Multidimensional Proportional Data Clustering Using Shifted-Scaled Dirichlet Model

Rua Tawfiq Alsuroji

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) at

Concordia University

Montréal, Québec, Canada

September 2018

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Rua Tawfiq Alsuroji**

Entitled: **Multidimensional Proportional Data Clustering Using Shifted-Scaled Dirichlet Model**

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 Committee:

_____ Chair
Dr. Amr Youssef

_____ External Examiner
Dr. Lyes Kadem

_____ Examiner
Dr. Abdessamad Ben Hamza

_____ Supervisor
Dr. Nizar Bouguila

Approved by _____
Abdessamad Ben Hamza, Chair
Department of the Concordia Institute for Information Systems Engineering

_____ 2018

_____
Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Multidimensional Proportional Data Clustering Using Shifted-Scaled Dirichlet Model

Rua Tawfiq Alsuroji

We have designed and implemented an unsupervised learning algorithm for a finite mixture model of shifted-scaled Dirichlet distributions for the cluster analysis of multivariate proportional data. The cluster analysis task involves model selection using Minimum Message Length to discover the number of natural groupings a dataset is composed of. Also, it involves an estimation step for the model parameters using the expectation maximization framework. This thesis aims to improve the flexibility of the widely used Dirichlet model by adding another set of parameters for the location (beside the scale parameter)

We have applied our estimation and model selection algorithm to synthetic generated data, real data and software modules defect prediction. The experimental results show the merits of the shifted scaled Dirichlet mixture model performance in comparison to previously used generative models.

Acknowledgments

First and foremost, all the thanks belong to Allah the Almighty. I owe it all to Him, for His grace and protection throughout my study.

Then, I would like to deeply thank several people who supported me during this journey. Starting with my supervisor Prof. Nizar Bouguila. I am very grateful to be under his supervision. He has made the most significant impact on my entire life with his constant encouragement, motivation, support, help, and valuable insight in guiding me. Also, I would like to thank committee members for accepting to appraise my thesis.

I would also like to extend my gratitude to all my lab research colleagues especially to Muhammad Azam who helped me a lot in math. My deep and sincere gratitude goes to Nuha Zamzami, thanking her is not enough for all what she has done for me; her patience, unlimited help, and huge support during my study mean a lot to me.

Moreover, all those who cared about me wherever they are, I appreciate their time and support that lie behind the success of my work, especially my best friend Randah Alharbi, my parents Tawfiq Alsuroji and Faiza Basam. I would never have completed this thesis without their endless love and support. As well as my siblings Rafat; Razan; and Mohammad, my parents in law Fouad Nabrawi and Fatin Sheikh. Finally, a special gratitude and deep love to my husband Mohannad Nabrawi, my daughter Sarah and my upcoming baby, they are a major motivation behind this challenge thank you for being around me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

With the continuous rapid development that our world has experienced in information technology systems such as the World Wide Web, significant advancements have changed the way we think and work. This quick development has accelerated the growth of the amount of available data and consequently the challenge of handling and extracting knowledge from these data.

Machine learning and data mining algorithms are widely used to analyze large datasets in order to discover unknown patterns and extract useful knowledge from them. These algorithms are utilized in many fields such as; medical sciences Khozeimeh, Alizadehsani, et al. (2017), crime-detection Nath (2006), risk assessment Kirkos, Spathis, and Manolopoulos (2007), and products' sales Sun, Choi, Au, and Yu (2008), in order to minimize costs, improve the quality, and boost the number of sales Khozeimeh, Alizadehsani, et al. (2017).

Clustering is among the significant tasks that have been discussed and captured scientists' attention in machine learning and data mining Erman, Arlitt, and Mahanti (2006). Clustering procedures are used in representing the presence of subpopulations within an

overall population, with the identification of the sub-population that each individual observation belongs to Figueiredo and Jain (2002). Model-based methods are clustering approaches that make inference through probabilistic assumptions of the data distributions.

A popular model-based approach in clustering is finite mixture which offers a considerable practical value in modeling heterogeneous data Li and Zhang (2008). This approach has provided a mathematical-basis for statistical modeling of many different phenomena in a wide variety of fields including: astronomy, biology, medicine, economics, and engineering G. McLachlan and Peel (2004).

1.2 Objectives

The main objective of this thesis is expanding the current research on finite mixture modeling. For this reason, we explore the use of finite mixture models in cluster analysis taking in consideration some important issues in developing the learning framework. These issues are:

(1) The challenge of choosing a flexible mixture density for the model based cluster analysis.

(2) The estimation approach for the parameters of the chosen mixture model.

(3) A model selection method which determines the optimal number of clusters that a data set comes from.

(4) Evaluation and validation of the cluster analysis method.

Therefore, we consider a generalization of the Dirichlet distribution called the Shifted Scaled Dirichlet (SSD) that offers better flexibility in modeling multivariate data vectors by employing the use of maximum likelihood estimation approach. We also implement

the Minimum Message Length (MML) as a model selection criterion to estimate the optimal number of clusters inherent within a data set. Moreover, we validate our clustering approach capabilities in some problems from different application areas. This will further elaborate the usefulness of the proposed finite mixture model in several real-life applications, particularly, issues related to the improvement of quality engineering systems.

1.3 Contributions

The major contribution of this work is proposing a finite mixture model based on the shifted scaled Dirichlet distribution which is a generalization of the Dirichlet distribution. The shifted scaled Dirichlet distribution introduces a new set of parameters related to location that can translate a distribution besides the scale and the shape parameters. This allows more flexibility in modeling natural and engineering phenomena.

In this thesis, we first develop an unsupervised algorithm for learning finite mixture models from multivariate proportional data. We then implement a model selection criterion that determines the optimal number of clusters that best describes a given dataset. To evaluate the merits of our approaches, we present our experimental results based on synthetic, real datasets and real-world applications such as detection of fault prone software modules and writer identification.

1.4 Thesis Layout Overview

The organization of this thesis is as follows: Chapter 2 reviews finite mixture modeling approach which is the foundation on which the thesis is built on. Furthermore, we consider model based clustering framework, parameter estimation techniques and some issues regarding model selection, cluster validation and generalization of the Dirichlet distribution.

In chapter 3, we propose and discuss in details our proposed model based on shifted

scaled Dirichlet distribution and elaborate on the explanation of the model parameters estimation process and the model selection criterion.

Chapter 4 is dedicated to present the used datasets and we clearly illustrate the experimental results that we obtained from the testing phase in an organized form, supported with figures and tables that contain the performance measures of our model comparing to some other widely used models.

Finally, chapter 5 summaries the conclusions drawn from the experiments in this thesis, highlights some limitations and challenges then, areas for future work.

Chapter 2

Background

2.1 Model Based Framework for Clustering

A wide range of possible statistical analysis techniques exist and can be used to draw some inferences from our data. Data clustering is one of those common learning methods that help to find a pattern in a collection of unlabeled data according to similarity or intrinsic characteristics. It is the task of assigning objects into groups called clusters in such a way that the samples in the same group are more similar to each other than those in other groups. The absence of the labels distinguishes data clustering techniques, which is called unsupervised learning, from other analysis techniques that belong to supervised learning such as classification. This missing information makes clustering a much more difficult task than classification, both in theory and practice. It is used in many fields, including machine learning, data mining, information retrieval, pattern recognition, image analysis, etc. Through clustering, we can make a complex data set simpler to understand. In addition, data can be compressed using clustering so that it takes less space for storage.

Many approaches have been developed such as, K-means clustering, which is by far the most popular clustering method, hierarchical clustering Johnson (1967); Sneath (1957), spectral clustering algorithms Meila and Shi (2001); A. Y. Ng, Jordan, and Weiss (2002);

Shi and Malik (2000) which are famous for being able to handle irregularly shaped clusters, the K-medoids algorithm Kaufman and Rousseeuw (1987) which primarily uses pairwise similarities while preserving the spirit of K-means clustering, and model-based clustering G. J. McLachlan and Basford (1988) that is based on probability models, such as mixture models. This thesis concentrates on the last mentioned family of clustering methods known as model-based clustering. This method models the density that generates the data directly, and thus the task of clustering becomes one of finding the modes of the probability distribution. A natural way to model this density is to use a mixture of several unimodal densities. This process is known as mixture modeling, and the underlying model is called a finite mixture model.

It is important to know that, the quality of a clustering method depends on the criterion that defines the similarity between data samples. In other words, obtaining a high quality clustering result occurs when the intra-cluster similarity is high and the inter-cluster similarity is low. This similarity criterion is expressed in terms of a distance measure which can be represented as either a probabilistic model or a distance metric in an Euclidean space. Yet, deciding if it is similar enough or good enough is subjective. The absence of clusters labels also makes evaluating the clustering results a difficult process. These methods, knowns as clustering validation techniques, which we talk about them later in section 2.4.

