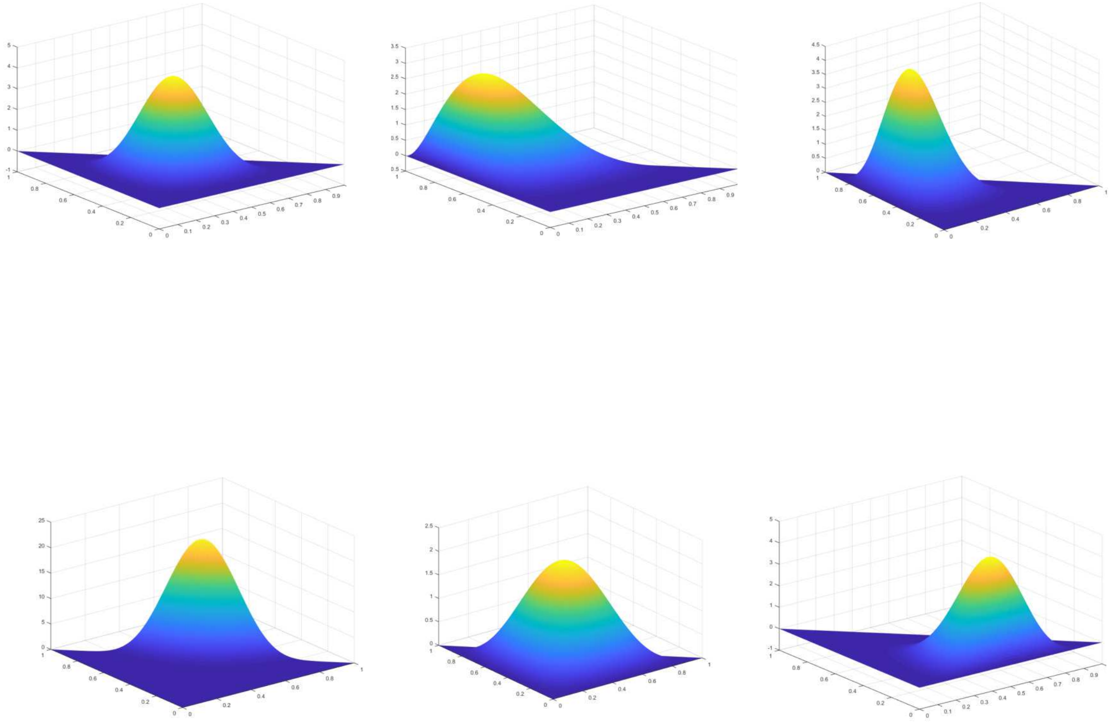


Figure 1: Different shapes of Scaled Dirichlet distribution

the optimal fit of the data points to the model [13]. Therefore, the flexibility of the chosen distribution plays an important role in the outcome of the model. Gaussian distribution has been a popular choice due to its adaptability to many cases [14], [15], [16], [17]. However, real life data come in with many different properties [18], many of which can be clearly seen as non-Gaussian, such as proportional data [19], for which Dirichlet family of distributions has been proven to be a more acclaimed choice for cluster analysis [20], [4], [21]. In addition, recent applications of Scaled Dirichlet distribution on anomaly detection and text clustering have proven its modeling capabilities [22], [23], [24]. With different parameter values, Scaled Dirichlet distribution's shapes are presented in Fig. 1.

The inference process is another crucial part in statistical modeling. Maximum Likelihood Estimation (MLE) is among the most used estimation approaches due to its simplicity in terms of implementation [25], [26]. Nonetheless, the process of maximizing the likelihood function could deviate from the global maximum and converge to a local maximum instead, which results in an unsatisfactory performance. Furthermore, ML also suffers from its sensitivity to the initialization [27]. Bayesian inference can overcome the previous disadvantages with the introduction of prior knowledge. Still, since the marginal distribution is intractable, it requires additional approximation methods such as Markov chain Monte Carlo (MCMC) [28] and Laplace's approximation [29]. Unfortunately, the drawbacks including complex computation and inability to ensure convergence outweigh the supplementary effort, causing some unnecessary compromises during implementation despite their applications among a variety of problems.

The variational approach has been then introduced with the inherited strengths from conventional Bayesian inference while avoiding its disadvantages [4], [30]. Its main idea is based on using an approximated variant of the true posterior distribution. Then, their difference is minimized by maximizing the lower bound of the joint likelihood function using Kullback-Leibler (KL) divergence. With the integrated approximation scheme, the variational framework can simultaneously update the model's parameters and determine the optimal number of components. Recently, it has received increasing attention with many applications in different domains such as image clustering, spam detection, and image segmentation. Furthermore, it has been shown that online learning can handle large scale data effectively [31].

1.2 Contributions

The goal of this thesis is to introduce several novel variational approaches for finite Scaled Dirichlet mixture models including the mean field variational inference without a local model selection, mean field variational inference with component splitting, and online stochastic variational inference. The contributions are listed as follows:

☞ **Data Clustering using Variational Learning of Finite Scaled Dirichlet Mixture Models**

> We propose the application of variational inference on Scaled Dirichlet mixture models, a more generalized and flexible distribution than Dirichlet, having an additional scale parameter, which determines the spread of the distribution.

The parameters as well as the model's complexity are optimized through the minimization of the KL divergence. This work has been accepted by the 28^{th} *International Symposium on Industrial Electronics*.

☞ **Data Clustering using Variational Learning of Finite Scaled Dirichlet Mixture Models with Component Splitting**

A variational Bayesian inference for finite Scaled Dirichlet mixture model is proposed along with component splitting, a local model selection framework. The main idea is starting from two components and then gradually adding new components by splitting existing ones based on their mixing weights. The optimal number of components is achieved when the splitting test is no longer applicable. This contribution has been submitted to the 16^{th} *International Conference on Image Analysis and Recognition*.

☞ **Data Clustering using Online Variational Learning of Finite Scaled Dirichlet Mixture Models**

We introduce an online variational Bayesian framework for finite Scaled Dirichlet mixture models. The proposed method is capable of estimating values for the parameters as well as computing the model's complexity in a sequential way for large scale data in real time. This research work has been submitted to the 20^{th} *International Conference on Information Reuse and Integration for Data Science*.

1.3 Thesis Overview

❑ Chapter 1 briefly introduced the fundamentals of cluster analysis along with several current prominent applications. The motivation for the determined probabilistic distribution and the variational inference framework are also clearly explained.

❑ In Chapter 2, we develop a variational inference learning approach for Scaled Dirichlet mixture models, which could simultaneously estimate the parameters and find the optimal number of components. Different real-life challenging problems including texture and object clustering are used for validating the performance of the proposed model.

❏ In chapter 3, we integrate component splitting, a local model selection method, to assist the model's complexity prediction process. Our model has been tested with extensive experiments consisting of spam email detection, texture, object, and scene clustering. The results have shown the effectiveness of the proposed approach.

❏ Chapter 4 describes the application of online variational approach on finite Scaled Dirichlet mixture models. With the idea of stochastic variational inference, both the parameters and model's complexity are computed efficiently for large scale datasets. The effectiveness of the model is tested with demanding applications such as email spam detection and image categorization.

❏ In conclusion, we briefly summarize our contributions and some remarks for potential future works

Chapter 2

Variational Learning of Finite Scaled Dirichlet Mixture Models

In this chapter, we propose a variational framework for Scaled Dirichlet mixture models. The prominent advantages include the ability to automatically update the parameters as well as estimate the model's complexity. Indeed, the variational inference can be seen as an optimization process, in which we focus on minimizing the difference between approximated posterior distribution and the true one using KL divergence. The performance of the proposed method is validated with different challenging problems such as texture and object clustering.

2.1 Finite Scaled Dirichlet Mixture Model

Assuming a set of N D-dimensional vectors generated from Scaled Dirichlet distribution $\mathcal{X} = \left(\vec{X}_1, ..., \vec{X}_N\right)$. Then, the vectors follow the probability density function $p\left(\vec{X}_i \mid \vec{\alpha}, \vec{\beta}\right)$:

$$p\left(\vec{X}_i \mid \vec{\alpha}, \vec{\beta}\right) = \frac{\Gamma\left(\alpha_+\right)}{\prod_{d=1}^{D} \Gamma\left(\alpha_d\right)} \frac{\prod\limits_{d=1}^{D} \beta_d^{\alpha_d} X_{id}^{\alpha_d - 1}}{\left(\sum\limits_{d=1}^{D} \beta_d X_{id}\right)^{\alpha_+}} \tag{1}$$

where $\Gamma(\cdot)$ is the Gamma function, $\vec{\alpha} = (\alpha_1, ..., \alpha_D)$, $\alpha_d > 0$ for $d = 1, ..., D$, $\vec{\beta} = (\beta_1, ..., \beta_D)$, $0 \leq \beta_d \leq 1$ for $d = 1, ..., D$, $\sum\limits_{d=1}^{D} \beta_d = 1$, and $\alpha_+ = \sum\limits_{d=1}^{D} \alpha_d$.

Then, the M-component finite Scaled Dirichlet mixture model (SDMM) is defined as folllows:

$$p\left(\vec{X}_i \mid \vec{\pi}, \vec{\alpha}_j, \vec{\beta}_j\right) = \sum_{j=1}^{M} \pi_j p\left(\vec{X}_i \mid \vec{\alpha}_j, \vec{\beta}_j\right) \tag{2}$$

where $\vec{\pi} = (\pi_1, ..., \pi_M)$ is the vector of mixing coefficients with respect to each component, which are positive and sum to 1. Then, $\vec{\alpha}_j$ and $\vec{\beta}_j$ denote the distribution's parameters with respect to component j. So, the likelihood function is:

$$p\left(\mathcal{X} \mid \vec{\pi}, \vec{\alpha}_j, \vec{\beta}_j\right) = \prod_{i=1}^{N}\left[\sum_{j=1}^{M} \pi_j p\left(\vec{X}_i \mid \vec{\alpha}_j, \vec{\beta}_j\right)\right] \tag{3}$$

For each vector $\vec{X}_i$, a M-dimensional assigning vector $\vec{Z}_i = (Z_{i1}, ..., Z_{iM})$, where $Z_{ij} \in \{0, 1\}$, $\sum_{j=1}^{M} Z_{ij} = 1$ and $Z_{ij} = 1$ if $\vec{X}_i$ belongs to component j and 0, otherwise. The conditional probability of $\mathcal{Z} = \left(\vec{Z}_1, ..., \vec{Z}_N\right)$ given $\vec{\pi}$ is:

$$p\left(\mathcal{Z} \mid \vec{\pi}\right) = \prod_{i=1}^{N}\prod_{j=1}^{M} \pi_j^{\mathcal{Z}_{ij}} \tag{4}$$

So, the conditional probability of data set $\mathcal{X}$ with the class labels $\mathcal{Z}$ is as follows:

$$p\left(\mathcal{X} \mid \mathcal{Z}, \vec{\alpha}, \vec{\beta}\right) = \prod_{i=1}^{N}\prod_{j=1}^{M} p\left(\vec{X}_i \mid \vec{\alpha}_j, \vec{\beta}_j\right)^{Z_{ij}} \tag{5}$$

Where $\vec{\alpha} = (\vec{\alpha}_1, ..., \vec{\alpha}_M)$ and $\vec{\beta} = \left(\vec{\beta}_1, ..., \vec{\beta}_M\right)$. The estimation of the mixture parameters and finding the optimal number of components M is a crucial part of a mixture model. The next section provides details about the variational Bayesian inference.

2.2 Variational Bayesian Learning

Following Bayesian inference, Gamma and Dirichlet distributions are chosen as priors for $\vec{\alpha}_{jd}$ and $\vec{\beta}_j$, respectively:

$$p\left(\alpha_{jd}\right) = \mathcal{G}\left(\alpha_{jd} \mid u_{jd}, v_{jd}\right) = \frac{v_{jd}^{u_{jd}}}{\Gamma\left(u_{jd}\right)} \alpha_{jd}^{u_{jd}-1} e^{-v_{jd}\alpha_{jd}} \tag{6}$$

$$p\left(\vec{\beta}_j\right) = \mathcal{D}\left(\vec{\beta} j \mid \vec{h}_j\right) = \frac{\Gamma\left(\sum_{d=1}^{D} h_{jd}\right)}{\prod_{d=1}^{D}\Gamma\left(h_{jd}\right)} \prod_{d=1}^{D} \beta_{jd}^{h_{jd}-1} \tag{7}$$

where $\vec{h}_j = (h_{j1}, ..., h_{jD})$, Gamma and Dirichlet distributions are denoted as $\mathcal{G}(.)$ and $\mathcal{D}(.)$, respectively; $\{u_{jd}\}$, $\{v_{jd}\}$, and $\{h_{jd}\}$ are positive hyperparameters. So

$$p\left(\vec{\alpha}\right) = \prod_{j=1}^{M} \prod_{d=1}^{D} p\left(\alpha_{jd}\right) \tag{8}$$

$$p\left(\vec{\beta}\right) = \prod_{j=1}^{M} \prod_{d=1}^{D} p\left(\beta_{jd}\right) \tag{9}$$

Thus, the joint distribution of all the random variables is as follows:

$$\begin{aligned}
p\left(\mathcal{X}, \Theta \mid \vec{\pi}\right) &= p\left(\mathcal{X} \mid \mathcal{Z}, \vec{\alpha}, \vec{\beta}\right) p\left(\mathcal{Z} \mid \vec{\pi}\right) p\left(\vec{\alpha}\right) p\left(\vec{\beta}\right) \\
&= \prod_{i=1}^{N} \prod_{j=1}^{M} \left[\pi_j \frac{\Gamma\left(\alpha_+\right)}{\prod_{d=1}^{D} \Gamma\left(\alpha_{jd}\right)} \frac{\prod\limits_{d=1}^{D} \beta_{jd}^{\alpha_{jd}} X_{id}^{\alpha_{jd}-1}}{\left(\sum\limits_{d=1}^{D} \beta_{jd} X_{id}\right)^{\alpha_+}} \right]^{Z_{ij}} \\
&\times \prod_{j=1}^{M} \prod_{d=1}^{D} \left[\frac{v_{jd}^{u_{jd}}}{\Gamma\left(u_{jd}\right)} \alpha_{jd}^{u_{jd}-1} e^{-v_{jd}\alpha_{jd}} \right. \\
&\times \left. \frac{\Gamma\left(\sum\limits_{d=1}^{D} h_{jd}\right)}{\prod_{d=1}^{D} \Gamma\left(h_{jd}\right)} \prod_{d=1}^{D} \beta_{jd}^{h_{jd}-1} \right]
\end{aligned} \tag{10}$$

x where $\Theta = \left\{\mathcal{Z}, \vec{\alpha}, \vec{\beta}\right\}$. The model's graphical representation is shown in Fig. 2.