Despite all challenges, clustering is gaining increasing popularity from statistics, computer science and many other areas. In particular, in the case of the model-based clustering, researchers are constantly exploring different probability density distributions that can analyze complex forms of datasets and provides the robustness, flexibility, and ease of use. Therefore, we use our proposed algorithm to cluster and optimize the fit between the dataset we have, and the model we have designed.

2.1.1 Finite Mixture Model

As we are usually incapable to use a single model to find patterns in data sets, the need arose to find a robust method that introduces the idea of using more than one model to better fit and then cluster data sets. Therefore, the model-based clustering approach assumes that data are generated from a mixture of probability distributions, each of which represents a different component. Additionally, the total number of components is countable where we describe it by the word, ”Finite”.

Finite mixture model (FMM) is a probabilistic model that combines two or more density functions. It is useful in various applications, *e.g.,* statistical pattern recognition Figueiredo and Jain (2002). In addition to the robust ground that FMM has in the theory of statistics and probability, they are a natural choice when the data to model is heterogeneous. Moreover, they are flexible in the approximation of any other statistical model G. J. McLachlan and Peel (2000).

By using the mixture model in clustering, the parameters of the probability distribution that can fit the patterns will be associated with data samples. Once this fit is completed, the data samples are assigned to the cluster that has the highest estimated posterior probability.

2.2 Maximum Likelihood Estimation

After proposing the model that we are going to use, we need an estimator for the parameters in the used model. The maximum likelihood estimator (MLE) is a popular choice for the finite mixture model. However, an optimization algorithm should be used to find the MLE numerically when an analytical solution does not exist. Most of these algorithms include calculating the derivatives of the objective function and should take into account the special structure of the FMM.

We can obtain the MLE estimates of the mixture parameters using Expectation Maximization (EM) and related techniques G. J. McLachlan and Peel (2000). The EM algorithm is a common general approach to maximum likelihood in the presence of incomplete data i.e. the assignment variable that indicates the component of a particular data sample is generated from is unknown E. S. Oboh (2016). Hence, the EM is used to fit finite mixture models with gradient ascent to the observed data where the convergence occurs at a Maximum Likelihood Estimate (MLE) of the mixture parameters Figueiredo and Jain (2002).

The MLE helps us to find an optimal value of the mixture model parameter by selecting the optimal parameter value that maximizes the product of the likelihood function of each data sample though applying two steps iteratively that we will discuss in the next section.

2.2.1 Expectation Maximization

As we mentioned in 2.2, the EM algorithm is a common general approach to maximum likelihood in the presence of incomplete data. The early work in Dempster, Laird, and Rubin (1977) presented how EM is used to iteratively compute the maximum likelihood estimate of incomplete data as we mention in 2.2.

The EM algorithm is first initialized with some random model parameters as starting values which is critical for the successful of the mixture parameters estimation. Many works such as E. S. Oboh (2016) and Bdiri and Bouguila (2012) make use of the well-known Kmeans algorithm and the Method Of Moments (MOM) in the task of parameter initialization in order to reduce the possibility of the convergence to local maxima. The moments method relies on low order statistics of the equations of the model distribution that we intend to compute its parameters.

The authors in Giordan and Wehrens (2015) discuss and make a comparison with Bouguila and Ziou (2007); Ronning (1989) works regarding the initialization step where they emphasized the importance of efficient re-parametrization technique which usually

occurs when the parameters become negative or exceed a very large number, that making the EM iteration convergence difficult.

After the initialization step, EM iteratively uses two steps. First, the expectation step (E-step) in which the posterior probability is computed. Second, the maximization step (M-step) where the likelihood function is maximized until convergence.

2.2.2 Newton-Raphson Method

Generally it has been proven in many works that no closed-form solution exists for the maximum likelihood estimate of the Dirichlet model parameter. This problem caused by the non-linearity of the parameter function becomes a challenging optimization problem. This challenge led to the necessity of using an iterative optimization technique such as gradient ascent, Newton Raphson, fixed point iteration, etc. In our work, we make use of the Newton Raphson method which is at present among the most common techniques to find the MLE for the parameter of the Dirichlet distribution since it converges very fast as compared with other optimization techniques Huang (2005).

Newton Raphson methods typically rely on a second-order derivative of the objective likelihood function (i.e. the Hessian matrix), and the inverted matrix is required. Inverting Hessian matrix becomes a very difficult and expensive process when we have high-dimensional data. However, Graybill (1983) introduced an approximation technique that allows an easy approach to invert Hessian matrices.

As we mentioned before, the initialization step is crucially important for the success of the estimation to be inside the parameter range. Besides, in the case of Dirichlet distribution, the parameters have to be non-negative. Therefore, we should consider the methods of moment which is used in Bdiri and Bouguila (2012); Bouguila, Ziou, and Vaillancourt (2004) as well to initialize the Newton-Raphson method.

2.3 Model Selection

A fundamental part of the unsupervised learning problem in mixture modeling is model selection that determines the number of clusters which best describes the data. Model selection is very important because the EM algorithm requires to pre-specify the number of components as an input. However, knowing the number is difficult in practice but fortunately there are many available methods in the case of finite mixture models. These methods are based on the likelihood that help in estimating the number of clusters which is a big advantage only mixture models are able to provide as rigorous reasoning, instead of simple heuristics, for the model selection criteria that they use.

Various model selection approaches have been used by researchers such as cross validation Shao (1993), deterministic methods, hypothesis testing and re-sampling. Since the last two approaches are still expensive to be applicable in computer vision and pattern recognition applications, our interest is on deterministic methods of model selection that can be divided as mentioned in Bouguila and Ziou (2007) into two main classes. The first class is based on the Bayesian approach, for example ; the Schwarzs Bayesian Information Criterion (BIC) and the Laplace Empirical Criterion (LEC), while the second class is based on information/coding theory concepts such as the Minimum Message Length (MML), the Mixture Minimum Description Length (MMDL), Bouguila and Ziou (2005b, 2007); Wallace and Dowe (2000), Akaikes Information Criterion (AIC), and the Minimum Description Length (MDL) criterion Bouguila and Ziou (2005b).

In this thesis, we use the minimum message length which has both Bayesian and information theoretic interpretation in its principle. From the Bayesian perspective, we find the optimal cluster number when maximizing the product between the parameter likelihood and its prior probability Wallace and Dowe (2000), while from an information theoretic perspective, it describes the data with minimal error Wallace and Dowe (2000).

2.4 Cluster Validation

Cluster validation is related to evaluating the goodness of clustering algorithm results Brock, Pihur, Datta, Datta, et al. (2011). Three empirical clustering validation statistics are discussed in Theodoridis, Koutroumbas, et al. (2008) to examine cluster validity which are:

(1) External cluster validation, which depends on comparing the cluster analysis results to an externally supplied class labels (true labels) that have been already known in advance, *e.g.,* entropy. The authors in Bdiri and Bouguila (2012); Bouguila and Ziou (2007); Bouguila et al. (2004) evaluate their clustering algorithm performance with a labeled dataset by using a confusion matrix to calculate some of performance measures such as, overall accuracy, average accuracy, precision, recall, etc.

(2) Internal cluster validation, which uses the internal information of a clustering process to evaluate the goodness of the result by considering how well the clusters are separated and compact without the respect to external information *e.g.,* Sum of Squared Error (SSE), Silhouette coefficient, Dunn index.

(3) Relative cluster validation, which compares different clustering structures by using different parameter values for the same algorithm (*e.g.,* changing the number of clusters k). It is commonly used for determining the optimal number of clusters, *e.g.,* often an external or internal index is used for this function (SSE or Entropy).

This remains an open research topic for both clustering validation and model selection with finite mixture models.

2.5 Generative models for proportional data

2.5.1 Dirichlet Model

The distribution is named after the 19th century Belgian mathematician Johann Dirichlet. It is widely known throughout statistics and probability, Bayesian analysis, statistical genetics, modeling of multivariate data, multivariate analysis, non-parametric inference, reliability theory, characterization problems, and many other areas Gupta and Richards (2001).

The Dirichlet is the multivariate generalization of the Beta distribution where it equals to the probability density function (PDF) of the Beta PDF when the outcomes are two, also it equals the uniform distribution when all parameters $(\alpha_1, \ldots, \alpha_K)$ are equal.