The main idea is to find the true posterior distribution $p\left(\Theta \mid \mathcal{X}, \vec{\pi}\right)$ by defining $\mathcal{Q}\left(\Theta\right)$ as an approximation to it. By applying the KL divergence, the difference between two distributions is measured as follows

$$\mathcal{L}\left(\mathcal{Q}\right) = \ln\ p\left(\mathcal{X} \mid \vec{\pi}\right) - KL\left(\mathcal{Q} \mid\mid P\right) \tag{11}$$

where

$$KL\left(\mathcal{Q} \mid\mid P\right) = -\int \mathcal{Q}\left(\Theta\right) \ln\left(\frac{p\left(\Theta \mid \mathcal{X}, \vec{\pi}\right)}{\mathcal{Q}\left(\Theta\right)}\right) d\Theta \tag{12}$$

$$\mathcal{L}\left(\mathcal{Q}\right) = \int \mathcal{Q}\left(\Theta\right) \ln\left(\frac{p\left(\mathcal{X}, \Theta \mid \vec{\pi}\right)}{\mathcal{Q}\left(\Theta\right)}\right) d\Theta \tag{13}$$

It is clear that the lower bound $\mathcal{L}\left(Q\right)$ reaches its maximum value when the KL divergence equals zero. However, it is hardly feasible to compute the true posterior directly. Therefore, by applying the mean field theory [32], we could factorize $\mathcal{Q}\left(\Theta\right)$ to become $\mathcal{Q}\left(\Theta\right) =$

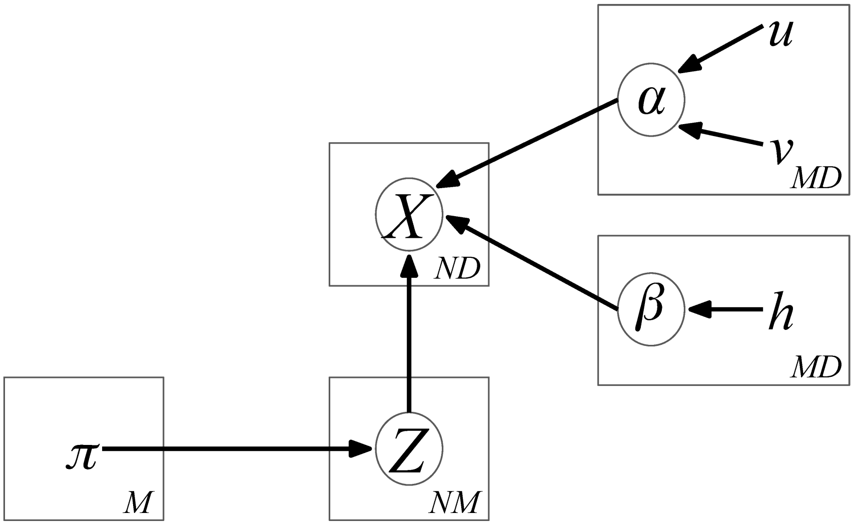


Figure 2: Graphical representation of the finite Scaled Dirichlet mixture model. Symbols with circles show the random variables and parameters. Plates denote repetitions, and the numbers in the lower right corners of the plates indicate the quantity of repetitions. The arcs give the conditional dependencies of the variables.

$\mathcal{Q}(\mathcal{Z})\,\mathcal{Q}(\vec{\alpha})\,\mathcal{Q}\left(\vec{\beta}\right)$. The maximization of lower bound $\mathcal{L}(Q)$ corresponding to each of the distributions $\mathcal{Q}_s(\Theta_s)$ is done via following function:

$$\mathcal{Q}_s(\Theta_s) = \frac{exp\left\langle \ln\ p(\mathcal{X},\Theta)\right\rangle_{j\neq s}}{\int exp\left\langle \ln\ p(\mathcal{X},\Theta)\right\rangle_{j\neq s} d\Theta} \tag{14}$$

where $\langle . \rangle_{j\neq s}$ represents the expectation of all the parameters with the exception of $j = s$. We utilize (14) to update our model until convergence:

$$\mathcal{Q}(\mathcal{Z}) = \prod_{i=1}^{N}\prod_{j=1}^{M} r_{ij}^{Z_{ij}} \tag{15}$$

$$\mathcal{Q}(\vec{\alpha}) = \prod_{j=1}^{M}\prod_{d=1}^{D} \mathcal{G}\left(\alpha_{jd} \mid u_{jd}^{*}, v_{jd}^{*}\right) \tag{16}$$

$$\mathcal{Q}\left(\vec{\beta}\right) = \prod_{j=1}^{M}\prod_{d=1}^{D} \mathcal{D}\left(\beta_{jd} \mid h_{jd}^{*}\right) \tag{17}$$

where

$$r_{ij} = \frac{p_{ij}}{\sum_{j=1}^{M} p_{ij}} \tag{18}$$

$$\begin{aligned}\boldsymbol{p}_{ij} = exp\Bigg\{ & \ln\pi_j + \tilde{R}_j + \sum_{d=1}^{D}\left[\overline{\alpha}_{jd}\ln\overline{\beta}_{jd} + (\overline{\alpha}_{jd}-1)\ln X_{id}\right] \\ & - \sum_{d=1}^{D}\overline{\alpha}_{jd}\ln\left(\sum_{d=1}^{D}\overline{\beta}_{jd}X_{id}\right)\Bigg\}\end{aligned} \tag{19}$$

$$\begin{aligned}\tilde{R}_j = & \ \ln\frac{\Gamma\left(\sum_{d=1}^{D}\overline{\alpha}_{jd}\right)}{\prod_{d=1}^{D}\Gamma\left(\overline{\alpha}_{jd}\right)} \\ & + \sum_{d=1}^{D}\overline{\alpha}_{jd}\left[\psi\left(\sum_{d=1}^{D}\overline{\alpha}_{jd}\right) - \psi\left(\overline{\alpha}_{jd}\right)\right]\left[\langle\ln\alpha_{jd}\rangle - \ln\overline{\alpha}_{jd}\right] \\ & + \frac{1}{2}\sum_{d=1}^{D}\overline{\alpha}_{jd}^{2}\left[\psi'\left(\sum_{d=1}^{D}\overline{\alpha}_{jd}\right) - \psi'\left(\overline{\alpha}_{jd}\right)\right] \\ & - \left\langle\left(\ln\alpha_{jd} - \ln\overline{\alpha}_{jd}\right)^{2}\right\rangle \\ & + \frac{1}{2}\sum_{a=1}^{D}\sum_{b=1,a\neq b}^{D}\overline{\alpha}_{ja}\overline{\alpha}_{jb}\Bigg\{\psi'\left(\sum_{d=1}^{D}\overline{\alpha}_{jd}\right)\left(\langle\ln\overline{\alpha}_{ja}\rangle - \ln\overline{\alpha}_{ja}\right) \\ & \times\left(\langle\ln\overline{\alpha}_{jb}\rangle - \ln\overline{\alpha}_{jb}\right)\Bigg\}\end{aligned} \tag{20}$$

$$\boldsymbol{u}_{jd}^{*} = \boldsymbol{u}_{jd} + \varphi_{jd}, \ \ \boldsymbol{v}_{jd}^{*} = \boldsymbol{v}_{jd} - \vartheta_{jd} \tag{21}$$

$$\begin{aligned}\varphi_{jd} = & \sum_{i=1}^{N}\langle Z_{ij}\rangle\overline{\alpha}_{jd}\Bigg[\psi\left(\sum_{d=1}^{D}\overline{\alpha}_{jd}\right) - \psi\left(\overline{\alpha}_{jd}\right) \\ & + \sum_{d\neq s}^{D}\psi'\left(\sum_{d=1}^{D}\overline{\alpha}_{jd}\right)\times\overline{\alpha}_{js}\left(\langle\ln\alpha_{js}\rangle - \ln\overline{\alpha}_{js}\right)\Bigg]\end{aligned} \tag{22}$$

$$\vartheta_{jd} = \sum_{i=1}^{N}\langle Z_{ij}\rangle\left[\ln\overline{\beta}_{jd} + \ln X_{id} - \ln\left(\sum_{d=1}^{D}\overline{\beta}_{jd}X_{id}\right)\right] \tag{23}$$

$$\boldsymbol{h}_{jd}^{*} = \boldsymbol{h}_{jd} + \tau_{jd} \tag{24}$$

$$\tau_{jd} = \sum_{i=1}^{N}\langle Z_{ij}\rangle\left[\overline{\alpha}_{jd} - \overline{\alpha}_{jd}\overline{\beta}_{jd}\frac{X_{id}}{\sum\limits_{d=1}^{D}\overline{\beta}_{jd}X_{id}}\right] \tag{25}$$

Algorithm 1 varSDMM Framework

```
1: Choose a large initial number of components $M$
2: Randomize initial values for $\{u_{jd}\}$, $\{v_{jd}\}$, $\{h_{jd}\}$
3: Initialize $r_{ij}$ using K-Means
4: E-step: Update $\mathcal{Q}\left(\mathcal{Z}\right)$ (15), $\mathcal{Q}\left(\vec{\alpha}\right)$ (16), $\mathcal{Q}\left(\vec{\beta}\right)$ (17)
5: M-step: Maximize $\mathcal{L}\left(\mathcal{Q}\right)$ corresponding to the current value of $(\vec{\pi})$ (30)
6: Repeat steps 4 and 5 until convergence
7: Determine number of components $M$ by naturally removing those with insignificant
   mixing coefficients (eg. smaller than $10^{-5}$)
8: Estimate new values for parameters $(\vec{\alpha})$ (16), $\left(\vec{\beta}\right)$ (17), and $(\vec{\pi})$ (30)
```

where $\psi\left(.\right)$ and $\psi'\left(.\right)$ are the digamma and trigamma functions, respectively. The expectation of values addressed in the equations above are

$$\left\langle Z_{ij}\right\rangle = r_{ij} \tag{26}$$

$$\overline{\alpha}_{jd} = \left\langle \alpha_{jd}\right\rangle = \frac{u_{jd}}{v_{jd}}, \quad \left\langle \ln\alpha_{jd}\right\rangle = \psi\big(u_{jd}\big) - \ln v_{jd} \tag{27}$$

$$\left\langle \big(\ln\alpha_{jd} - \ln\overline{\alpha}_{jd}\big)^2\right\rangle = \Big[\psi\big(u_{jd}\big) - \ln u_{jd}\Big]^2 + \psi'\big(u_{jd}\big) \tag{28}$$

$$\overline{\beta}_{jd} = \left\langle \beta_{jd}\right\rangle = \frac{h_{jd}}{\sum\limits_{d=1}^{D} h_{jd}} \tag{29}$$

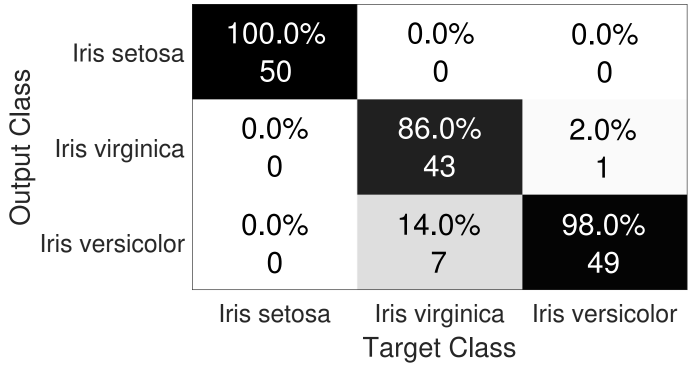
The complete summary for the VSDMM algorithm is presented in Algorithm 1. The maximization of lower bound $\mathcal{L}\left(\mathcal{Q}\right)$ along with variational updates for $\mathcal{Q}\left(\mathcal{Z}\right)$, $\mathcal{Q}\left(\vec{\alpha}\right)$, and $\mathcal{Q}\left(\vec{\beta}\right)$, allows the estimation of the mixing coefficients $\vec{\pi}$. By setting the derivative of $\mathcal{L}\left(Q\right)$ corresponding to $\vec{\pi}$ to zero, we obtain:

$$\pi_j = \frac{1}{N}\sum_{i=1}^{N} r_{ij} \tag{30}$$

During the variational learning, the mixing coefficients of components with insignificant contribution to analyze the data would be reduced to zero. Therefore, those components are automatically eliminated from the model. The algorithm reaches convergence if the difference of the lower bound values in two consecutive iterations is insignificant.

Table 1: Results on Iris dataset using different models

Method	Accuracy(%)
varSDMM	94.70
GMM	89.00
varGMM	86.70
varDMM	83.00



Output Class \ Target Class	Iris setosa	Iris virginica	Iris versicolor
Iris setosa	100.0% 50	0.0% 0	0.0% 0
Iris virginica	0.0% 0	86.0% 43	2.0% 1
Iris versicolor	0.0% 0	14.0% 7	98.0% 49

Figure 3: Confusion Matrix using varSDMM on Iris dataset

2.3 Experimental Results

In this section, we detail the experiment scope, configurations, and results compared with two finite variational mixture models based on Gaussian (varGMM) and Dirichlet (varDMM) distributions and a MLE-based finite Gaussian mixture model (GMM). The efficiency of varSDMM has been tested on several data categorization applications. The initial values of the hyperparameters $\{u_{jd}\}$, $\{v_{jd}\}$, $\{h_{jd}\}$ may affect the model's accuracy significantly. Therefore, finding a good combination of initialized hyperparameters is essential in order to improve the convergence speed as well as the detection of optimal number of components. Pre-processing data before running the algorithm has also enhanced the overall perfromance. Several normalization techniques have been applied namely Rescaling (min-max normalization), Mean normalization, and standardization.

2.3.1 Multivariate Data Categorization

We considered one of the classic datasets in machine learning, Iris dataset which was first introduced in [33], and now it is available for research purposes on UCI - Machine Learning Repository [34]. The dataset is created from 150 flowers, which are evenly divided into

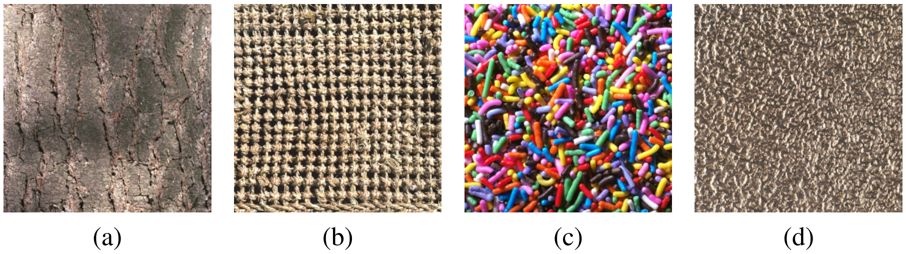


Figure 4: Images from Vistex: (a) Bark, (b) Fabric, (c) Food, (d) Metal

three groups, represented by the flowers' names: Iris setosa, Iris virginica, and Iris versicolor. While the first group is relatively distinct, the remaining clusters somewhat overlap each other raising a challenge. There are five features of the flowers: species, sepal length, sepal width, petal length, petal width.

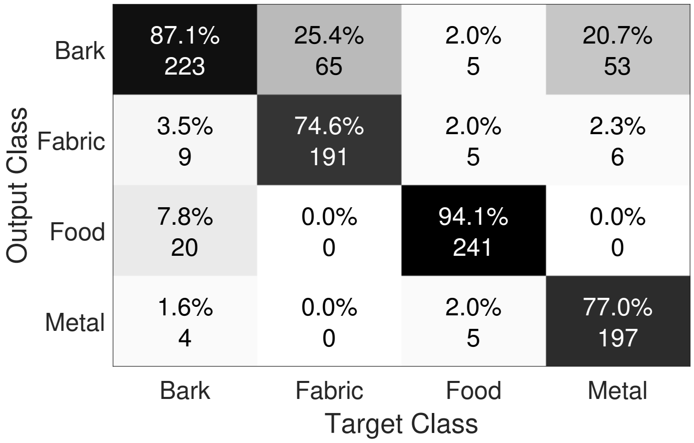
The categorization results are shown in Table 1 along the confusion matrix in Fig. 3. Clearly, varSDMM outperforms varDMM, varGMM, and GMM in terms of accuracy. It is noted that this result was achieved given that min-max normalization was applied on the dataset before running our model.

2.3.2 Texture Categorization

Texture categorization is another challenging task we address, an efficient texture analysis framework can help enhance the performance of other applications namely object segmentation or scene recognition [35]. For our experiment, we used the Vistex texture database from MIT Media Lab. Four homogeneous groups were considered: Bark, Fabric, Food, and Metal, each from which we sampled four images making total sample size of 16. However, we decided to challenge the extent of our model by considering each *512* $\times$ *512* original image as a mother image, then dividing them into 64 *64* $\times$ *64* images. Thus, the new sample size is 1024, with 256 images in each category. Examples from each group are presented in Fig. 4.

The texture characteristics are represented via co-occurrence matrix [36]. Each co-occurrence was computed with regards to its neighborhoods: (1;0), (1;$\frac{\pi}{4}$), (1;$\frac{\pi}{2}$), and (3;$\frac{\pi}{4}$). We calculated co-occurrence matrix of each of the neighborhood considering four features: Contrast, Correlation, Energy, Homogeneity. Thus, we combined them together to obtain a $16D$ feature vector.

The confusion matrix in Fig. 5 shows that the majority of the images are accurately



	Bark	Fabric	Food	Metal
Bark	87.1% 223	25.4% 65	2.0% 5	20.7% 53
Fabric	3.5% 9	74.6% 191	2.0% 5	2.3% 6
Food	7.8% 20	0.0% 0	94.1% 241	0.0% 0
Metal	1.6% 4	0.0% 0	2.0% 5	77.0% 197

Figure 5: Confusion Matrix using varSDMM on Vistex dataset

Table 2: Results on Vistex dataset using different models

Method	Accuracy(%)
varSDMM	83.20
varGMM	74.40
varDMM	63.90
GMM	62.20

categorized, especially those from group Food. Table 2 shows that varSDMM's accuracy is significantly higher than those of other methods. It is also worth mentioning that min-max normalization helped improving the result by approximately 2%.

2.3.3 Object Categorization

Image categorization has always been a frequently discussed topic in computer vision [37]. Indeed, many research contributions have tackled this problem with different scopes and approaches namely classification of sport activities [4], scenes [5], medical-related images (eg. different body parts) [38], [39].

For this experiment, we address the object categorization task using Caltech 101 dataset [40]. There are 101 groups of different objects, animals, faces, etc. Due to the immense imbalance of number of images among the groups, we sampled two datasets: dataset A consists of 200 images evenly divided into four groups: Starfish, Soccer ball, Faces, and Ketch; dataset B has 550 images from 4 groups: Motorbikes (150), Airplanes (150), Faces (150), Hawksbill (100). The sample images from two datasets are presented in Fig. 6.

(a) (b) (c) (d) h (e) (f) (g)

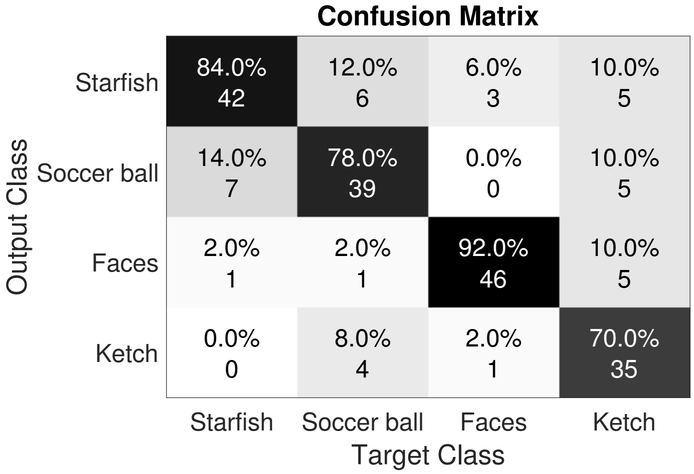
Figure 6: Examples from Caltech101: (a) Starfish, (b) Soccer ball, (c) Face, (d) Ketch, (e) Motorbike, (f) Airplane, (g) Hawksbill

Table 3: Results on datasets A and B from Caltech101 using different models

Method	Accuracy (%)	
	Dataset A	Dataset B
varSDMM	81.00	84.00
varDMM	80.00	55.40
varGMM	73.00	72.50
GMM	67.50	75.50

An accurate representation in feature space of a dataset is an important task before carrying out any prediction process. In other words, it requires an efficient descriptor having most of the important features. Thus, we chose SIFT (Scale Invariant Feature transform) [41] since it has proved its capability and robustness in different classification problems [21], [4], [5]. SIFT's descriptors are presented as $128D$ vectors, all of which are put into a collection of local features. Then, we use K-means to perform clustering process in order to construct the dictionary of visual words. Each cluster centroid is considered as a visual word and the vocabulary of the dictionary is a predetermined number of the centroids.

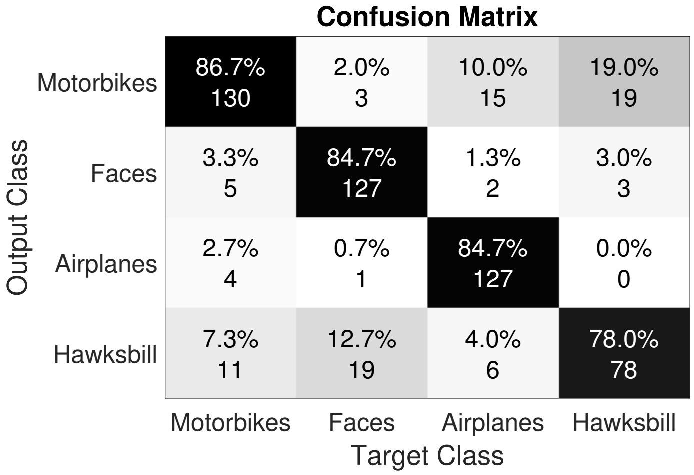
After several tests, we determined that the optimal number of visual words was 50. The confusion matrices when applying vadSDMM on datasets A and B are shown in Fig. 7 and Fig. 8, respectively. Then, in order to confirm the efficiency of our model, we compared our results with other models, the summaries are presented in Table 3 for both datasets A and B. Thus, we have tested varSDMM on four datasets and compared the results with other variational models to prove the capability and effectiveness of our model with different challenges.



Confusion Matrix

Output Class \ Target Class	Starfish	Soccer ball	Faces	Ketch
Starfish	84.0% 42	12.0% 6	6.0% 3	10.0% 5
Soccer ball	14.0% 7	78.0% 39	0.0% 0	10.0% 5
Faces	2.0% 1	2.0% 1	92.0% 46	10.0% 5
Ketch	0.0% 0	8.0% 4	2.0% 1	70.0% 35

Figure 7: Confusion Matrix using varSDMM on dataset A



Confusion Matrix

Output Class \ Target Class	Motorbikes	Faces	Airplanes	Hawksbill
Motorbikes	86.7% 130	2.0% 3	10.0% 15	19.0% 19
Faces	3.3% 5	84.7% 127	1.3% 2	3.0% 3
Airplanes	2.7% 4	0.7% 1	84.7% 127	0.0% 0
Hawksbill	7.3% 11	12.7% 19	4.0% 6	78.0% 78

Figure 8: Confusion Matrix using varSDMM on dataset B

Chapter 3

Variational Bayesian Learning of finite Scaled Dirichlet mixture models with Component Splitting

Previously, we have successfully applied the variational Bayesian learning framework on finite Scaled Dirichlet mixture model, in which the parameters' estimation was accurately achieved without the cumbersome computational cost of conventional Bayesian methods. In this chapter, component splitting, a local model selection scheme, is integrated into the framework to compute the model's complexity. The structure of the model has been explained in Section 2.1. The model is tested with different challenging problems including spam detection and image clustering to validate its effciency.

3.1 Variational Bayesian Learning with Component Splitting

The use of component splitting is inherited from [42]. First, the mixture components are divided into two parts, $fixed$ components and $free$ components. While the $M - s$ fixed components already provided a reasonable fit for the data, the model selection process operates on the s free ones. Therfore, the prior ditribution of Z can be rewritten as follows:

$$p(\mathcal{Z} \mid \vec{\pi}, \vec{\pi}^*) = \prod_{i=1}^{N} \left[\prod_{j=1}^{s} \pi_j^{Z_{ij}} \prod_{j=s+1}^{M} \pi_j^{*Z_{ij}} \right] \tag{31}$$

where $\vec{\pi} = \{\pi_j\}$ are the mixing coefficients of the free components, $\vec{\pi}^* = \{\pi_j^*\}$ are the mixing coefficients of the fixed ones, and their sum must be 1: $\sum_{j=1}^{s} \pi_j + \sum_{j=s+1}^{M} \pi_j^* = 1$. Considering π_j^* as a random variable, the prediction for optimal number of components is then computed solely on the free components by maximizing the marginal likelihood given $\{\pi_j\}$. Then, according to [42], we have prior distribution for $\vec{\pi}^*$:

$$p(\vec{\pi}^* \mid \vec{\pi}) = \left(1 - \sum_{k=1}^{s} \pi_k\right)^{-M+s} \frac{\Gamma(\sum_{j=s+1}^{M} c_j)}{\prod_{j=s+1}^{M} \Gamma(c_j)} \prod_{j=s+1}^{M} \left(\frac{\pi_j^*}{1 - \sum_{k=1}^{s} \pi_k}\right)^{c_j - 1} \tag{32}$$

We choose Gamma and Dirichlet distribution as priors for $\vec{\alpha}_{jd}$ and $\vec{\beta}_j$, respectively:

$$p\left(\alpha_{jd}\right) = \mathcal{G}\left(\alpha_{jd} \mid u_{jd}, v_{jd}\right) = \frac{v_{jd}^{u_{jd}}}{\Gamma\left(u_{jd}\right)} \alpha_{jd}^{u_{jd}-1} e^{-v_{jd}\alpha_{jd}} \tag{33}$$

$$p\left(\vec{\beta}_j\right) = \mathcal{D}\left(\vec{\beta} j \mid \vec{h}_j\right) = \frac{\Gamma\left(\sum\limits_{d=1}^{D} h_{jd}\right)}{\prod_{d=1}^{D} \Gamma\left(h_{jd}\right)} \prod_{d=1}^{D} \beta_{jd}^{h_{jd}-1} \tag{34}$$

where $\vec{h}_j = (h_{j1}, ..., h_{jD})$, $\mathcal{G}(\cdot)$ and $\mathcal{D}(\cdot)$ represent Gamma and Dirichlet distributions, respectively; $\{u_{jd}\}$, $\{v_{jd}\}$, and $\{h_{jd}\}$ are hyperparameters, where $u_{jd} > 0$, $v_{jd} > 0$, and $h_{jd} > 0$. Therefore

$$p\left(\vec{\alpha}\right) = \prod_{j=1}^{M} \prod_{d=1}^{D} p\left(\alpha_{jd}\right), \; p\left(\vec{\beta}\right) = \prod_{j=1}^{M} \prod_{d=1}^{D} p\left(\beta_{jd}\right) \tag{35}$$

We have the joint distribution of all the random variables:

$$\begin{aligned}
& p\left(\mathcal{X}, \Theta \mid \vec{\pi}\right) = p\left(\mathcal{X} \mid \mathcal{Z}, \vec{\alpha}, \vec{\beta}\right) p\left(\mathcal{Z} \mid \vec{\pi}, \vec{\pi^*}\right) p\left(\vec{\pi^*} \mid \vec{\pi}\right) p\left(\vec{\alpha}\right) p\left(\vec{\beta}\right) \\
& = \prod_{i=1}^{N} \prod_{j=1}^{M} \left[\pi_j \frac{\Gamma\left(\alpha_+\right)}{\prod_{d=1}^{D} \Gamma\left(\alpha_{jd}\right)} \frac{\prod\limits_{d=1}^{D} \beta_{jd}^{\alpha_{jd}} X_{id}^{\alpha_{jd}-1}}{\left(\sum\limits_{d=1}^{D} \beta_{jd} X_{id}\right)^{\alpha_+}} \right]^{Z_{ij}} \times \prod_{i=1}^{N} \left[\prod_{j=1}^{s} \pi_j^{Z_{ij}} \prod_{j=s+1}^{M} \pi_j^{*Z_{ij}} \right] \\
& \times \left(1 - \sum_{k=1}^{s} \pi_k\right)^{-M+s} \times \frac{\Gamma\left(\sum_{j=s+1}^{M} c_j\right)}{\prod_{j=s+1}^{M} \Gamma\left(c_j\right)} \prod_{j=s+1}^{M} \left(\frac{\pi_j^*}{1 - \sum_{k=1}^{s} \pi_k}\right)^{c_j - 1} \\
& \times \prod_{j=1}^{M} \prod_{d=1}^{D} \frac{v_{jd}^{u_{jd}}}{\Gamma\left(u_{jd}\right)} \alpha_{jd}^{u_{jd}-1} e^{-v_{jd}\alpha_{jd}} \times \frac{\Gamma\left(\sum\limits_{d=1}^{D} h_{jd}\right)}{\prod_{d=1}^{D} \Gamma\left(h_{jd}\right)} \prod_{d=1}^{D} \beta_{jd}^{h_{jd}-1}
\end{aligned} \tag{36}$$

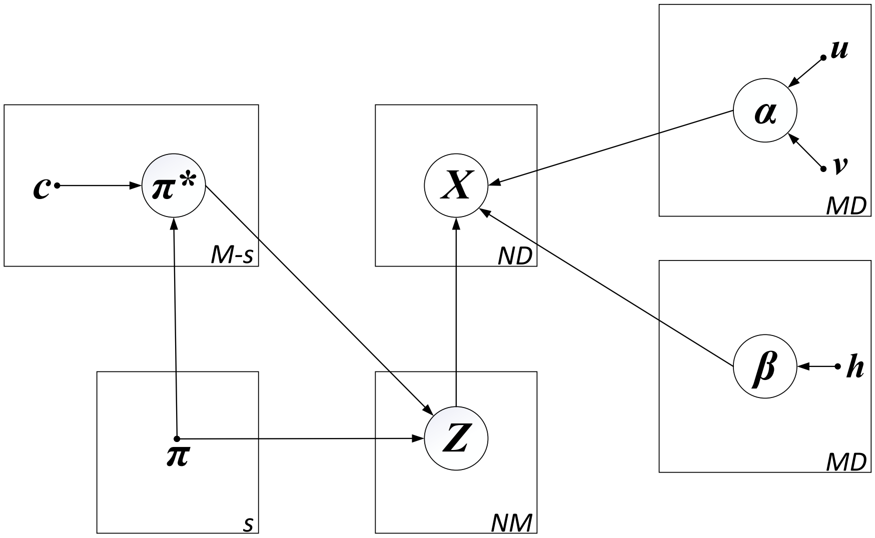


Figure 9: Graphical representation of the finite Scaled Dirichlet mixture model with component splitting. Symbols in circles denote parameters and random variables, arcs describe the conditional dependencies of the variables, plates show repetitions, and the numbers in the lower right corners of the plates explain the quantity of repetitions.