Before the Dirichlet distribution gained its popularity, the normal distribution (Gaussian) was widely applied in most of multivariate data clustering algorithms. While the Gaussian distribution is a probability distribution over all the real numbers, the Dirichlet is a probability distribution over a probability simplex, *i.e.,* it is a probability distribution on the simplex of sets of positive numbers that added up to 1. Therefore, it is closely related to the multinomial distribution which makes it a good candidate to model distributions over distributions or distributions over functions.

Moreover, the symmetry property that Gaussian distribution has, makes it difficult to detect asymmetric patterns in data or to analyze data generated from non-Gaussian sources Medasani and Krishnapuram (1999). Yet, the Dirichlet is flexible and can be applied to asymmetric patterns depending on its shape parameter value Bouguila et al. (2004). Introducing more parameters to the Dirichlet distribution has been main focus of our research work to enhance the flexibility of the model, and release the limitations related to the covariance structure.

2.5.2 Generalization of the Dirichlet Model

Modeling compositional data requires a distribution that can be defined on the bounded domain, the simplex. The most commonly studied distribution on the simplex is the Dirichlet. It gains its popularity from its conjugate property with the multinomial likelihood in Bayesian analysis and its computational efficiency as well as easiness of parameter interpretation. However, it is quite limited to see it in applications due to the most extreme forms of required independence. Ongaro and Migliorati (2013).

Overfitting is crucial issue that happens when a learning algorithm is more accurate in fitting known data and less accurate in predicting new data. the concepts of generalization and overfitting are closely related. Also, it more likely occurs with nonparametric and nonlinear models that have more flexibility of fitting a large number of data forms. Therefore, it is a big challenge to find a model that can better detect unseen data and provide a useful probability without overfitting. Even recognizing the existence of the overfitting problem is in itself a difficult process. This was a main concern of many research efforts Bouguila and Ziou (2006a); G. Monti, Mateu i Figueras, Pawlowsky-Glahn, Egozcue, et al. (2011); Ongaro, Migliorati, Monti, et al. (2008); Pawlowsky-Glahn and Buccianti (2011). In addition to the previous issues, the number of extra parameters that we introduce in building a model that generalizes another model might also cause overfitting.

However, in our case we present a generalization that has two extra parameters to the shape parameter of the Dirichlet, one called the scale parameter which has already been introduced in G. Monti et al. (2011) and implemented in E. S. Oboh (2016), and location parameter that has been introduced in G. Monti et al. (2011) and which we are proposing for clustering problems. This distribution is known as the Shifted Scaled Dirichlet distribution (SSD). Authors in G. Monti et al. (2011) introduce this generalization as a natural generalization of the classical Dirichlet model, *i.e,* the model obtained after applying perturbation and powering to the Dirichlet random composition. This kind of generalization

permits more flexibility in different real-life situations and phenomena.

Chapter 3

Proposed Model

3.1 Shifted-Scaled Dirichlet Distribution

The Shifted-Scaled Dirichlet model is a natural generalization of the Dirichlet distribution obtained after applying the perturbation and powering operations to the classical Dirichlet random composition. These operations define a vector-space structure in the simplex, and play the same role as the sum and product by scalars in real space G. S. Monti, Mateu-Figueras, and Pawlowsky-Glahn (2011). By introducing another set of parameters, we can acquire many useful probability models K. W. Ng, Tian, and Tang (2011). The shifted scaled Dirichlet, subsequently, keeps $(2D + 1)$ degrees of freedom which grant it the flexibility for diverse real data applications Hankin et al. (2010); B. S. Oboh and Bouguila (2017). As stated by G. Monti et al. (2011) when we apply only a power transformation to the classic Dirichlet random composition, the result changes the measure of dispersion around the mean and a scaled Dirichlet is obtained. While applying a power and then a perturbation transformations will result a scaling and a translation to the density, which means that the shifted-scaled Dirichlet is formed once this equally constraint for scaling and location are relaxed (See Appendix A).

As it has been widely known that, the Dirichlet distribution models proportional data.

Therefore, we will show that the shifted scaled Dirichlet distribution can be used as well to model multivariate proportional data constrained on a simplex. Let us define $\mathbf{X} = (x_1, \ldots, x_D)$ as a random vector of proportions, where $\sum_{d=1}^{D} x_d = 1$. Therefore, the probability of $\mathbf{X}$ that follows a shifted scaled Dirichlet distribution ($X \sim p\boldsymbol{SD}^D(\boldsymbol{\alpha}, \boldsymbol{\beta}, \tau)$) with parameters $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_D) \in R_+^D, \beta = (\beta_1, ..., \beta_D) \in S^D, \tau \in R_+$ is given by:

$$\mathcal{SSD}(\mathbf{X}|\theta) = \frac{\Gamma(\alpha_+)}{\prod_{d=1}^{D}\Gamma(\alpha_d)} \frac{1}{\tau^{D-1}} \frac{\prod_{d=1}^{D} \beta_d^{-\frac{\alpha_d}{\tau}} x_d^{(\frac{\alpha_d}{\tau}-1)}}{(\sum_{d=1}^{D} \frac{x_d}{\beta_d}^{\frac{1}{\tau}})^{\alpha_+}} \tag{1}$$

where Γ denotes the Gamma function, $\boldsymbol{\alpha}$ is the shape parameter, $\boldsymbol{\beta}$ is the location parameter, τ is a scale parameter, and $\alpha_+ = \sum_{d=1}^{D} \alpha_d$. These parameters empower our model with the flexibility to fit any data set. The shape parameter $\boldsymbol{\alpha}$ symbolizes the form of the distribution, the scale parameter τ controls how the density plot is spread out, and the $\boldsymbol{\beta}$ follows the location of the data densities.

Assuming that a set $\mathcal{X} = \{\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_N}\}$ composed of data vectors independent and identically distributed (I.I.D) the resulting likelihood is:

$$P(\mathcal{X}|\theta) = \prod_{n=1}^{N} \frac{\Gamma(\alpha_+)}{\prod_{d=1}^{D}\Gamma(\alpha_d)} \frac{1}{\tau^{D-1}} \frac{\prod_{d=1}^{D} \beta_d^{-\frac{\alpha_d}{\tau}} x_{nd}^{(\frac{\alpha_d}{\tau}-1)}}{(\sum_{d=1}^{D} \frac{x_{nd}}{\beta_d}^{\frac{1}{\tau}})^{\alpha_+}} \tag{2}$$

3.1.1 Shape Parameter

The shape parameter (α) that simply represents the form of the shifted scaled distribution where the more flexibility that α has, the better the modeling and clustering are. Figure 3.1 shows a $2D$ density plot with different cases for the shape parameter. First, when the shape is less than 1, we get a convex distribution while in the second case, we have a higher shape parameter that result in concave plots of different shapes.

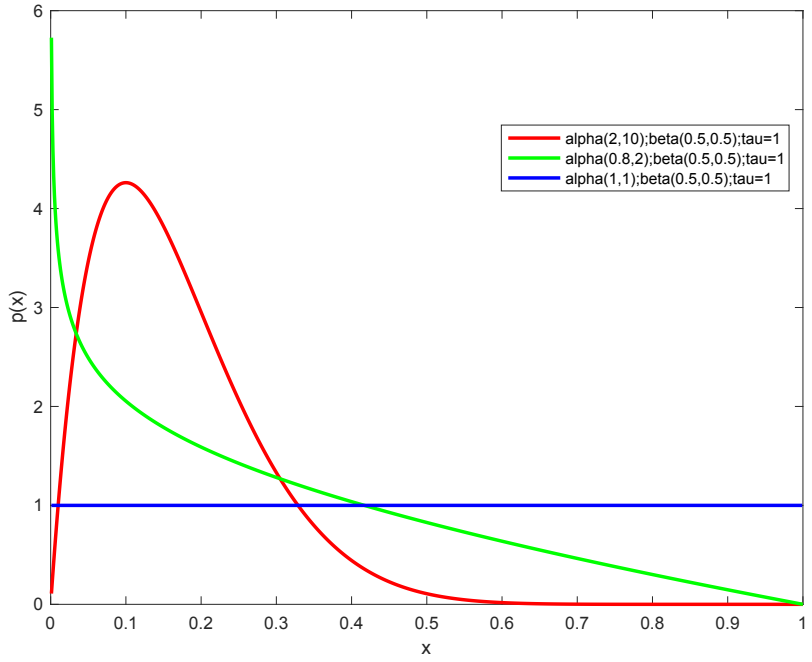

Figure 3.1: Artificial plot describing the properties of the shape parameter

3.1.2 Scale parameter

The scale parameter (τ) which is a scalar that simply stretches or shrinks the distribution, i.e. controlling the density plot spreading out. Regardless of the scale parameter whether it has a constant value of 1 or any higher value, we have realized that the density shape stays the same and the changing in the value does not affect the form. Figure 3.2 shows a $2D$ density plot with different values for the scale parameter. As we can see the different values for the scale affect the spread of the distribution that has the same shape values.