where $\Theta = \left\{\mathcal{Z}, \vec{\alpha}, \vec{\beta}, \vec{\pi^*}\right\}$ is the set of unknown parameters. The model's graphical representation is shown in Figure 9

The goal is to find the true posterior distribution $p\left(\Theta \mid \mathcal{X}, \vec{\pi}\right)$ by creating $Q\left(\Theta\right)$ as an approximated distribution to it. By applying the KL divergence, the difference between two distributions is computed as follows

$$\mathcal{L}\left(Q\right) = \ln\, p\left(\mathcal{X} \mid \vec{\pi}\right) - KL\left(Q \mid\mid P\right) \tag{37}$$

The maximum value of lower bound $\mathcal{L}\left(Q\right) = \int Q\left(\Theta\right) \ln\left(\frac{p(\mathcal{X},\Theta|\vec{\pi})}{Q(\Theta)}\right) d\Theta$ is achieved when the KL divergence is zero. Since the true posterior is intractable, the mean field theory [32] is applied to factorize $Q\left(\Theta\right)$ so that $Q\left(\Theta\right) = Q\left(\mathcal{Z}\right) Q\left(\vec{\alpha}\right) Q\left(\vec{\beta}\right) Q\left(\vec{\pi^*}\right)$. The maximization of lower bound $\mathcal{L}\left(Q\right)$ with respect to each sub-distribution $Q_s\left(\Theta_s\right)$ is:

$$Q_s\left(\Theta_s\right) = \frac{exp\left\langle \ln\, p\left(\mathcal{X}, \Theta\right)\right\rangle_{j\neq s}}{\int exp\left\langle \ln\, p\left(\mathcal{X}, \Theta\right)\right\rangle_{j\neq s} d\Theta} \tag{38}$$

where $\left\langle \cdot \right\rangle_{j\neq s}$ denotes the expectation of the parameters with the exception of $j = s$. Then, (38) is used for updating the algorithm to reach convergence:

$$\mathcal{Q}\left(\mathcal{Z}\right) = \prod_{i=1}^{N}\left[\prod_{j=1}^{s} r_{ij}^{Z_{ij}} \prod_{j=s+1}^{M} r_{ij}^{*Z_{ij}}\right] \tag{39}$$

$$\mathcal{Q}(\vec{\pi}^*) = \left(1-\sum_{k=1}^{s}\pi_k\right)^{-M+s} \frac{\Gamma\left(\sum_{j=s+1}^{M} c_j^*\right)}{\prod_{j=s+1}^{M}\Gamma\left(c_j^*\right)} \prod_{j=s+1}^{M}\left(\frac{\pi_j^*}{1-\sum_{k=1}^{s}\pi_k}\right)^{c_j^*-1} \tag{40}$$

$$Q\left(\vec{\alpha}\right) = \prod_{j=1}^{M}\prod_{d=1}^{D}\mathcal{G}\left(\alpha_{jd} \mid u_{jd}^*, v_{jd}^*\right) \tag{41}$$

$$Q\left(\vec{\beta}\right) = \prod_{j=1}^{M}\prod_{d=1}^{D}\mathcal{D}\left(\beta_{jd} \mid h_{jd}^*\right) \tag{42}$$

where

$$r_{ij} = \frac{\tilde{r}_{ij}}{\sum_{j=1}^{s}\tilde{r}_{ij} + \sum_{j=s+1}^{M}\tilde{r}_{ij}^*},\ r_{ij}^* = \frac{\tilde{r}_{ij}^*}{\sum_{j=1}^{s}\tilde{r}_{ij} + \sum_{j=s+1}^{M}\tilde{r}_{ij}^*} \tag{43}$$

$$\begin{aligned}\tilde{r}_{ij} = exp\Bigg\{ &\ln\pi_j + \tilde{R}_j + \sum_{d=1}^{D}\left[\overline{\alpha}_{jd}\ln\overline{\beta}_{jd} + \left(\overline{\alpha}_{jd}-1\right)\ln X_{id}\right] \\ &- \sum_{d=1}^{D}\overline{\alpha}_{jd}\ln\left(\sum_{d=1}^{D}\overline{\beta}_{jd}X_{id}\right)\Bigg\}\end{aligned} \tag{44}$$

$$\begin{aligned}\tilde{r}_{ij}^* = exp\Bigg\{ &\langle\ln\pi_j^*\rangle + \tilde{R}_j + \sum_{d=1}^{D}\left[\overline{\alpha}_{jd}\ln\overline{\beta}_{jd} + \left(\overline{\alpha}_{jd}-1\right)\ln X_{id}\right] \\ &- \sum_{d=1}^{D}\overline{\alpha}_{jd}\ln\left(\sum_{d=1}^{D}\overline{\beta}_{jd}X_{id}\right)\Bigg\}\end{aligned} \tag{45}$$

$$\begin{aligned}\tilde{R}_j =\ &\ln\frac{\Gamma\left(\sum_{d=1}^{D}\overline{\alpha}_{jd}\right)}{\prod_{d=1}^{D}\Gamma\left(\overline{\alpha}_{jd}\right)} + \sum_{d=1}^{D}\overline{\alpha}_{jd}\left[\psi\left(\sum_{d=1}^{D}\overline{\alpha}_{jd}\right) - \psi\left(\overline{\alpha}_{jd}\right)\right]\left[\langle\ln\alpha_{jd}\rangle - \ln\overline{\alpha}_{jd}\right] \\ &+ \frac{1}{2}\sum_{d=1}^{D}\overline{\alpha}_{jd}^2\left[\psi'\left(\sum_{d=1}^{D}\overline{\alpha}_{jd}\right) - \psi'\left(\overline{\alpha}_{jd}\right)\right] - \left\langle\left(\ln\alpha_{jd} - \ln\overline{\alpha}_{jd}\right)^2\right\rangle \\ &+ \frac{1}{2}\sum_{a=1}^{D}\sum_{b=1,a\neq b}^{D}\overline{\alpha}_{ja}\,\overline{\alpha}_{jb}\left\{\psi'\left(\sum_{d=1}^{D}\overline{\alpha}_{jd}\right)\left(\langle\ln\overline{\alpha}_{ja}\rangle - \ln\overline{\alpha}_{ja}\right)\times\left(\langle\ln\overline{\alpha}_{jb}\rangle - \ln\overline{\alpha}_{jb}\right)\right\}\end{aligned} \tag{46}$$

$$c_j^* = \sum_{i=1}^{N}r_{ij}^* + c_j,\ \ u_{jd}^* = u_{jd} + \varphi_{jd},\ \ v_{jd}^* = v_{jd} - \vartheta_{jd},\ \ h_{jd}^* = h_{jd} + \tau_{jd} \tag{47}$$

$$\begin{aligned}\varphi_{jd} = \sum_{i=1}^{N} \langle Z_{ij} \rangle \overline{\alpha}_{jd} \Bigg[\psi \left(\sum_{d=1}^{D} \overline{\alpha}_{jd} \right) - \psi \left(\overline{\alpha}_{jd} \right) \\ + \sum_{d \neq s}^{D} \psi' \left(\sum_{d=1}^{D} \overline{\alpha}_{jd} \right) \times \overline{\alpha}_{js} \left(\langle \ln \alpha_{js} \rangle - \ln \overline{\alpha}_{js} \right) \Bigg]\end{aligned} \tag{48}$$

$$\vartheta_{jd} = \sum_{i=1}^{N} \langle Z_{ij} \rangle \left[\ln \overline{\beta}_{jd} + \ln X_{id} - \ln \left(\sum_{d=1}^{D} \overline{\beta}_{jd} X_{id} \right) \right] \tag{49}$$

$$\tau_{jd} = \sum_{i=1}^{N} \langle Z_{ij} \rangle \left[\overline{\alpha}_{jd} - \overline{\alpha}_{jd} \overline{\beta}_{jd} \frac{X_{id}}{\sum\limits_{d=1}^{D} \overline{\beta}_{jd} X_{id}} \right] \tag{50}$$

where $\psi\left(\cdot\right)$ and $\psi'\left(\cdot\right)$ denote the digamma and trigamma functions, respectively. The expectation of the aforementioned equations are

$$\langle Z_{ij} \rangle = r_{ij}, \ \ \text{for } j = 1, ..., s, \langle Z_{ij} \rangle = r_{ij}^*, \ \ \text{for } j = s+1, ..., M \tag{51}$$

$$\overline{\alpha}_{jd} = \langle \alpha_{jd} \rangle = \frac{u_{jd}}{v_{jd}}, \quad \langle \ln \alpha_{jd} \rangle = \psi\big(u_{jd}\big) - \ln v_{jd}, \quad \overline{\beta}_{jd} = \langle \beta_{jd} \rangle = \frac{h_{jd}}{\sum\limits_{d=1}^{D} h_{jd}} \tag{52}$$

$$\left\langle \big(\ln \alpha_{jd} - \ln \overline{\alpha}_{jd} \big)^2 \right\rangle = \Big[\psi\big(u_{jd}\big) - \ln u_{jd} \Big]^2 + \psi'\big(u_{jd}\big) \tag{53}$$

$$\left\langle \pi_j^* \right\rangle = \left(1 - \sum_{k=1}^{s} \pi_k \right) \frac{\sum_{i=1}^{N} r_{ij}^* + c_j}{\sum_{k=s+1}^{M} \Big(\sum_{i=1}^{N} r_{ik}^* + c_k \Big)} \tag{54}$$

$$\left\langle \ln \pi_j^* \right\rangle = \ln \Big(1 - \sum_{k=1}^{s} \pi_k \Big) + \psi \Big(\sum_{i=1}^{N} r_{ij}^* + c_j \Big) - \psi \Big(\sum_{i=1}^{N} \sum_{k=s+1}^{M} r_{ik}^* + c_k \Big) \tag{55}$$

The estimation for the free mixing coefficients $\vec{\pi}$ is computed from the maximization of lower bound $\mathcal{L}\left(Q\right)$ and the variational updates for $\mathcal{Q}\left(\mathcal{Z}\right)$, $\mathcal{Q}\left(\vec{\pi}^*\right)$, $\mathcal{Q}\left(\vec{\alpha}\right)$, and $\mathcal{Q}\left(\vec{\beta}\right)$. We have the derivative of $\mathcal{L}\left(Q\right)$ with respect to $\vec{\pi}$ after setting it to zero:

$$\pi_j = \left(1 - \sum_{k=s+1}^{M} \langle \pi_k^* \rangle \right) \frac{\sum_{i=1}^{N} r_{ij}}{\sum_{i=1}^{N} \sum_{k=1}^{s} r_{ik}} \tag{56}$$

Algorithm 2 varSDMM with Component Splitting Framework

```
Initialize number of components $M$ to 2
Randomize initial values for $\{u_{jd}\}$, $\{v_{jd}\}$, $\{h_{jd}\}$
Start the variational inference without the local model selection
If only one component remains, the algorithm ends
Sort all the elements in $M$ in descending order by their mixing coefficients
For each element $j$ in $M$:
    • Split $j$ into $j_j$ and $j_2$ as the free components
    • Set:
        ◊ $\pi_{j_1} = \pi_{j_2} = \pi_j/2$
        ◊ $u_{jd_1} = u^*_{jd}, u_{jd_2} = u^*_{jd}$
        ◊ $v_{jd_1} = v^*_{jd}, v_{jd_2} = v^*_{jd}$
        ◊ $h_{jd_1} = h^*_{jd}, h_{jd_2} = h^*_{jd}$
    • $c^*_j = \sum_{i=1}^{N} r^*_{ij}$ for each $j$ in the fixed components
    • Apply variational inference with component splitting by updating $\mathcal{Q}(\mathcal{Z})$ (39), $\mathcal{Q}(\vec{\pi}^*)$ (40), $\mathcal{Q}(\vec{\alpha})$ (41), $\mathcal{Q}\left(\vec{\beta}\right)$ (42) until convergence
    • Use (56) to calculate the suitable number of components
    • Split test fails if only one remaining component left.
    • If both components are redundant, split test fails and move on the next component
    • If both components remains, then $M = M + 1$
Repeat steps 5, 6 until the splitting test fails in all the components
```

3.2 Model Selection via Component Splitting

First, the algorithm starts with the variational learning without local model selection where $M = 2$. If the result has two components, the splitting process proceeds; otherwise, the algorithm ends if there is only one component. When the splitting test is passed, one of the components is split into two free components. Next, the model with local model selection operates on the free components while leaving the fixed ones intact. Two common possibilities could occur after the inference: first, both free components are kept due to their meaningful contribution to fit the data; second, only one component is kept while the insignificant one is removed. However, when there are some outliers in the data set, both the free components could end up being redundant, then this particular split is restored in order to avoid an infinite loop. Then, after each successful split, the number of components gradually increases until all the split tests fail. The complete summary of the model's process is presented in Algorithm 2.

3.3 Experimental Results

In this section, we discuss the performance of our proposed method (varSDMM) as compared to MLE-based Gaussian mixture model (GMM), variational Gaussian mixture model (varGMM), variational Dirichlet mixture model (varDMM). Two challenging real life applications are considered including spam email detection of both texts as well as image categorization consisting of textures, objects, and scenes.

3.3.1 Spam detection

For the past two decades, e-mail has become an essential means of communication, especially in the workplace environment. However, e-mails are also one of the most common target for network-based attacks namely phishing [43], [44], [45], [46]. Spam emails containing not only texts, but also deceiving images combined with the evolve of various scam techniques are drawing increasing interest as a challenging task that needs immediate actions.

Since the performance of any model depends greatly on the the quality of preprocessing steps, an accurate mathematical representation in feature space of the images is crucial prior to applying the inference process. Therefore, SIFT (Scale Invariant Feature transform) [41] is used for preprocessing the images. Then, all the $128D$ descriptors of SIFT are grouped into a corpus of local features. Next, we use K-means to cluster the collection to construct the visual words vocabulary, in which the centroids are the number of visual words. The performance of each result is validated using four important measures: Accuracy($\frac{TP+TN}{TP+TN+FP+FN}$), Precision($\frac{TP}{TP+FP}$), Recall($\frac{TP}{TP+FN}$), False Positive Rate (FPR) ($\frac{FP}{FP+TN}$).

For textual spam e-mail detection, we chose the Spambase data set [47], in which the histogram of the occurrences of the words is used as a feature. We chose 3626 instances in the data set, half of which was spam and the other half was non-spam. The results in Table 4 shows that our proposed model outperforms others in all aspects.