3.1.3 Location parameter

The location parameter (β) that simply shifts the distribution which adds more flexibility to the model in fitting the data and identifying the patterns that a dataset has. Figure 3.3 shows a $2D$ density plot with different values that shows the Dirichlet case, then

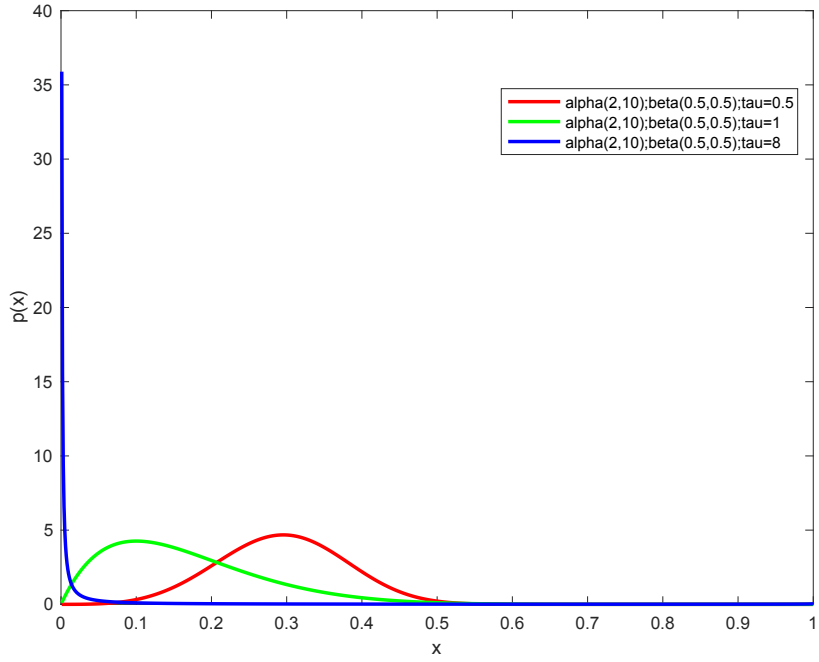

Figure 3.2: Artificial plot describing the properties of the scale parameter

the scaled-Dirichlet, and finally the shifted-scaled Dirichlet which shifts the previous one (scaled Dirichlet) by changing the location of the distributions and shift them according to the parameters values we assign.

3.2 Finite shifted-scaled Dirichlet Mixture Model

A finite mixture model is a convex collection of two or more probability density functions that has the capability in approximating any arbitrary distribution Costa Filho (2008). That is, for a data population $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ with N observations in which each sample is a D-dimensional vector, $\mathbf{X}_n = (x_{n1}, \ldots, x_{nD})$, is modeled in terms of a mixture of several components K that the data population comes from. Each component which is called cluster has a simple parametric form which is in our case, shifted scaled Dirichlet,

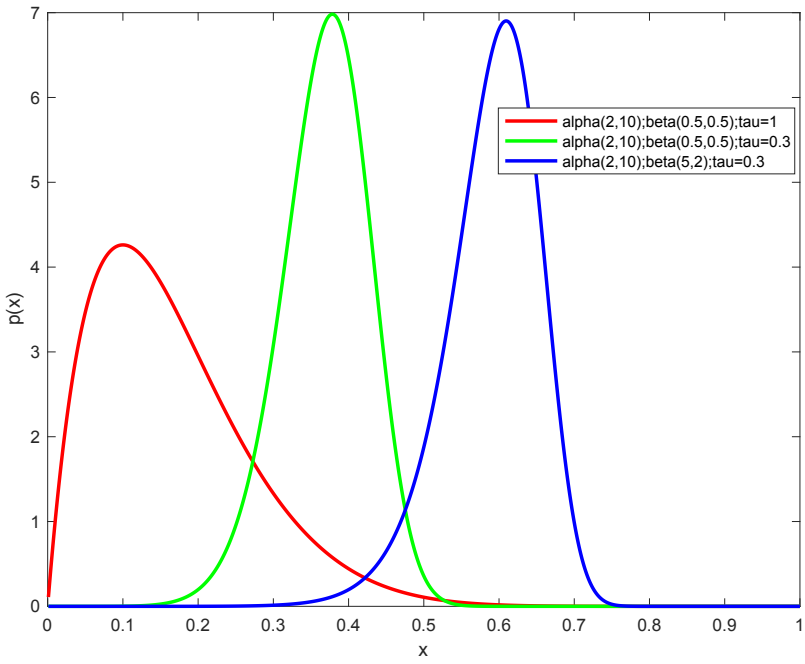

Figure 3.3: Artificial plot describing the properties of the location parameter

and the mixture model is, thus, defined as:

$$p(\mathbf{X}|\Theta) = \sum_{j=1}^{K} \pi_j \, \mathcal{SSD}(\mathbf{X}|\theta_j) \tag{3}$$

where the complete model parameters are denoted by $\Theta = \{\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K\}$ in which $\theta_j = \{\boldsymbol{\alpha_j}, \boldsymbol{\beta_j}, \tau_j\}$ represents the parameter vectors for the jth population, and π_j is the mixing weight satisfying $\sum_{j=1}^{K} \pi_j = 1$, and $0 \leq \pi_j \leq 1$. Therefore, we form the corresponding likelihood of $\mathcal{X}$ with N-observations, assuming that each $\mathbf{X}_n$ is independently distributed as:

$$p(\mathcal{X}|\Theta) = \prod_{n=1}^{N} \sum_{j=1}^{K} \pi_j \, \mathcal{SSD}(\mathbf{X}_n|\theta_j) \tag{4}$$

where the summation inside the product in Eq.(4) prohibits the possibility of analytical solutions.

3.3 Parameters Estimation of the Finite Shifted Scaled Dirichlet Mixture Model

Estimating the model parameters is a really critical issue in a finite mixture modeling. In order to infer each parameter equation, we make use of the maximum likelihood estimation (MLE) approach. MLE has become widely popular and acceptable in solving this problem Costa Filho (2008) through Expectation Maximization (EM) approach on the complete likelihood Dempster et al. (1977) that can be formed as,

$$\Theta^* = \arg\max_{\Theta} \mathcal{L}(\mathcal{X}, \Theta) \tag{5}$$

which is commonly useful in observations that can be viewed as incomplete data Dempster et al. (1977); G. McLachlan and Krishnan (2007). By incomplete data we presume the absence of the assignment variable that refers to a cluster of a particular data sample. Let $\mathcal{Z} = \{Z_1, \ldots, Z_n\}$ denotes the latent variables or hidden assignment, where our prior knowledge about $\mathcal{Z}$ is given only by the posterior distribution $p(\mathcal{Z}|\mathcal{X}, \Theta)$. Since we cannot use the complete-data likelihood, its expected value under the posterior distribution of the latent variable is considered, which corresponds to the E step in the EM algorithm. Thus, unobserved latent variables $\mathbf{Z}_n$ is a K-dimensional binary random vector where $\sum_{j=1}^{K} z_{nj} = 1$, *i.e.* z_{nj} is the hidden membership assignment of each data sample to jth cluster, where $z_{nj} \in \{0, 1\}$, as:

$$z_{nj} = \begin{cases} 1 & \text{if } \mathbf{X_n} \text{ belongs to a component} j, \\ 0 & \text{otherwise}, \end{cases} \tag{6}$$

Therefore, the complete data likelihood is given by:

$$\log(p(\mathcal{X}, Z|\Theta)) = \mathcal{L}(\Theta, \mathcal{X}, Z) = \prod_{n=1}^{N}\sum_{j=1}^{K} z_{nj}\Big((\pi_j) * p(\mathbf{X_n}|\boldsymbol{\alpha_j}, \boldsymbol{\beta_j}, \tau_j)\Big) \tag{7}$$

Then, we replace z_{nj} by its expectation, which we can call it, posterior probability $\hat{z}_{nj}$ as following,

$$\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta) = \prod_{n=1}^{N}\sum_{j=1}^{K} \hat{z}_{nj}\Big(\log(\pi_j) + \log p(\mathbf{X}_n|\boldsymbol{\alpha_j}, \boldsymbol{\beta_j}, \tau_j)\Big) \tag{8}$$

where

$$\hat{z}_{nj} = P(j|\mathbf{X_n}, \theta_j) = \frac{\pi_j p(\mathbf{X_n}|\boldsymbol{\alpha_j}, \boldsymbol{\beta_j}, \tau_j)}{\sum_{j=1}^{K} \pi_j p(\mathbf{X_n}|\boldsymbol{\alpha_j}, \boldsymbol{\beta_j}, \tau_j)} \tag{9}$$