Three real life image spam data sets were considered: Personal Image Spam (2995 images) [48], SpamArchive Image Spam (3014 images) [48], and Princeton Spam Image

Table 4: Results on Spambase (%) using different models

Method	Accuracy	Precision	Recall	False Positive Rate
varSDMM	85.60	99.61	70.44	0.28
varDMM	83.84	97.23	69.06	1.99
GMM	73.08	73.24	72.75	26.59
varGMM	71.37	69.56	76.01	33.26

STOCK IN PLAY for MONDAY
FEB 5th
Fire Mountain Beverage Company
OTC : FBVG.PK
Current Price : $0.03
5 Day Expected : $0.50
BREAKING NEWS
Fire Mountain Beverage Company Receives Over 40 Truckloads of New Orders From Major California Supermarket Chain
This is a Real Company with real products and real earnings
GET IN ON THE GROUND FLOOR
(095) 258-00-83
www.sobak.net
DAZER II
ТОЛЬКО
ОТПУГИВАТЕЛЬ СОБАК DAZER II
СДЕЛАННЫЙ В США
ПО-НАСТОЯЩЕМУ ЭФФЕКТИВЕН
HoodiaLife
Lose 25 pounds in 1 Month!
Discover the New Miracle Supplement for Safe, Effective Weight Loss!
Click Here To Order!
Featured on: OPRAH 60 MINUTES abc ABC-7
STEVEN SPIELBERG Presents
BACK TO THE FUTURE
ROBERT ZEMECKIS

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

Figure 10: Images from (a) Personal Image Spam, (b) SpamArchive, (c) Princeton, (d) Personal Image Ham.

(1063 images)[1]. One common legitimate (ham) email data set Personal Image Ham (1650 images) [48] is used for clustering analysis. Sample images from these data sets are shown in Figure 10. After several trials, the optimal number of visual vocabulary is 50. The results shown in Table 5 validates varSDMM's performance over other models.

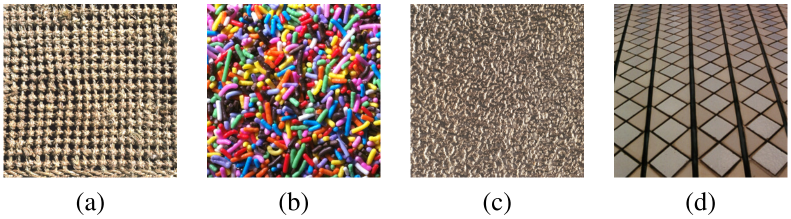
3.3.2 Texture Categorization

An efficient texture classification framework could not only help improve the performance of object clustering, but also the categorization of sophisticated collections of various objects such as human organs or scenes [35]. In this experiment, two real-life challenging texture datasets were used: Amsterdam Library of Textures (ALOT) [49] and Vistex. Particularly, we tested 600 images evenly divided into six clusters from ALOT: Macaroni, Corn Flakes, Silver foil, Banana peel, Mustard seed, and Plaster; sample images are in Fig. 12. The preprocessing step was similar to that mentioned in Section 3.3.1 with the optimal value for vocabulary was 50. For Vistex dataset, there are 16 observations which are equally divided into 4 groups: Fabric, Food, Metal, and Tile. However, in order to avoid ambiguity, each *512* × *512* observation is separated into 8 *64* × *64* parts making the total sample size 1024. Then, each instance is then represented as a $16D$ feature vector after

[1] http://www.cs.princeton.edu/cass/spam/

Table 5: Results on image spam detection using different models

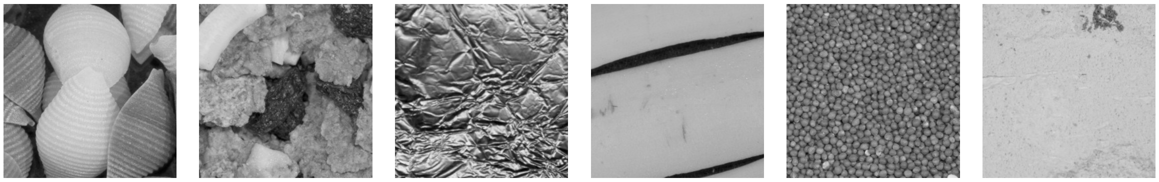
Method	Measure (%)	Dredze	SpamArchive	Princeton
varSDMM	Accuracy	88.63	86.94	86.18
	Precision	96.25	98.98	90.66
	Recall	85.71	80.62	72.15
	False Positive Rate	6.06	1.52	4.79
varDMM	Accuracy	87.56	80.87	84.11
	Precision	94.58	96.74	81.35
	Recall	85.61	72.86	71.14
	False Positive Rate	8.91	4.48	11.39
varGMM	Accuracy	86.29	81.56	84.37
	Precision	89.91	96.95	85.38
	Recall	88.68	73.79	72.53
	False Positive Rate	18.06	4.24	8.36
GMM	Accuracy	87.26	80.83	84.56
	Precision	91.73	95.45	85.86
	Recall	88.18	73.86	72.53
	False Positive Rate	14.42	6.42	7.70



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

Figure 11: Images from Vistex: (a) Fabric, (b) Food, (c) Metal, (d) Tile

using co-occurrence matrix [36], which has been explained in Section 2.3. Examples from Vistex dataset are presented in Fig. 11



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

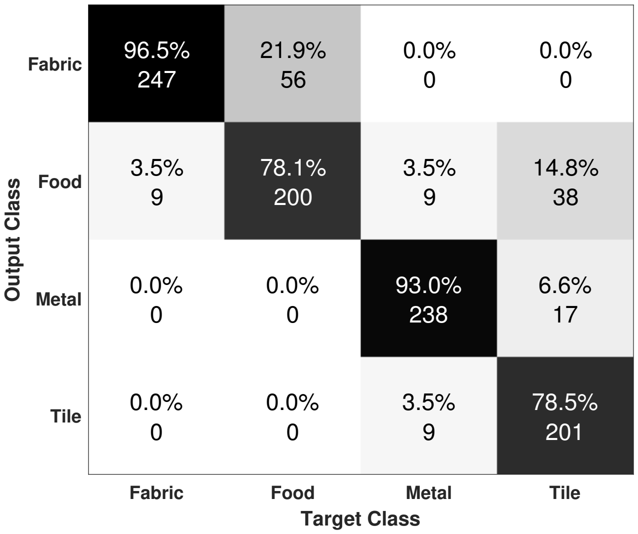
Figure 12: Sample images from ALOT: (a) Macaroni, (b) Corn Flakes, (c) Silver foil, (d) Banana peel, (e) Mustard seed, (f) Plaster

.

Table 6: Results on texture datasets using different models

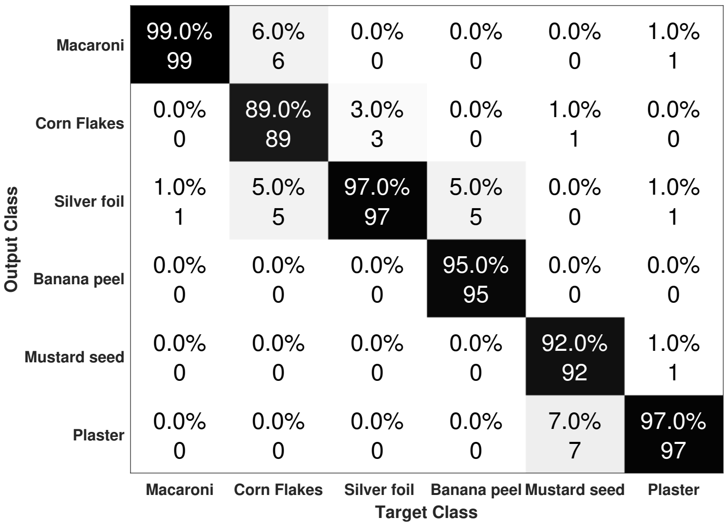
Method	Accuracy (%)	
	ALOT	Vistex
varSDMM	94.83	86.52
varDMM	78.83	75.00
varGMM	76.16	79.98
GMM	71.50	79.19

The results are presented in Table 6, showing that the proposed model surpasses other novel approaches by a significant margin. Particularly, despite the fact that there are many similar texture details among the groups, the result in confusion matrix for Vistex in Fig. 13 shows that the greatest amount of misclassification in a cluster is only 21.90%. Furthermore, it is clear that the proposed method is capable of achieving at least 89.00% of accuracy in each cluster when tested with ALOT as presented in Fig. 14.



	Fabric	Food	Metal	Tile
Fabric	96.5% 247	21.9% 56	0.0% 0	0.0% 0
Food	3.5% 9	78.1% 200	3.5% 9	14.8% 38
Metal	0.0% 0	0.0% 0	93.0% 238	6.6% 17
Tile	0.0% 0	0.0% 0	3.5% 9	78.5% 201

Figure 13: Confusion Matrix using varSDMM on Vistex



	Macaroni	Corn Flakes	Silver foil	Banana peel	Mustard seed	Plaster
Macaroni	99.0% 99	6.0% 6	0.0% 0	0.0% 0	0.0% 0	1.0% 1
Corn Flakes	0.0% 0	89.0% 89	3.0% 3	0.0% 0	1.0% 1	0.0% 0
Silver foil	1.0% 1	5.0% 5	97.0% 97	5.0% 5	0.0% 0	1.0% 1
Banana peel	0.0% 0	0.0% 0	0.0% 0	95.0% 95	0.0% 0	0.0% 0
Mustard seed	0.0% 0	0.0% 0	0.0% 0	0.0% 0	92.0% 92	1.0% 1
Plaster	0.0% 0	0.0% 0	0.0% 0	0.0% 0	7.0% 7	97.0% 97

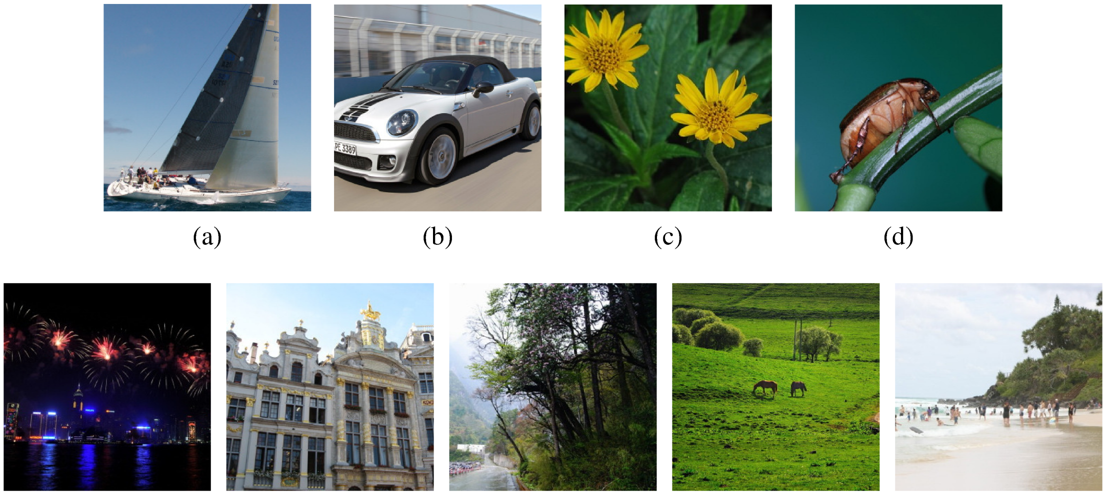
Figure 14: Confusion Matrix using varSDMM on ALOT

3.3.3 Object & Scene Categorization

The task to automatically differentiate random objects has always been frequently discussed in computer vision [37]. Indeed, even similar objects could raise significant problems due to different angles, surrounding environments, and various depth of the captured images. Furthermore, recent research works have addressed related challenging clustering

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

Figure 15: Examples from Caltech. (a) Bikes, (b) Faces, (c) Planes, (d) Camels.



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

(e) (f) (g) (h) (i)

Figure 16: First row: sample objects from GHIM10K - (a) Boats, (b) Cars, (c) Flowers, (d) Bugs. Second row: sample scenes from GHIM10K - (e) Firework, (f) Building, (g) Tree, (h) Grass, and (i) Beach

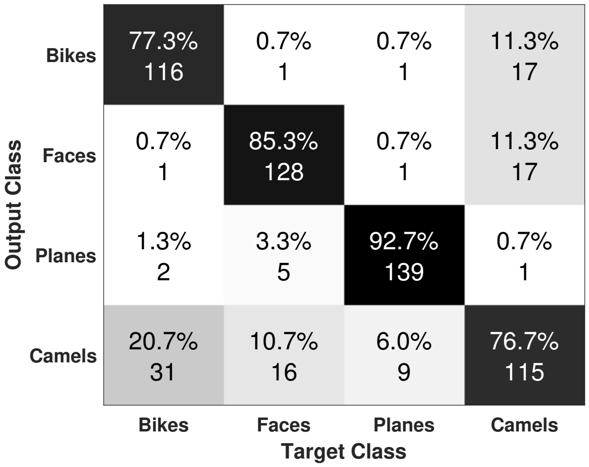
analysis, such as sports activities [4] and scenes [5]. Thus, three object clustering applications are discussed in this experiment, and the efficiency of our model is confirmed by comparison with other novel methods.

We tested our model with two challenging real life data sets: Caltech256 [50] and GHIM10K[2]. In other words, we had a 600-image data set from Caltech256 evenly divided into four classes: Bikes, Faces, Planes, and Camels and a 400-image data set from GHIM10K whose clusters included Boats, Cars, Flowers, and Bugs with 100 images in each cluster. The objects were captured from different angles, distances, lighting conditions, and background environments to elevate the demand of the challenge. We also tested our model with a dataset consisting of five scenes from GHIM10K: Firework, Building,

[2] `http://www.ci.gxnu.edu.cn/cbir/dataset.aspx`

Table 7: Results on object and scene datasets using different models

Method	Accuracy (%)		
	Caltech	GHIM10K - Object	GHIM10K - Scene
varSDMM	83.00	94.25	89.00
varDMM	69.50	83.75	80.54
varGMM	76.00	83.50	71.36
GMM	76.30	83.25	65.81

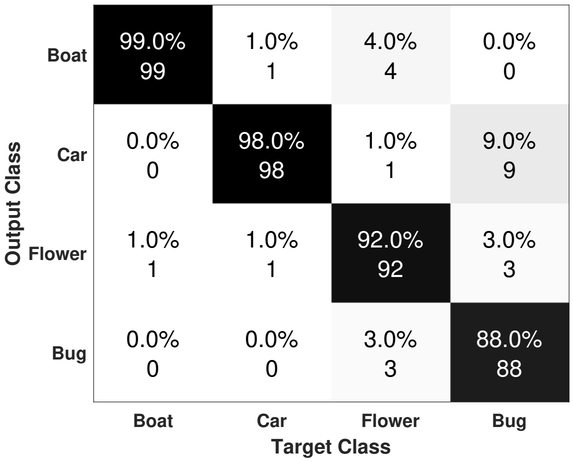


	Bikes	Faces	Planes	Camels
Bikes	77.3% 116	0.7% 1	0.7% 1	11.3% 17
Faces	0.7% 1	85.3% 128	0.7% 1	11.3% 17
Planes	1.3% 2	3.3% 5	92.7% 139	0.7% 1
Camels	20.7% 31	10.7% 16	6.0% 9	76.7% 115

Figure 17: Confusion Matrix using varSDMM on Caltech256

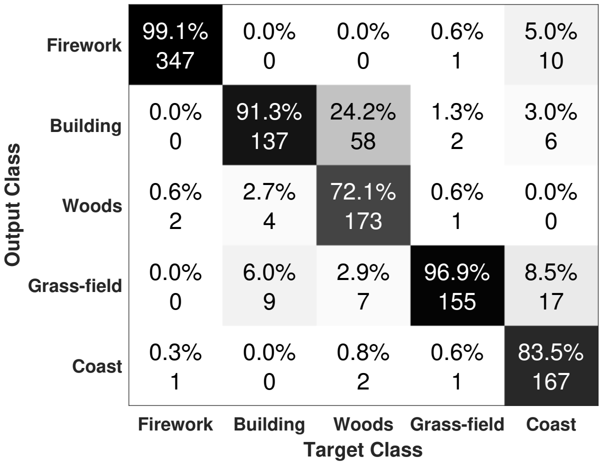
Woods, Grass-field, Coast. Examples from these datasets are presented in Fig. 15 and Fig. 16. The preprocessing step was the same as that described in Section 3.3.1, and the optimal number of vocabulary was also 50.