As we mentioned before, we make use of EM algorithm through two steps for learning our mixture model. Firstly, **E-step**, where we compute the posterior probabilities by using Eq. 9. Secondly, **M-step**, where we update the model parameter estimates by maximizing the following:

$$\begin{aligned} \Theta &= \arg\max_{\Theta}\{\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)\} \\ &= \arg\max_{\Theta} \prod_{n=1}^{N}\sum_{j=1}^{K} \hat{z}_{nj}\Big(\pi_j + (\mathbf{X_n}|\theta_j)\Big). \end{aligned} \tag{10}$$

For the purpose of facilitating the parameters estimation process, we do maximize the

log of the likelihood in Eq. 8 that we can express it as:

$$\log(\mathbf{X_n}|\boldsymbol{\alpha_j}, \boldsymbol{\beta_j}, \tau_j) = \log[\prod_{n=1}^{N}(\frac{\Gamma(\alpha_+)}{\prod_{d=1}^{D}\Gamma(\alpha_{jd})} * \frac{1}{\tau_j^{D-1}} * \frac{\prod_{d=1}^{D}\beta_{jd}^{-(\frac{\alpha_{jd}}{\tau_j})}x_{nd}^{(\frac{\alpha_{jd}}{\tau_j})-1}}{(\sum_{d=1}^{D}(\frac{x_{nd}}{\beta_{jd}})^{\frac{1}{\tau_j}})^{\alpha_+}})]$$

$$= \sum_{n=1}^{N}([\log\Gamma(\alpha_+) - \sum_{d=1}^{D}\log\Gamma(\alpha_{jd})]+$$

$$[\log(1) - (D-1)\log(\tau_j)]+$$

$$[(\sum_{d=1}^{D} -(\frac{\alpha_{jd}}{\tau_j}\log(\beta_{jd})) + \sum_{d=1}^{D}(\frac{\alpha_{jd}}{a_j} - 1 * \log(x_{nd}))) - (\alpha_+\log(\sum_{d=1}^{D}(\frac{x_{nd}}{\beta_{jd}})^{\frac{1}{\tau_j}}))]) \tag{11}$$

Then, find Θ_{MLE} when the derivatives are equal to zero. The following subsections describes the whole process.

3.3.1 Mixing weight parameter estimation: π_j

In order to derive its equation, two constraints should be considered, $\sum_{j=1}^{K}\pi_j = 1$ and $0 \leq \pi_j \leq 1$. Therefore, we introduce Lagrange multipliers in terms of finding π_j. Hence, the augmented log likelihood is:

$$\phi(\Theta, Z, X, \Lambda) = \sum_{n=1}^{N}\sum_{j=1}^{K}\hat{z}_{nj}(\log\pi_j + \log(\vec{X_n}|\boldsymbol{\alpha_j}, \boldsymbol{\beta_j}, \tau_j)) + \Lambda(1 - \sum_{j=1}^{K}\pi_j) \tag{12}$$

Taking the derivative of Eq. 12 with respect to π_j for each cluster, we obtain,

$$\pi_j = \frac{1}{N}\sum_{n=1}^{N}\hat{z}_{nj} \tag{13}$$

3.3.2 The Distribution parameters estimation: α_j, β_j, and τ_j

In order to get the parameters equations, we maximize $\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)$ by taking the first derivative with respect to α_{jd}, β_{jd}, and τ_j are calculated respectively in 14, 15, 16 considering β_{jd} constraints $0 \leq \beta_{jd} \leq 1 \, ; \sum_{d=1}^{D} \beta_{jd} = 1$ (See Appendix B).

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)}{\partial \alpha_{jd}} = \sum_{n=1}^{N} \hat{z}_{nj} \Big(\Psi(\alpha_+) - \Psi(\alpha_{jd}) + \\ \frac{\log(x_{nd}) - \log(\beta_{jd})}{\tau_j} - \log(\sum_{d=1}^{D} (\frac{x_{nd}}{\beta_{jd}})^{\frac{1}{\tau_j}}) \Big) \end{aligned} \tag{14}$$

while Ψ is the digamma function (the logarithmic derivative of the Gamma function).

$$\beta_{jd} = \frac{\sum_{n=1}^{N} \hat{z}_{nj} \frac{\alpha_+ x_{nd}}{\tau_j \beta_{jd} \sum_{d=1}^{D} \frac{x_{nd}}{\beta_{jd}}} - \frac{\alpha_{jd}}{\tau_j}}{\sum_{n=1}^{N} \hat{z}_{nj} \sum_{d=1}^{D} \frac{\alpha_+ x_{nd}}{\tau_j \beta_{jd} \sum_{d=1}^{D} \frac{x_{nd}}{\beta_{jd}}} - \sum_{d=1}^{D} \frac{\alpha_{jd}}{\tau_j}} \tag{15}$$

$$\begin{aligned} \tau_j = \sum_{n=1}^{N} \hat{z}_{nj} \frac{\sum_{d=1}^{D} \alpha_{jd} \Big(\log(\beta_{jd}) - \log(X_{nd}) \Big)}{D-1} \\ + \sum_{n=1}^{N} \hat{z}_{nj} \frac{\alpha_+ \log \sum_{d=1}^{D} \frac{x_{nd}}{\beta_{jd}}}{D-1} \end{aligned} \tag{16}$$

As a result of the non-linearity of the first derivative of α parameter, there is no closed-form solution for it. This issue leads to the necessity of an optimization technique to handle it such as, gradient ascent, Newton Raphson, fixed point iteration, etc. In our work, we employ the Newton Raphson method that allows the fastest convergence among other techniques Huang (2005). The Newton Raphson method for $\boldsymbol{\alpha}$ parameter is expressed as:

$$\alpha_j^{new} = \alpha_j^{old} - H^{-1} G \tag{17}$$

Where H is called the Hessian matrix associated with $\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)$, and G is called the gradient which is the first derivatives vector.

Then, by calculating the second and mixed derivatives with respect to α_{jd}, we obtain:

$$\frac{\partial^2 \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)}{\partial^2 \alpha_{jd}^2} = \sum_{n=1}^{N} z_{nj} \Big(\Psi'(\alpha_+) - \Psi'(\alpha_{jd}) \Big) \tag{18}$$

$$\frac{\partial^2 \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)}{\partial^2 \alpha_{jd_1} \alpha_{jd_2}} = \sum_{n=1}^{N} z_{nj} \Psi'(\alpha_+) \tag{19}$$

where Ψ' is the trigamma function (the logarithmic derivative of the digamma function).

$$H(\alpha_{j,d}, \alpha_{j,d}) = \sum_{n=1}^{N} z_{nj} \times \begin{bmatrix} \Psi'(\alpha_+) - \Psi'(\alpha_1) & \dots & \Psi'(\alpha_+) \\ \vdots & \ddots & \vdots \\ \Psi'(\alpha_+) & \dots & \Psi'(\alpha_+) - \Psi'(\alpha_D) \end{bmatrix} \tag{20}$$

Note that, the Hessian matrix should be transformed to its inverse before it can be calculated in the Newton-Raphson maximization step. Yet, we should keep in mind that the Hessian block matrix has to be positive or semi-positive definite before its inverse be computed in our case. Since it is difficult to do that, we do need to relax this constraint by making use of its diagonal approximation. Consequently, this approximation, allows the inverse to be trivially computed.

3.3.3 Initialization and Estimation Algorithm

EM algorithm is very sensitive to the initialization step Gentle (1998); Hu (2015). In this regard, K-means clustering (the most commonly used clustering algorithm) is used to initialize the mixing proportions. Moreover, for initializing $\boldsymbol{\alpha}$ parameter we make use of the method of moments Minka (2000), for initializing $\boldsymbol{\beta}$ parameter, we create a proportions vector (summed to one) , and for initializing $\boldsymbol{\tau}$, we assigned a scalar of 1.

The method of moments estimates the model parameters based on their moment equations. Since a closed form solution to the shifted scaled Dirichlet distribution and its moment equations do not exist, we initialize α by using the moment equation of the Dirichlet distribution.

Initialization Algorithm:

The following is brief steps for the initialization process:

(1) Apply K-means algorithm to the data $\mathcal{X}$ to get the pre-defined K-components and their elements.