The accuracy of varSDMM is compared with other widely used models in Table 7, confirming its flexibility and capability to efficiently differentiate various objects in different environments. The confusion matrices for object clustering in Fig. 17 and Fig. 18 validate the performance of the proposed method for this demanding task. In other words, the majority of the objects are accurately clustered despite various complex background noises and different angles. Furthermore, scene clustering is another challenging problem, which contain a large amount of similar details among the scenes. From Fig. 19, it can be observed that a significant portion from group Building is labeled to group Woods, it is due to the fact that buildings are captured with many trees in front causing the missclassification.



Output Class \ Target Class	Boat	Car	Flower	Bug
Boat	99.0% 99	1.0% 1	4.0% 4	0.0% 0
Car	0.0% 0	98.0% 98	1.0% 1	9.0% 9
Flower	1.0% 1	1.0% 1	92.0% 92	3.0% 3
Bug	0.0% 0	0.0% 0	3.0% 3	88.0% 88

Figure 18: Confusion Matrix using varSDMM on GHIM10K for object clustering



Output Class \ Target Class	Firework	Building	Woods	Grass-field	Coast
Firework	99.1% 347	0.0% 0	0.0% 0	0.6% 1	5.0% 10
Building	0.0% 0	91.3% 137	24.2% 58	1.3% 2	3.0% 6
Woods	0.6% 2	2.7% 4	72.1% 173	0.6% 1	0.0% 0
Grass-field	0.0% 0	6.0% 9	2.9% 7	96.9% 155	8.5% 17
Coast	0.3% 1	0.0% 0	0.8% 2	0.6% 1	83.5% 167

Figure 19: Confusion Matrix using varSDMM on GHIM10K for scene clustering

Chapter 4

Online Variational Learning of Finite Scaled Dirichlet Mixture Models

An immense amount of data is created daily through various activities, especially those on social media. Indeed, a method is only considered efficient when it can handle large scale datasets in real time. Motivated by the aforementioned challenge, we introduce an online variational learning approach for Scaled Dirichlet mixture models. As the fundamental structure of finite mixture models has been discussed in Section 2.1, we adopt the idea of stochastic variational inference. In other words, the global knowledge is obtained from the information in an individual observation. Therefore, as new data point coming in, the model's prediction becomes more accurate. The proposed inference is capable of reaching convergence faster than conventional mean field variational inference, which improves the scalability of finite mixture models in order to handle large scale data sequentially in real time. Experiments with object, scene clustering and spam email detection validate the superior performance of our model over other comparable methods.

4.1 Online Variational Bayesian Learning

We introduce an approximated variant $Q\left(\Theta\right)$ of the true posterior distribution $p\left(\Theta \mid \mathcal{X}, \vec{\pi}\right)$. Then, we focus on minimizing the difference between them by using KL divergence as presented in the following equations:

$$\mathcal{L}\left(Q\right) = \ln\ p\left(\mathcal{X} \mid \vec{\pi}\right) - KL\left(Q \mid\mid P\right) \tag{57}$$

where

$$KL\left(Q\mid\mid P\right)=-\int Q\left(\Theta\right)\ln\left(\frac{p\left(\Theta\mid\mathcal{X},\vec{\pi}\right)}{Q\left(\Theta\right)}\right)d\Theta \tag{58}$$

$$\mathcal{L}\left(Q\right)=\int Q\left(\Theta\right)\ln\left(\frac{p\left(\mathcal{X},\Theta\mid\vec{\pi}\right)}{Q\left(\Theta\right)}\right)d\Theta \tag{59}$$

It can be observed that when KL divergence reaches zero, the maximum value of the lower bound $\mathcal{L}\left(Q\right)$ is achieved. Unfortunately, the true posterior distribution is intractable due to its computational complexity. However, we can overcome it by utilizing the mean field theory [32] by factorizing $Q\left(\Theta\right)$ to become $Q\left(\Theta\right)=Q\left(\mathcal{Z}\right)Q\left(\vec{\alpha}\right)Q\left(\vec{\beta}\right)$. Generally, the lower bound $\mathcal{L}\left(Q\right)$ with respect to each distribution $Q_s\left(\Theta_s\right)$ can reach its maximum by:

$$Q_s\left(\Theta_s\right)=\frac{exp\left\langle\ln\ p\left(\mathcal{X},\Theta\right)\right\rangle_{j\neq s}}{\int exp\left\langle\ln\ p\left(\mathcal{X},\Theta\right)\right\rangle_{j\neq s}d\Theta} \tag{60}$$

where $\left\langle.\right\rangle_{j\neq s}$ represents the expectation of all the parameters excluding that case of $j=s$. However, in order to efficiently extend variational framework for online learning, the variational inference is considered as a gradient method [51]. The main idea centralizes the lower bound being a function for the distributions' parameters. In other words, since the model adopts Bayesian inference, the conjugate priors guarantee a functional variant of all factors in the variational posterior probability. Furthermore, as new data are added gradually overtime, the variational lower bound is calculated with respect to a fixed N number of observations. Then, we have the expected value of $p(\mathcal{X})$ in logarithm form as follows:

$$\left\langle\ln p(\mathcal{X})\right\rangle_{\phi}=\int\phi(\mathcal{X})\ln\left(\int p(X\mid\Theta)p(\Theta)d\Theta\right)d\mathcal{X} \tag{61}$$

where $\phi(\mathcal{X})$ is the approximated probability distribution fitting the observed data. Next, the expected value of the lower bound is described as:

$$\begin{aligned}\langle \mathcal{L}(Q)\rangle_{\phi} &= \left\langle \sum_{\mathcal{Z}} \int \left\{ Q(\Omega)Q(\mathcal{Z}) \right.\right. \\ &\times \ln\left[\frac{p(\mathcal{X},\mathcal{Z}\mid\Omega)p(\Omega)}{Q(\Omega)Q(\mathcal{Z})}\right]\left.\left.\vphantom{\int}\right\} d\Omega \right\rangle_{\phi} \\ &= N\int Q(\Omega)d\Omega \left\langle \sum_{\vec{Z}} Q(\vec{Z})\ln\frac{p(\vec{X},\vec{Z}\mid\Omega)}{Q(\vec{Z})}\right\rangle_{\phi} \\ &+ \int Q(\Omega)\ln\left[\frac{p(\Omega)}{Q(\Omega)}\right]d\Omega \end{aligned} \tag{62}$$

where $\Omega = \left\{\vec{\alpha},\vec{\beta}\right\}$. With the size of observed data denoted as t, the estimation for the lower bound corresponding the observed data is given as:

$$\begin{aligned}\mathcal{L}^{(t)}(Q) &= \frac{N}{t}\sum_{i=1}^{t}\int Q(\Omega)d\Omega\sum_{\vec{Z}_i} Q(\vec{Z}_i) \\ &\times \ln\left[\frac{p(\vec{X}_i,\vec{Z}_i\mid\Omega)}{Q(\vec{Z}_i)}\right] + \int Q(\Omega)\ln\left[\frac{p(\Omega)}{Q(\Omega)}\right]d\Omega \end{aligned} \tag{63}$$

Indeed, the main goal is calculating the expected log evidence (61) for an invariant amount of data, which is estimated from the expected lower bound (62). By keeping N fixed while t increases, our online variational framework gradually maximizes the lower bound (63). Particularly, with the observed data $\{X_1, ..., X_{(t-1)}\}$, (63) can be updated for data point X_t corresponding to $Q(\vec{Z}_t)$, while $Q(\Omega)$ and π_j is set to $Q^{t-1}(\Omega)$ and π_j^{t-1}, respectively. Thus, we have the optimal approximation for $Q(\vec{Z}_t)$ as follows:

$$Q\left(\vec{Z}_t\right) = \prod_{j=1}^{M} r_{tj}^{Z_{tj}} \tag{64}$$

$$r_{tj} = \frac{p_{tj}}{\sum_{j=1}^{M} p_{tj}} \tag{65}$$

$$\begin{aligned} p_{tj} = exp&\left\{ \sum_{d=1}^{D}\left[\overline{\alpha}_{jd}\ln\overline{\beta}_{jd} + (\overline{\alpha}_{jd}-1)\ln X_{td}\right] \right. \\ &\left. - \sum_{d=1}^{D}\overline{\alpha}_{jd}\ln\left(\sum_{d=1}^{D}\overline{\beta}_{jd}X_{td}\right) + \tilde{R}_j + \ln\pi_j^{(t-1)} \right\} \end{aligned} \tag{66}$$

$$
\begin{aligned}
\tilde{R}_j =\ & \ln \frac{\Gamma\left(\sum_{d=1}^{D} \overline{\alpha}_{jd}\right)}{\prod_{d=1}^{D} \Gamma\left(\overline{\alpha}_{jd}\right)} \\
& + \sum_{d=1}^{D} \overline{\alpha}_{jd} \left[\psi\left(\sum_{d=1}^{D} \overline{\alpha}_{jd}\right) - \psi\left(\overline{\alpha}_{jd}\right)\right] \times \left[\langle \ln \alpha_{jd} \rangle - \ln \overline{\alpha}_{jd}\right] \\
& + \frac{1}{2} \sum_{d=1}^{D} \overline{\alpha}_{jd}^{2} \left[\psi'\left(\sum_{d=1}^{D} \overline{\alpha}_{jd}\right) - \psi'\left(\overline{\alpha}_{jd}\right)\right] - \left\langle \left(\ln \alpha_{jd} - \ln \overline{\alpha}_{jd}\right)^2 \right\rangle \\
& + \frac{1}{2} \sum_{a=1}^{D} \sum_{b=1, a \neq b}^{D} \overline{\alpha}_{ja}\, \overline{\alpha}_{jb} \Bigg\{ \psi'\left(\sum_{d=1}^{D} \overline{\alpha}_{jd}\right) \\
& \times \left(\langle \ln \overline{\alpha}_{ja} \rangle - \ln \overline{\alpha}_{ja}\right) \times \left(\langle \ln \overline{\alpha}_{jb} \rangle - \ln \overline{\alpha}_{jb}\right) \Bigg\}
\end{aligned}
\tag{67}
$$

Then, with the application of the gradient method, we set $Q(\vec{Z}_t)$ fixed, so that the lower bound (63) is maximized with respect to $Q^{(t)}(\Omega)$ and $\pi_j^{(t)}$. Therefore, the natural gradients are estimated by multiplying the gradients of the parameters with the inverse of the coefficient matrix, which is then removed so that the natural gradients for the posterior probabilities can be computed for an efficient online learning framework. Thus, we have the optimal solutions for parameters' updates:

$$
Q^{(t)}\left(\vec{\alpha}\right) = \prod_{j=1}^{M} \prod_{d=1}^{D} \mathcal{G}\left(\alpha_{jd}^{(t)} \mid \boldsymbol{u}_{jd}^{*(t)}, \boldsymbol{v}_{jd}^{*(t)}\right) \tag{68}
$$

$$
Q^{(t)}\left(\vec{\beta}\right) = \prod_{j=1}^{M} \prod_{d=1}^{D} \mathcal{D}\left(\beta_{jd} \mid \boldsymbol{h}_{jd}^{*(t)}\right) \tag{69}
$$

where

$$
\boldsymbol{u}_{jd}^{*(t)} = \boldsymbol{u}_{jd}^{*(t-1)} + \rho_t \Delta u_{jd}^{*(t)} \tag{70}
$$

$$
\boldsymbol{v}_{jd}^{*(t)} = \boldsymbol{v}_{jd}^{*(t-1)} + \rho_t \Delta v_{jd}^{*(t)} \tag{71}
$$

$$
\boldsymbol{h}_{jd}^{*(t)} = \boldsymbol{h}_{jd}^{*(t-1)} + \rho_t \Delta h_{jd}^{*(t)} \tag{72}
$$

We have the solution for the mixing coefficient $\pi_j^{(t)}$:

$$
\pi_j^{(t)} = \pi_j^{(t-1)} + \rho_t \Delta \pi_j^{(t)} \tag{73}
$$

Where ρ_t denotes the learning rate [52] following the equation:

$$\rho_t = (\eta_0 + t)^{-\epsilon} \tag{74}$$

which $\epsilon \in (0.5, 1]$ and $\eta \geq 0$. The main goal of the learning rate is ignoring the previous incorrect estimations of the lower bound and accelerate the convergence rate. Then, the natural gradients are given as:

$$\begin{aligned}\Delta u_{jd}^{*(t)} &= u_{jd}^{*(t)} - u_{jd}^{*(t-1)} = u_{jd} - u_{jd}^{*(t-1)} \\ &+ Nr_{tj}\overline{\alpha}_{jd}\left[\psi\left(\sum_{d=1}^{D}\overline{\alpha}_{jd}\right) - \psi\left(\overline{\alpha}_{jd}\right)\right. \\ &\left.+ \sum_{d \neq s}^{D}\psi'\left(\sum_{d=1}^{D}\overline{\alpha}_{jd}\right) \times \overline{\alpha}_{js}\left(\langle \ln \alpha_{js}\rangle - \ln \overline{\alpha}_{js}\right)\right]\end{aligned} \tag{75}$$

$$\begin{aligned}\Delta v_{jd}^{*(t)} &= v_{jd}^{*(t)} - v_{jd}^{*(t-1)} = v_{jd} - v_{jd}^{*(t-1)} \\ &- Nr_{tj}\left[\ln \overline{\beta}_{jd} + \ln X_{td} - \ln\left(\sum_{d=1}^{D}\overline{\beta}_{jd}X_{td}\right)\right]\end{aligned} \tag{76}$$

$$\begin{aligned}\Delta h_{jd}^{*(t)} &= h_{jd}^{*(t)} - h_{jd}^{*(t-1)} = h_{jd} - h_{jd}^{*(t-1)} \\ &+ Nr_{tj}\left[\overline{\alpha}_{jd} - \overline{\alpha}_{jd}\overline{\beta}_{jd}\frac{X_{td}}{\sum\limits_{d=1}^{D}\overline{\beta}_{jd}X_{td}}\right]\end{aligned} \tag{77}$$

$$\Delta \pi_j^{(t)} = \pi_j^{(t)} - \pi_j^{(t-1)} = \left(\frac{N}{t}\right) r_{tj} - \pi_j^{(t-1)} \tag{78}$$

where $\psi\left(.\right)$ and $\psi'\left(.\right)$ denote the digamma and trigamma functions, respectively. The expectations in the aforementioned equations are:

$$\Big\langle \ln \alpha_{jd} \Big\rangle = \psi\big(u_{jd}\big) - \ln v_{jd} \tag{79}$$

$$\Big\langle \big(\ln \alpha_{jd} - \ln \overline{\alpha}_{jd}\big)^2 \Big\rangle = \Big[\psi\big(u_{jd}\big) - \ln u_{jd}\Big]^2 + \psi'\big(u_{jd}\big) \tag{80}$$