(2) Calculate the π_j parameter as, $\pi_j = \frac{\text{No\#Elements in Cluster j}}{\text{No\#Observations}}$.

(3) Apply the method of moment Minka (2000) for each cluster j to get the shape parameter vector α_j.

(4) Initialize the scale parameter τ_j with a scalar one for each j.

(5) Initialize the Location parameter vector β_j with a proportion vector of ones for each j. (where the dimensions summation for each β_j equals to one).

Main Parameters Estimation Algorithm:

The complete algorithm of the shifted scaled Dirichlet mixture parameter estimation can be summarized as:

(1) **INPUT**: Data set $\mathcal{X}$ with D-dimensional N observations, and a determined number of clusters K.

(2) Perform the initializations algorithm.

(3) E Step: Calculate the posterior probability $\hat{z}_{nj}$ of an object assigned to a cluster using Eq. 9.

(4) M Step:

- Update π_j using Eq. 13.
- Update α_j, β_j, and τ_j using Eq.17, 15, and 16 respectively.

(5) Terminate and return the final parameters estimates if the convergence test passed; otherwise go to 3.

3.4 MML Approach for Model Selection

An important part of an unsupervised learning problem concerns determining the number of components which best describes the data Minka (2000). In the previous section, we mentioned that we pre-defined the number of clusters before executing the EM algorithm; however, the goal of model selection is to help us infer the number of optimal clusters. Assuming that our data is fundamentally modeled by a mixture of distributions, we consider the application of Minimum Message Length (MML) principle to solve the problem of model selection since it has been found that MML model selection method ,which is based upon information theory, outperforms many other approaches with a superior performance Bouguila and Ziou (2005a, 2006b). As the name implies, the Minimum Message Length inductive inference is based on evaluating models according to their ability to compress a message containing the data Wallace and Dowe (2000). A high compression is obtained by forming suitable statistical models to code the data where the function of a model or parameter estimate provides a probability distribution Baxter (1996). Each message contains two parts, the first part encodes the model by using the prior information about the model only, whereas the second part encodes the data by using the first part Dowe and Farr (1997). The same model should be used by the sender and receiver in order to have the same probability distribution Wallace and Freeman (1987). The number of bits in which a data, $\mathcal{X}$, should be encoded is called the information content of $\mathcal{X}$. From information-theory point of view,

the optimal number of components is the one that allows the efficient transmission (*i.e.,* with minimum amount of information) of the data from a sender to a receiver Wallace and Freeman (1987). The message length formula for a mixture of distributions is defined by Baxter (1996); Wallace and Dowe (2000) as:

$$\begin{aligned} MessLen \simeq & -\log h(\Theta) - \log p(\mathcal{X}|\Theta) + \frac{1}{2}\log(|F(\Theta)|) \\ & + \frac{N_p}{2}(1 + \log(KN_p)) \end{aligned} \tag{21}$$

where $h(\Theta)$ is the prior probability distribution, $p(\mathcal{X}|\Theta)$ is the likelihood of the complete data, N_p is the number of free parameters to be estimated where it is in our case equal to $K(2D+1)-1$. KN_p is the optimal quantization lattice constant for $\mathbf{R}^{N_p}$ Conway and Sloane (2013). As N_p increases, KN_p tends to the asymptotic value given by $\frac{1}{2\pi e} \simeq 0.05855$ which can be approximated by $\frac{1}{12}$ Bouguila and Ziou (2007). The determinant of the Fisher information matrix $|F(\Theta)|$ is derived by taking the second derivative of the negative log-likelihood Wallace and Dowe (2000).

The estimation of the number of clusters is carried out by finding the minimum message length *MessLemgth* with regards to Θ. Subsequently, we will first develop the determinant of the Fisher information $|F(\Theta)|$ for a mixture of shifted scaled Dirichlet distributions and then propose a prior distribution $h(\Theta)$ about our knowledge of its parameters.

3.4.1 Fisher Information matrix for the Finite Mixture of Shifted-Scaled Dirichlet Distributions:

The Fisher information matrix is sometimes called the curvature matrix since it is the second derivative of the likelihood function. This matrix is the expected value of the Hessian matrix of the logarithm of minus the likelihood of the mixture Bouguila and Ziou (2007). Fisher information matrix is specified as the product of the determinant of the

Fisher information $|F(\theta_j)|$ of the estimated parameters, $\theta_j = (\alpha_j, \beta_j, \tau_j)$ for each component j and the determinant of the Fisher information of mixing weights $|F(\pi_j)|$ which is computed in Baxter and Oliver (2000) as follows:

$$|F(\Theta)| = |F(\pi_j)| \prod_{j=1}^{K} |F(\theta_j)| \tag{22}$$

We determine $|F(\pi_j)|$ for jth cluster as a multinomial distribution with parameters $(\pi_1, \ldots, \pi_K)$ that is calculated in Baxter and Oliver (2000) as:

$$|F(\pi)| = \frac{N}{\prod_{j=1}^{K} \pi_j} \tag{23}$$

where π_j is the mixing weight for each cluster that satisfies two constraints, $\sum_{j=1}^{K} \pi_j = 1$ and $0 \leq \pi_j \leq 1$, and N is the number of data observations. In the case of a mixture model, the Fisher information matrix can be computed as proposed in Figueiredo and Jain (2002) after assigning each data vector to the respective clusters. Assuming that the j^{th} cluster contains $\mathcal{X}_j = \{X_l, \ldots, X_{l+\eta_j-1}\}$ observations where $l \leq N$ and η_j is the observations number in each cluster j with the parameters $\alpha_j, \beta_j, \tau_j$. The negative of the log-likelihood function given the vectors $\theta_j = \{\boldsymbol{\alpha_j}, \boldsymbol{\beta_j}, \tau_j\}$ of a single shifted scaled Dirichlet distribution can be written as:

$$\begin{aligned} -\mathcal{L}(\mathcal{X}_j|\theta_j) &= -\log\Big(\prod_{i=l}^{l+\eta_j-1} p(\mathbf{X}_i|\theta_j)\Big) \\ &= -\sum_{i=l}^{l+\eta_j-1} \log p(\mathbf{X}_i|\theta_j) \end{aligned} \tag{24}$$

Then, computing $|F(\theta_j)|$ by taking the negative of the second derivative of its log-likelihood function as follows:

$$-\frac{\partial^2 \log p(\mathcal{X}|\theta)}{\partial \alpha_{jd}^2} = -nj\Big(\Psi'(\alpha_+) - \Psi'(\alpha_{jd})\Big) \tag{25}$$

$$-\frac{\partial^2 \log p(\mathcal{X}|\theta)}{\partial^2 \alpha_{jd_1}\alpha_{jd_2}} = -nj\Psi'(\alpha_+) \tag{26}$$

$$-\frac{\partial^2 \log p(\mathcal{X}|\theta)}{\partial \beta_{jd}^2} = -nj[\frac{\alpha_{jd}}{\tau_j \beta_{jd}^2}] + \sum_{n=1}^{N}[\alpha_+ \frac{x_{nd}^2}{\tau_j \beta_{jd}^6 (\sum_{d=1}^{D} \frac{x_{nd}}{\beta_{jd}})^2}] \tag{27}$$

$$-\frac{\partial^2 \log p(\mathcal{X}|\theta)}{\partial^2 \beta_{jd_1}\beta_{jd_2}} = -nj[\frac{1}{\tau_j}] + \sum_{n=1}^{N}[\alpha_+ \frac{x_{nd}^2}{\tau_j (\sum_{d=1}^{D} \frac{x_{nd}}{\beta_{jd}})^2}] \tag{28}$$

$$\begin{aligned} -\frac{\partial^2 \log p(\mathcal{X}|\theta)}{\partial \tau_j^2} &= -nj\Big([\frac{D-1}{\tau_j^2}] - \sum_{d=1}^{D}[\frac{\alpha_{jd}}{\tau_j^4}\log \beta_{jd}]\Big) \\ &- \sum_{n=1}^{N}\Big(\sum_{d=1}^{D}[\frac{\alpha_{jd}}{\tau^4}\log x_{nd}] - [\alpha_+ \frac{\log \sum_{d=1}^{D} \frac{x_{nd}}{\beta_{jd}}}{\tau^4}]\Big) \end{aligned} \tag{29}$$

Note that, we make use of the Fisher diagonal approximation when $d_1 = d_2 = d$ for our parameters derivatives to avoid some numerical problems that could be occur while computing the whole matrix.

Because $F(\theta_j)$ has a block structure for each component, we have computed the determinant of each block matrix using the solution provided in Powell (2011), where in our case it is a $(2D+1) \times (2D+1)$ block matrix.