When a new data point is included, an additional distribution is added to the lower bound. Since the online learning framework can be considered as a stochastic approximation algorithm [53], in which the lower bound may not always increase and the convergence is ensured within the following conditions:

$$\sum_{t=1}^{\infty} \rho_t = \infty, \qquad \sum_{t=1}^{\infty} \rho_t^2 < \infty \tag{81}$$

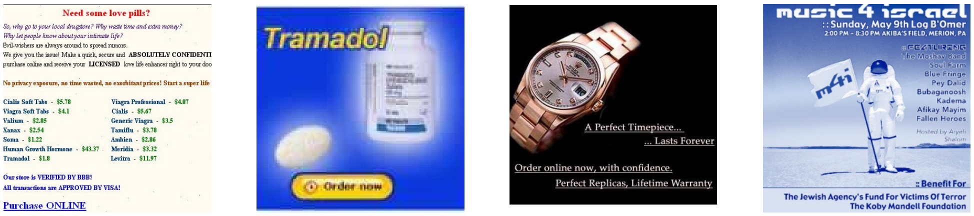
The summary of our model is described in Algorithm 3, in which K-means is used to initialize the parameters with respect to the observed data, then we update the variational solutions by iterating until convergence using EM. In order to achieve the optimal number of components, those with insignificant mixing coefficients (close to 0) are automatically removed.

Algorithm 3 OSDMM Framework

```
Choose an large initial number of components $M$
Initialize values for $\{u_{jd}\}$, $\{v_{jd}\}$, $\{h_{jd}\}$
Initialize $r_{ij}$ using K-Means
for $t = 1$ to $N$ do
    Variational E-Step:
    Update $Q(\mathcal{Z}_t)$ by estimating $r_{tj}$ from (64)
    Variational M-Step:
    Calculate learning rate through (74)
    Compute natural gradients $\Delta u_{jd}^{*(t)}$, $\Delta v_{jd}^{*(t)}$, $\Delta h_{jd}^{*(t)}$, and $\Delta \pi_j^{(t)}$ using (75), (76), (77), and (78), respectively
    Update new variational estimations for $Q^{(t)}(\vec{\alpha})$ (68), $Q^{(t)}(\vec{\beta})$ (69), $\pi_j^{(t)}$ (73)
    Repeat E-step and M-step until new observation is included
end for
```

4.2 Experimental Results

In this section, we validate the performance of OVSDMM with two challenging problems including spam email detection and image clustering. The results are compared with 3 other online variational mixture models using different distributions: Dirichlet (OVDMM), Inveted Dirichlet (OVIDMM), and Gaussian (OVGMM). The preprocessing steps for images consist of 2 main steps: SIFT features extraction and Bag-of-Visual-Words (BoVW)


Need some love pills?
So, why go to your local drugstore? Why waste time and extra money?
Why let people know about your intimate life?
Evil-wishers are always around to spread rumors.
We give you the issue! Make a quick, secure and ABSOLUTELY CONFIDENTI
purchase online and receive your LICENSED love life enhancer right to your doo
No privacy exposure, no time wasted, no exorbitant prices! Start a super life
Cialis Soft Tabs - $5.78
Viagra Soft Tabs - $4.1
Valium - $2.85
Xanax - $2.54
Soma - $1.22
Human Growth Hormone - $43.37
Tramadol - $1.8
Viagra Professional - $4.07
Cialis - $5.67
Generic Viagra - $3.5
Tamiflu - $3.78
Ambien - $2.86
Meridia - $3.32
Levitra - $11.97
Our store is VERIFIED BY BBB!
All transactions are APPROVED BY VISA!
Purchase ONLINE
Tramadol
Order now
A Perfect Timepiece...
... Lasts Forever
Order online now, with confidence.
Perfect Replicas, Lifetime Warranty
music 4 israel
:: Sunday, May 9th Log B'Omer
2:00 PM - 8:30 PM AKIBA'S FIELD, MERION, PA
:: FEATURING
The Moshav Band
Soul Farm
Blue Fringe
Pey Dalid
Bubaganoosh
Kadema
Afikay Mayim
Fallen Heroes
Hosted by Aryeh Shalom
m4i
:: Benefit For
The Jewish Agency's Fund For Victims Of Terror
The Koby Mandell Foundation

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

Figure 20: Examples from (a) Personal Image Spam, (b) SpamArchive, (c) Princeton, and (d) Personal Image Ham.

Table 8: Results of different models on Spambase (%)

Method	Accuracy	Precision	Recall	FPR
OVSDMM	85.27	97.97	72.04	1.49
OVDMM	74.41	86.48	57.86	9.05
OVIDMM	75.48	86.55	60.34	9.38
OVGMM	80.25	84.17	74.52	14.01

construction, which are further explained in Section 4.2.1. The initial number of components is 10 with equal mixing weights. The initialization of the hyperparameters u, v, and h varies with respect to the amount of considered observations as well as the vocabulary size of BoVW. Since our model adopted the iterative scheme EM, the value of initial parameters may affect the overall outcome and the convergence rate rather significantly. Therefore, it is beneficial to test several cases in order to have the optimal initialization.

4.2.1 Spam Detection

Nowadays, we are constantly exchanging information through various mobile messaging applications, and the ubiquitous existence of them has shown their unarguable importance. However, there are many situations where informality can result in devastating consequences. Therefore, emails have been the prominent choice for such occasions [54]. Indeed, the vast usage of emails among co-operations has made it a promising target for various attacks and one of the most financially costly problems. In other words, apart from daily legitimate emails, an immense amount of new spam commercial emails arise along with the demand for additional servers in order to solve the storage problem. Furthermore, spam emails have been the leading inducement for the productivity related decrements of the affected individuals. In addition, spam emails can contain fraudulent schemes beneath

Table 9: Results of different models on image-based spam datasets

Method	Measure (%)	Personal	Spam Archive	Princeton
OVSDMM	Accuracy	88.85	97.04	86.44
	Precision	96.93	98.55	89.18
	Recall	85.41	96.85	74.41
	FPR	4.91	2.61	5.82
OVDMM	Accuracy	87.30	89.52	85.18
	Precision	92.90	96.88	84.68
	Recall	86.94	86.56	75.92
	FPR	12.06	5.09	8.85
OVIDMM	Accuracy	87.06	93.03	74.42
	Precision	92.57	93.05	63.82
	Recall	86.91	96.42	80.15
	FPR	12.67	13.15	29.27
OVGMM	Accuracy	86.72	94.43	81.98
	Precision	91.54	92.95	84.00
	Recall	87.48	98.87	66.70
	FPR	14.67	13.70	8.18

attractive click-baits such as phishing [43], [55], [56]. Thus, an efficient tool to automatically detect spam emails is of the utmost importance.

In this experiment, we challenge our model with a text-based spam dataset Spambase [47] and 3 image-based spams datasets: Personal Image Spam [48], SpamArchive Image Spam [48], and Princeton Spam Image[1]; in which Personal Image Spam contains a non-spam dataset in order to perform cluster analysis. In Spambase dataset, there are 3626 observations, in which half of them is spam and the other half is non-spam. The features are the histograms of the occurrences of the important words. For image-based datasets, we select random sizes for 3 spam datasets: Personal Image Spam (2995 images), SpamArchive Image Spam (3014 images), and Princeton Spam Image (1063 images); where as the common non-spam (ham) email dataset Personal Image Ham has 1650 images. Examples from the aforementioned datasets are shown in Fig. 20. Due to the magnitude of spam detection problems, it is paramount to construct an accurate mathematical collection of common patterns from the dataset. Therefore, we use SIFT (Scale Invariant Feature transform) [41] to extract important features from the images as it has shown its consistency from previous

[1] http://www.cs.princeton.edu/cass/spam/

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

Figure 21: Examples from Corel-10K: (a) Mushroom, (b) Card, (c) Pottery, (d) Egg, and (e) Bead.

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

Figure 22: Examples from GHIM-10K: (a) Firework, (b) Building, (c) Tree, (d) Grass, and (e) Beach.

works [4], [5]. All the $128D$ descriptors extracted by SIFT are concatenated into a collection of local features. Then, K-means is used to cluster the corpus to build the visual words vocabulary, in which the number of visual words is represented by the centroids. Finally, our BoVW is constructed from the histograms of the vocabulary frequencies.

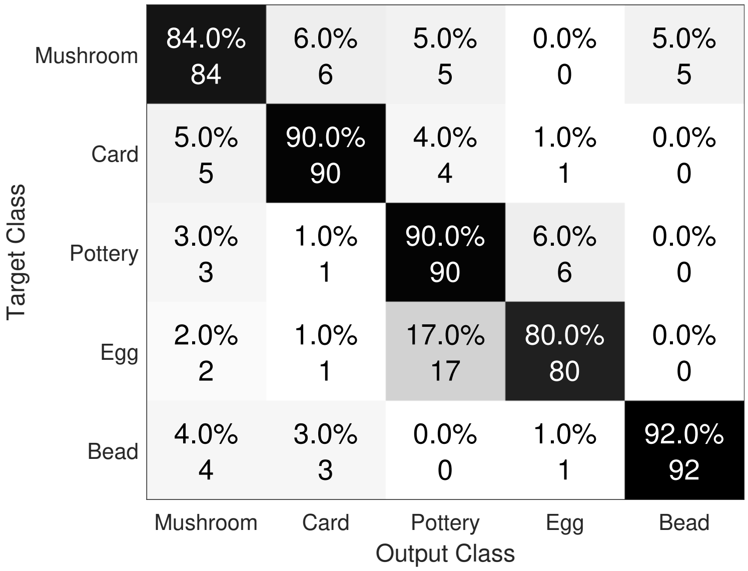
(a) (b) (c) (d) (e)

Figure 23: Examples from 15-Scene: (a) Suburb, (b) Store, (c) Coast, (d) Forest, and (e) Building.

Generally, the performance of relating cluster analysis only considers the Accuracy ($\frac{TP+TN}{TP+TN+FP+FN}$). However, in order to avoid ambiguity, we also include several other metrics: Precision($\frac{TP}{TP+FP}$), Recall($\frac{TP}{TP+FN}$), and False Positive Rate (FPR) ($\frac{FP}{FP+TN}$); in

Table 10: Results of different models on image clustering datasets

Method	Accuracy (%)		
	Corel-10K	GHIM-10K	15-Scene
OVSDMM	87.20	86.63	91.30
OVDMM	80.20	83.54	82.25
OVIDMM	71.00	58.10	84.83
OVGMM	39.60	83.36	48.70



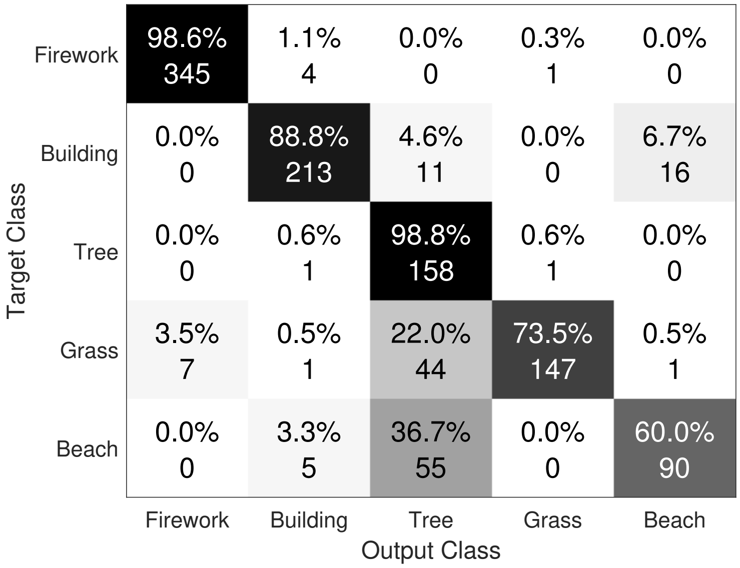
Target Class \ Output Class	Mushroom	Card	Pottery	Egg	Bead
Mushroom	84.0% 84	6.0% 6	5.0% 5	0.0% 0	5.0% 5
Card	5.0% 5	90.0% 90	4.0% 4	1.0% 1	0.0% 0
Pottery	3.0% 3	1.0% 1	90.0% 90	6.0% 6	0.0% 0
Egg	2.0% 2	1.0% 1	17.0% 17	80.0% 80	0.0% 0
Bead	4.0% 4	3.0% 3	0.0% 0	1.0% 1	92.0% 92

Figure 24: Confusion Matrix for Corel-10K dataset using OVSDMM

which we expect that our model can achieve the smallest percentage of FPR meaning the least amount of legitimate emails are incorrectly classified as spams. Indeed, an effective spam detector must ensure both its effectiveness in terms of identifying true spams and its ability to keep number of legitimate emails which are incorrectly classified as spams at minimum. After several tests, the optimal vocabularies for BoVW is 50. The results in Table 8 and Table 9 show that for both text-based and image-based spam clustering tasks, our proposed model not only achieves the highest accuracy, but also has the lowest FPR.