As soon as we get the Fisher information for a single shifted scaled Dirichlet distribution, we are able to use it for calculating the Fisher information for a mixture of our distribution as following,

$$\log |F(\Theta)| = \log(N) - \sum_{j=1}^{K} \log(\pi_j) + \sum_{j=1}^{K} \log |F(\theta_j)| \tag{30}$$

3.4.2 Prior Distribution:

MML criterion performance success is dependent on the choice of prior distribution $h(\Theta)$ for the parameters of the shifted scaled Dirichlet mixture model. However, we do not have a prior knowledge about the mixture parameters, We should assign distributions that better describe our prior knowledge of the vectors of mixing parameter and the parameter vectors of the shifted scaled Dirichlet finite mixture model taking into account that these parameters are independent of each others Bouguila and Ziou (2007), which we can represent it as follows,

$$h(\Theta) = h(\pi)\ h(\alpha)\ h(\beta)\ h(\tau) \tag{31}$$

Both the mixing weight π and the location parameter β are defined on the simplex where $\sum_{j=1}^{K} \pi_j = 1$, and $\sum_{d=1}^{D} \beta_d = 1$. Thus, a symmetric Dirichlet distribution with parameters $\varphi = (\varphi_1, \ldots, \varphi_K)$, or $\varphi = (\varphi_1, \ldots, \varphi_D)$, is a natural choice as a prior for the mixing probabilities and the location parameter, respectively, and defined as,

$$h(\pi_1, \ldots, \pi_K) = \frac{\Gamma(\sum_{j=1}^{K} \varphi_j)}{\prod_{j=1}^{K} \Gamma(\varphi_j)} \prod_{j=1}^{K} \pi_j^{\varphi-1} \tag{32}$$

$$h(\beta_1, \ldots, \beta_D) = \frac{\Gamma(\sum_{d=1}^{D} \varphi_d)}{\prod_{d=1}^{D} \Gamma(\varphi_d)} \prod_{d=1}^{D} \beta_{jd}^{\varphi-1} \tag{33}$$

choosing $\varphi = 1$ gives a uniform prior density choice as follows, Baxter and Oliver (2000); Wallace and Dowe (2000):

$$h(\pi) = \Gamma(K) = (K-1)\,! \tag{34}$$

$$h(\beta) = \Gamma(D) = (D-1)\,! \tag{35}$$

In addition, the absence of other knowledge about the shape parameter α_{jd}, we assume that

the components of $\boldsymbol{\alpha_j}$ are independent as well:

$$h(\alpha) = \prod_{j=1}^{K} h(\alpha_j) = \prod_{j=1}^{K} \prod_{d=1}^{D} h(\alpha_{jd}) \tag{36}$$

As a result, the principle of ignorance under the uniform distribution is used for the prior, as it is shown experimentally in Bdiri and Bouguila (2012), over the range of $[0, e^6 \frac{\|\alpha_j\|}{\alpha_{jd}}]$. Then, α_{jd} is the estimated parameter vector and $\|\hat{\alpha_j}\|$ is the norm of the shape vector. We choose the to use a simple uniform prior, which is known to give good results according to Ockham's razor Bouguila and Ziou (2006b); Jefferys and Berger (1992),

$$h(\alpha_{jd}) = e^6 \frac{\alpha_{jd}}{||\alpha_j||} \tag{37}$$

As we mentioned, the scale parameter is a scalar that we can give it a prior value equals to:

$$h(\tau_j) = \frac{1}{10} \tag{38}$$

Therefore, substituting the log of the prior for the shape, location, and scale parameters in Eq. (31), gives the prior probability of the shifted scaled Dirichlet mixture parameters. The log of the prior distribution is given by:

$$\begin{aligned} log(h(\Theta)) = & \sum_{j=1}^{K-1} \log(j) - 6KD - D \sum_{j=1}^{K} \log(||\alpha_j||) \\ & + \sum_{j=1}^{K} \sum_{d=1}^{D} \log(\alpha_{jd}) + \sum_{d=1}^{D-1} \log(d) + \log(\frac{1}{10}) \end{aligned} \tag{39}$$

The finite mixture of shifted scaled Dirichlet distributions message length is obtained by substituting Eqs. (30) and (39) into Eq. (21).

3.4.3 Complete Learning Algorithm

We can re-write "Main Parameters Estimation Algorithm" mentioned in 3.3.3 to have the complete algorithm of our mixture model estimation together with the MML as:

(1) **INPUT**: D-dimensional data set $\mathcal{X}$ with N observations for each K candidate value.

(2) Perform the Initialization algorithm in 3.3.3.

(3) Apply EM algorithm of the mixture model as mentioned in steps 3:5 in 3.3.3.

(4) Calculate the associated criterion $MML(K)$ using Eq.21.

(5) Select the optimal model K^* such that: .

$$K^* = \arg\min_K MML(K)$$

Chapter 4

Experimental Results

4.1 Overview

In this chapter, an examination for the performance of the Shifted Scaled Dirichlet finite Mixture Model (SSDMM) takes place. We test our model in comparison with some popular models such as, classical Dirichlet Mixture Model (DMM), Scaled Dirichlet Mixture Model (SDMM) E. S. Oboh (2016), and Gaussian Mixture Model (GMM). To evaluate the proposed model, we have considered the following:

(1) Synthetic datasets.

(2) Real datasets.

(3) Software modules defect-prone prediction.

(4) Writer identification classification.

The performance is measured by its ability to estimate model parameters, specifying the number of clusters within datasets, and having a good clustering result. The measures we used to evaluate the proposed clustering approach is discussed in next section.

4.2 Performance Measures

To validate our learning algorithm performance, we make use of the confusion matrix which is also known as the error matrix. This method is suitable because we know the labels of the used datasets.

	Real Value	
	TP	FP
Predicted Positive	True Positive	False Positive
	FN	TN
Predicted Negative	False Negative	True Negative

We define the terms as follow,

- **True Positive**: The number of samples correctly marked as positive.
- **True Negative**: The number of samples correctly marked as negative.
- **False Positive**: The number of samples incorrectly marked as positive (type1 error).
- **False Negative**: The number of samples incorrectly marked as negative (type2 error).

The perfect case, which is hardly occurring, when we obtain a diagonal matrix with only true positive and true negative values. There are many indicators or scores that can be computed from the confusion matrix and we use some of them, for example:

- **Overall Accuracy**, which calculates how accurate is our predictive model.

$$\text{OverallAccuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{40}$$

- **Average Accuracy**, which calculates the average of each accuracy per class.

$$\text{Avg. Accuracy} = \frac{1}{M} \sum_{m=1}^{M} \frac{TP}{NumInClass}; \text{ where } M \text{ is number of clusters} \tag{41}$$

- **Precision (at Macro Level)**, that measures how frequently it is correct when the prediction is yes.

$$\text{Precision } = \frac{1}{M}\sum_{m=1}^{M}\frac{TP}{TP+FP};\ \text{where } M \text{ is number of clusters} \tag{42}$$

- **Recall (at Macro Level)**, which is also known as true positive rate that measures the proportion of the actual positives that are correctly identified.

$$\text{Recall } = \frac{1}{M}\sum_{m=1}^{M}\frac{TP}{TP+FN};\ \text{where } M \text{ is number of clusters} \tag{43}$$

- **False Alarm**, which is also known as false positive rate that measures how frequently it is wrong when the prediction is yes.

$$\text{False Alarm } = \frac{FP}{TP+FP} \tag{44}$$

4.3 Synthetic data sets

We have implemented our model on one-dimensional and multi-dimensional synthetic data. The purpose of using synthetic data is to objectively evaluate our learning algorithm performance with known model parameters and mixture components. Thus, we test our algorithm through various synthetic datasets that have different parameter vectors and number of mixture components known a priori. Moreover, we create plots to describe the shape and surface of the synthetic datasets used to show our learning algorithm capabilities. It is also important to note that the synthetic data were generated with constant β, τ parameters.