4.2.2 Object & Scene Categorization

Image clustering is among the most challenging topics in computer vision [57], [58], [59], [60]. Indeed, an immense problem when performing cluster analysis is the fact that an



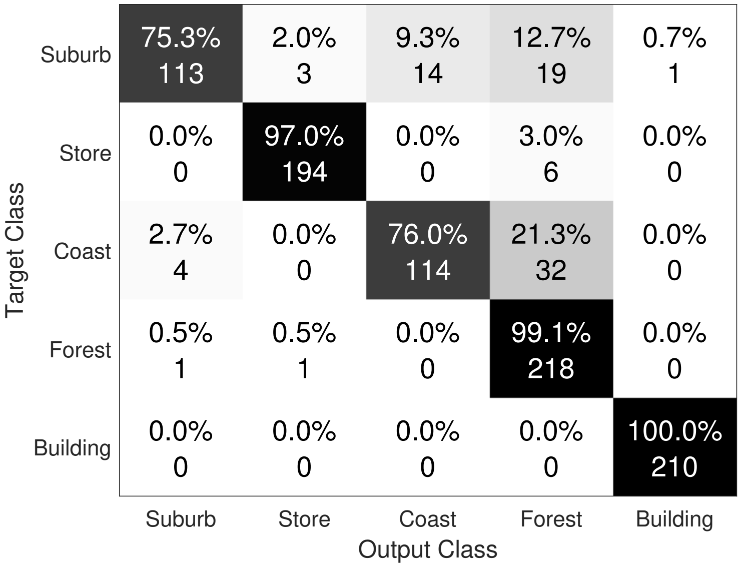
Target Class \ Output Class	Firework	Building	Tree	Grass	Beach
Firework	98.6% 345	1.1% 4	0.0% 0	0.3% 1	0.0% 0
Building	0.0% 0	88.8% 213	4.6% 11	0.0% 0	6.7% 16
Tree	0.0% 0	0.6% 1	98.8% 158	0.6% 1	0.0% 0
Grass	3.5% 7	0.5% 1	22.0% 44	73.5% 147	0.5% 1
Beach	0.0% 0	3.3% 5	36.7% 55	0.0% 0	60.0% 90

Figure 25: Confusion Matrix for GHIM-10K dataset using OVSDMM

observation in real-life environment could be captured in different postures, hues, and distances. Furthermore, noises could also come from background surroundings having similar features as the target object causing higher probability of misclassification. In this experiment, we investigate our model performance not only for object but also scene clustering, for which features extraction is an important step. Recent works on image clustering using finite mixture models have provided good performance which has motivated us to further explore the capabilities of probabilistic models with this challenging task [4], [5].

In our experiments, we considered 3 real-life datasets: Corel-10K [61], GHIM-10K [62], and 15-Scene [63]. Most images are captured in natural environments from different angles along with other items making different scenes having a considerable number of similar features. It is the mixed components that raise as a significant challenge for any interested method. For Corel-10K dataset, we choose 500 images which are evenly into 5 groups: Mushroom, Card, Pottery, Egg, and Bead. Then, we select 5 clusters from GHIM-10K: Firework (350 images), Building (240 images), Tree (160 images), Grass (200 images), and Beach (150 images) making the total sample size of 1100. Finally, there are 930 images in 15-Scene from 5 classes: Suburb (150 images), Store (200 images), Coast (150 images), Forest (220 images), and Building (210 images). Examples from 3 datasets are given in Fig. 21, Fig. 22, and Fig. 23.

The preprocessing steps also include SIFT and BoVW as explained in Section 4.2.1. After several trials, the optimal BoVW size is also 50. Table 10 shows that the proposed



Target Class \ Output Class	Suburb	Store	Coast	Forest	Building
Suburb	75.3% 113	2.0% 3	9.3% 14	12.7% 19	0.7% 1
Store	0.0% 0	97.0% 194	0.0% 0	3.0% 6	0.0% 0
Coast	2.7% 4	0.0% 0	76.0% 114	21.3% 32	0.0% 0
Forest	0.5% 1	0.5% 1	0.0% 0	99.1% 218	0.0% 0
Building	0.0% 0	0.0% 0	0.0% 0	0.0% 0	100.0% 210

Figure 26: Confusion Matrix for 15-Scene dataset using OVSDMM

model is significantly more accurate for cluster analysis with respect to other methods. Furthermore, from Corel-10K confusion matrix in Fig. 24, the misclassification between Pottery and Egg is caused by the fact that many instances in Pottery have oval shapes similar to those in Egg. Likewise, those scenes with a considerable amount of incorrectly clustered images as represented in confusion matrices for GHIM-10K and 15-Scene in Fig. 25 and Fig. 26 all contain similar features related to trees. Thus, with the results from several challenging real-life datasets, OVSDMM's effectiveness has been validated for cluster analysis.

Chapter 5

Conclusion

With the ubiquitous appearances of proportional data in text and multimedia environments, we focus on cluster analysis of this kind of data by developing three effective variational approaches for finite Scaled Dirichlet mixture models. Indeed, previous promising results of Dirichlet distribution on various challenging applications have motivated us to further explore the extent of this family of distributions.

In chapter 2, we have introduced variational learning for finite Scaled Dirichlet mixture models, which follows the idea of minimizing the difference between approximated posterior distribution and the real one using KL divergence. Besides being a statistical inference, the variational framework can also be seen as an optimization process, in which the parameters and model's complexity are estimated simultaneously along with the maximization of the variational lower bound. Through extensive experiments including object and texture images clustering, the proposed model has proven its efficiency by reaching convergence rapidly with accurate estimations.

Then, in chapter 3, component splitting, a local model selection scheme, is employed to provide an elegant approach to determine the optimal number of components. In other words, after successfully applying conventional variational approach for two components, the framework gradually splits the components with the highest mixing weights until all the components no longer satisfy the splitting test. Our method is tested with different real-life challenging applications namely spam detection, image clustering including textures, objects, and scenes. Despite the amount of noise in the observations raising as a significant obstacle, most of the data points are accurately clustered, which validate the performance of our model.

Finally, we have implemented an online approach for finite Scaled Dirichlet mixture models in chapter 4, which adopts the idea of stochastic variational inference in order to efficiently estimate the parameters and model's complexity. In other words, in each iteration, the global knowledge is updated with the local information from analyzing an individual observation. Therefore, the convergence rate is significantly more effective than using mean field variational inference, in which all the data points must be processed in each iteration. Indeed, the application of online learning not only improves the scalability of finite mixture models in order to handle large scale data effectively, but also open the feasibility of dealing with demanding challenges in real time. The performance of our model is tested with different prominent problems such as spam detection, image clustering including objects and scenes.

Thus, the variational framework has proven to be an efficient alternative to conventional Bayesian inference as its ability to guarantee convergence without the computational cost when using other widely used estimation schemes such as MCMC or Laplace approximation. In addition, since the proposed variational inference adopts KL divergence in the optimization process, many other divergences could be utilized to introduce new variants of variational framework such as expectation propagation and belief propagation. Extending to infinite case is also an interesting future work to the proposed methods.

Bibliography

[1] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[2] L. Kaufman and P. Rousseeuw. *Finding Groups in Data: An Introduction To Cluster Analysis*. 1990.

[3] I. Park, R. Sharman, R. Rao, and S. Upadhyaya. The effect of spam and privacy concerns on e-mail users' behavior. *ACM Transactions on Information and System Security*, 2016.

[4] W. Fan, N. Bouguila, and D. Ziou. Variational learning for finite dirichlet mixture models and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 2012.

[5] K. Ihou and N. Bouguila. A new latent generalized dirichlet allocation model for image classification. In *2017 Seventh International Conference on Image Processing Theory, Tools and Applications*, 2017.

[6] J. Wang and J. Chiang. A cluster validity measure with outlier detection for support vector clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B - Cybernetics*, 2008.

[7] C. Aggarwal and C. Reddy. *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC, 2013.

[8] K. He, M. Zhang, J. He, and Y. Chen. Probabilistic model checking of pipe protocol. In *2015 International Symposium on Theoretical Aspects of Software Engineering*, 2015.

[9] X. Xie, W. Huang, H. Wang, and Z. Liu. Image de-noising algorithm based on gaussian mixture model and adaptive threshold modeling. In *2017 International Conference on Inventive Computing and Informatics*, 2017.

[10] M. Azam and N. Bouguila. Speaker verification using adapted bounded gaussian mixture model. In *2018 IEEE International Conference on Information Reuse and Integration*, 2018.

[11] M. Azam and N. Bouguila. Unsupervised keyword spotting using bounded generalized gaussian mixture model with ica. In *2015 IEEE Global Conference on Signal and Information Processing*, 2015.

[12] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B - Methodological*, 1977.

[13] G. McLachlan and D. Peel. *Finite mixture models*. Wiley Series in Probability and Statistics, New York, 2000.

[14] M. Azam and N. Bouguila. Bounded generalized gaussian mixture model with ica. *Neural Processing Letters*, 2018.

[15] I. Channoufi, S. Bourouis, N. Bouguila, and K. Hamrouni. Image and video denoising by combining unsupervised bounded generalized gaussian mixture modeling and spatial information. *Multimedia Tools and Applications*, 2018.

[16] X. Wang and H. Wang and. An improved gaussian mixture model based on least-squares cross-validation and gaussian pso with gaussian jump. In *2012 International Conference on Machine Learning and Cybernetics*, 2012.

[17] S. Banerjee and M. Agrawal. On the performance of underwater communication system in noise with gaussian mixture statistics. In *2014 Twentieth National Conference on Communications*, 2014.

[18] F. Provost and T. Fawcett. Data science and its relationship to big data and data-driven decision making. *Big Data*, 2013.

[19] Z. Xu, K. Kersting, and C. Bauckhage. Efficient learning for hashing proportional data. In *2012 IEEE 12th International Conference on Data Mining*, 2012.

[20] T. Bdiri, N. Bouguila, and D. Ziou. Variational bayesian inference for infinite generalized inverted dirichlet mixtures with feature selection and its application to clustering. *Applied Intelligence*, 2016.

[21] W. Fan and N. Bouguila. A variational component splitting approach for finite generalized dirichlet mixture models. In *2012 International Conference on Communications and Information Technology*, June 2012.

[22] B. Oboh and N. Bouguila. Unsupervised learning of finite mixtures using scaled dirichlet distribution and its application to software modules categorization. In *2017 IEEE International Conference on Industrial Technology*, 2017.

[23] R. Alsuroji, N. Zamzami, and N. Bouguila. Model selection and estimation of a finite shifted-scaled dirichlet mixture model. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018.

[24] S. Bourouis, A. Zaguia, N. Bouguila, and R. Alroobaea. Deriving probabilistic svm kernels from flexible statistical mixture models and its application to retinal images classification. *IEEE Access*, 2019.

[25] B. Alghabashi and N. Bouguila. A Finite multi-dimensional generalized Gamma Mixture Model. In *The 2018 IEEE International Conference on Smart Data*, Halifax, Canada, 2018.

[26] T. Bdiri and N. Bouguila. Positive vectors clustering using inverted dirichlet finite mixture models. *Expert Syst. Appl.*, 2012.

[27] B. Everitt. Maximum likelihood estimation of the parameters in a mixture of two univariate normal distributions; a comparison of different algorithms. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 1984.

[28] W. Dayi, H. Zhe, and L. Yanyan. Study on the wear failure rule of marine diesel engine based on mcmc method. In *2018 IEEE 9th International Conference on Software Engineering and Service Science*, 2018.

[29] S. Fu and N. Bouguila. A bayesian intrusion detection framework. In *2018 International Conference on Cyber Security and Protection of Digital Services*, 2018.

[30] E. Epaillard and N. Bouguila. Variational bayesian learning of generalized dirichlet-based hidden markov models applied to unusual events detection. *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

[31] R. Fujimaki, Y. Sogawa, and S. Morinaga. Online heterogeneous mixture modeling with marginal and copula selection. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, New York, NY, USA, 2011. ACM.

[32] G. Parisi. *Statistical field theory*. Frontiers in physics. Addison-Wesley Pub. Co., 1988.

[33] R. FISHER. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 1936.

[34] D. Dua and E. Taniskidou. UCI machine learning repository, 2017.

[35] T. Braunl, S. Feyrer, W. Rapf, and M. Reinhardt. Texture recognition. 2001.

[36] R. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 1973.

[37] C. Aggarwal. *Data Classification: Algorithms and Applications*. Frontiers in physics. Chapman and Hall/CRC, 2014.

[38] D. Yin, J. Pan, P. Chen, and R. Zhang. Medical image categorization based on gaussian mixture model. In *2008 International Conference on BioMedical Engineering and Informatics*, 2008.

[39] H. Greenspan and A. Pinhas. Medical image categorization and retrieval for pacs using the gmm-kl framework. *IEEE Transactions on Information Technology in Biomedicine*, 2007.

[40] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, 2004.

[41] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

[42] C. Constantinopoulos and A. Likas. Unsupervised learning of gaussian mixtures based on variational component splitting. *IEEE Transactions on Neural Networks*, 2007.

[43] J. Hong. The state of phishing attacks. *Commun. ACM*, 2012.

[44] J. Wang, T. Herath, R. Chen, A. Vishwanath, and H. R. Rao. Research article phishing susceptibility: An investigation into the processing of a targeted spear phishing email. *IEEE Transactions on Professional Communication*, 2012.

[45] C. Pham, L. Nguyen, N. Tran, E. Huh, and C. Hong. Phishing-aware: A neuro-fuzzy approach for anti-phishing on fog networks. *IEEE Transactions on Network and Service Management*, 2018.

[46] Z. Dou, I. Khalil, A. Khreishah, A. Al-Fuqaha, and M. Guizani. Systematization of knowledge (sok): A systematic review of software-based web phishing detection. *IEEE Communications Surveys Tutorials*, 2017.

[47] D. Dua and C. Graff. UCI machine learning repository, 2017.

[48] M. Dredze, R. Gevaryahu, and A. Elias-Bachrach. Learning fast classifiers for image spam. 2007.

[49] G. Burghouts and J. Geusebroek. Material-specific adaptation of color invariant features. *Pattern Recognition Letters*, 2009.

[50] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007.

[51] M. Sato. Online model selection based on the variational bayes. *Neural Computation*, 2001.

[52] M. Hoffman, F. Bach, and D. Blei. Online learning for latent dirichlet allocation. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*. Curran Associates, Inc., 2010.

[53] H. Kushner and G. Yin. *Stochastic Approximation Algorithms and Applications*. 1997.

[54] H. Xu and B. Yu. Automatic thesaurus construction for spam filtering using revised back propagation neural network. *Expert Systems with Applications*, 2010.

[55] T. Churi, P. Sawardekar, A. Pardeshi, and P. Vartak. A secured methodology for anti-phishing. In *2017 International Conference on Innovations in Information, Embedded and Communication Systems*, 2017.

[56] G. Geng, Z. Yan, Y. Zeng, and X. Jin. Rrphish: Anti-phishing via mining brand resources request. In *2018 IEEE International Conference on Consumer Electronics*, 2018.

[57] A. Tariq and H. Foroosh. T-clustering: Image clustering by tensor decomposition. In *2015 IEEE International Conference on Image Processing*, 2015.

[58] X. Zhang, Z. Li, B. Hou, and L. Jiao. Spectral clustering based unsupervised change detection in sar images. In *2011 IEEE International Geoscience and Remote Sensing Symposium*, 2011.

[59] S. Chew and N. Cahill. Normalized cutswith soft must-link constraints for image segmentation and clustering. In *2014 IEEE Western New York Image and Signal Processing Workshop*, 2014.

[60] T. Kinsman, P. Bajorski, and J. B. Pelz. Hierarchical image clustering for analyzing eye tracking videos. In *2010 Western New York Image Processing Workshop*, 2010.

[61] G. Liu, J. Yang, and Z. Li. Content-based image retrieval using computational visual attention model. *Pattern Recognition*, 2015.

[62] G. Liu, Z. Li, L. Zhang, and Y. Xu. Image retrieval based on micro-structure descriptor. *Pattern Recognition*, 2011. Computer Analysis of Images and Patterns.

[63] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.