4.3.1 One-dimensional data

We make use of the synthetic data generated from a Dirichlet mixture and let our algorithm learn its shape parameters while setting the other parameters (location β and scale τ) vectors to a constant value of one. Table 4.1 shows the first generated dataset with real and estimated parameters, where Figure (4.1-a) displays three well separated mixture components while Figure (4.1-b) displayed the three components overlapping. Moreover, Figure (4.3-a) shows how the MML was able to determine the exact number of clusters within the dataset with 3 components.

k	n_j	d	Real parameters' values				Estimated parameters' values			
			p_j	α	β	τ	p_j	α	β	τ
1	1000	1	0.33	2	0.5	1	0.33	2	0.5	1
		2		10	0.5			9.98	0.5	
2	1000	1	0.33	20	0.5	1	0.33	19.67	0.5	1
		2		20	0.5			20.29	0.5	
3	1000	1	0.34	10	0.5	1	0.34	10.4	0.5	1
		2		2	0.5			2.07	0.5	

Table 4.1: One-dimensional synthetic data

4.3.2 Multi-dimensional data

We show here examples of multi-dimensional datasets ($D = 3$) that we have generated. Data are created from two, three, and four shifted scaled Dirichlet densities with different parameters. The values of the real and estimated parameters are shown in Table (4.2) for the 2, 3, and 4-components, respectively, where in Figure (4.2), we display well separated mixtures components. Additionally, Figure (4.3)-(b, c, and d) show how the MML was able to determine the exact number of clusters within the 3D datasets with 2, 3, and 4 components.

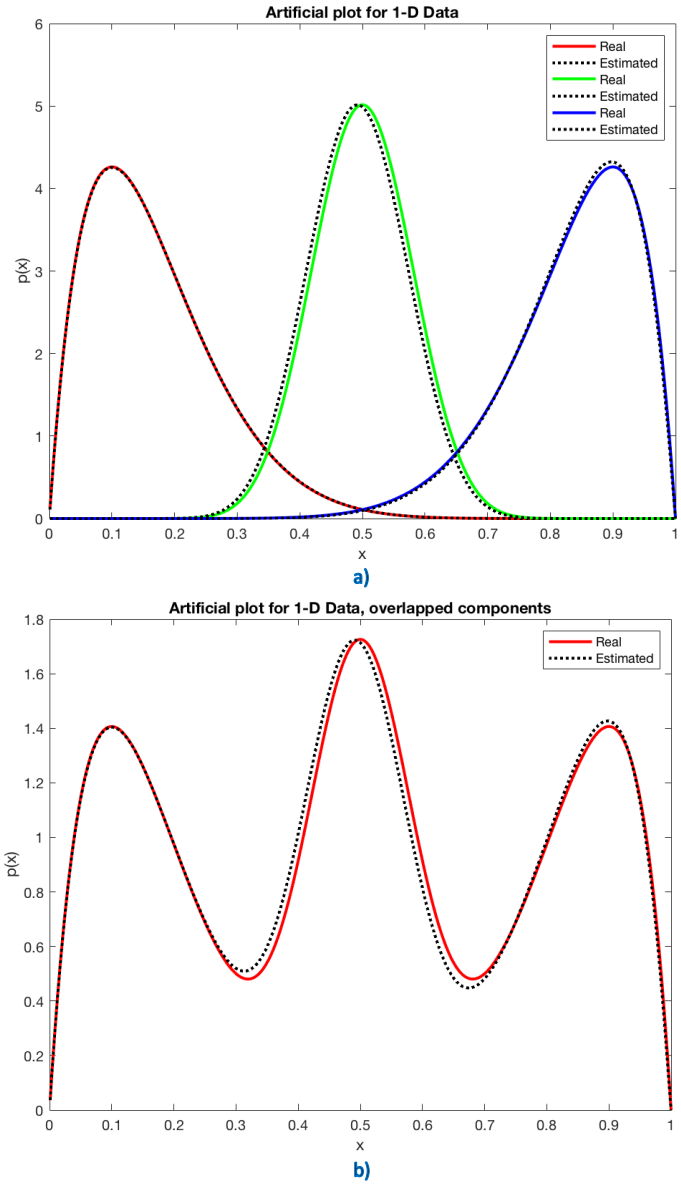

Figure 4.1: One-dimensional generated synthetic dataset plot

k	n_j	d	Real parameters' values				Estimated parameters' values			
			p_j	α	β	τ	p_j	α	β	τ
1	1000	1	0.5	20	0.33	1	0.5	19.95	0.33	1
		2		30	0.33			29.98	0.33	
		3		18	0.34			17.98	0.34	
2	1000	1	0.5	15	0.33	1	0.5	15.12	0.33	1
		2		10	0.33			09.90	0.33	
		3		30	0.34			30.04	0.34	
1	450	1	0.33	65	0.33	1	0.35	64.89	0.33	1
		2		15	0.33			15.28	0.33	
		3		30	0.34			29.66	0.34	
2	450	1	0.33	30	0.33	1	0.32	30.24	0.33	1
		2		34	0.33			33.75	0.33	
		3		35	0.34			35.09	0.34	
3	450	1	0.34	25	0.33	1	0.33	25.26	0.33	1
		2		65	0.33			65.22	0.33	
		3		30	0.34			29.72	0.34	
1	500	1	0.17	16	0.33	1	0.23	16.09	0.33	1
		2		19	0.33			19.26	0.33	
		3		21	0.34			20.79	0.34	
2	1000	1	0.33	18	0.33	1	0.25	18.29	0.33	1
		2		43	0.33			43.53	0.33	
		3		21	0.34			20.25	0.34	
3	1000	1	0.33	43	0.33	1	0.29	42.69	0.33	1
		2		30	0.33			30.37	0.33	
		3		18	0.34			17.86	0.34	
4	500	1	0.17	30	0.33	1	0.23	29.74	0.33	1
		2		21	0.33			20.98	0.33	
		3		20	0.34			20.26	0.34	

Table 4.2: Multi-dimensional synthetic data with 2,3, and 4 clusters.

4.4 Real Datasets

We consider five real data sets; the first one related to life science (Iris flower dataset or Fisher's Iris dataset[1] dataset), where the second and third are related to medical science (Haberman's Survival dataset[2], and Immunotherapy dataset[3]), then the fourth one is

[1] https://archive.ics.uci.edu/ml/datasets/iris

[2]https://archive.ics.uci.edu/ml/datasets/Haberman's+Survival

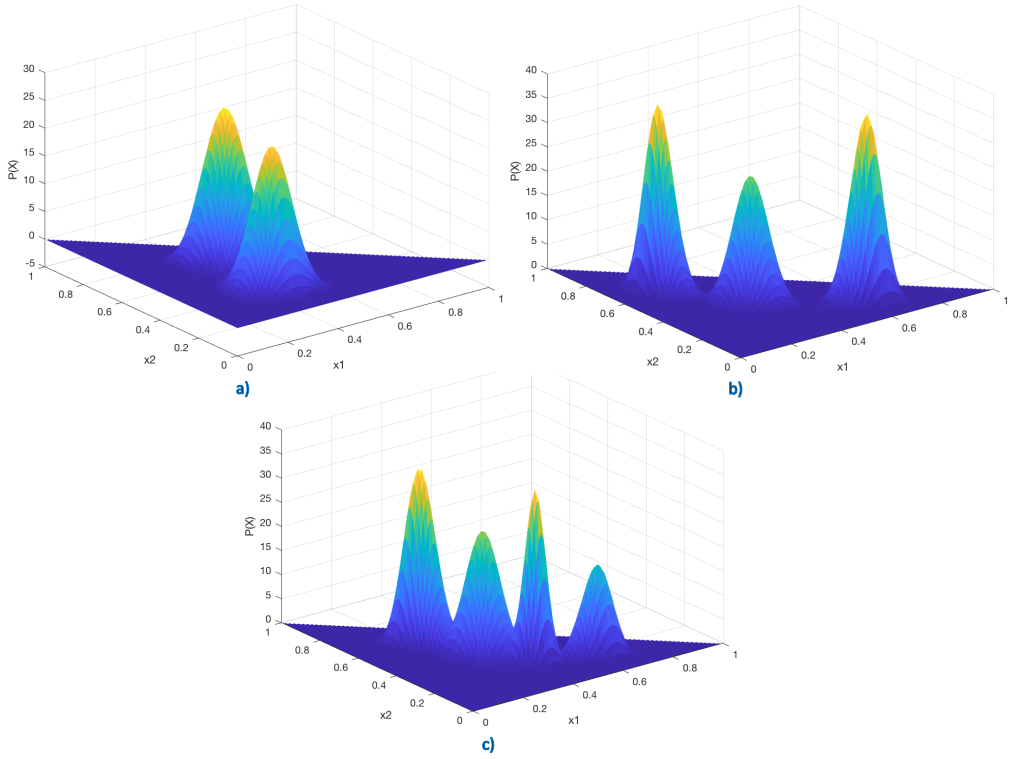

Figure 4.2: Multi-dimensional generated synthetic dataset plot

related to the business field (Absenteeism at work dataset[4]) to detect the reasons behind the employees absenteeism, and finally Wholesale Customers dataset[5] to find meaningful customer segments within a data population.

First, **Iris flower dataset** (Iris dataset for short) is a multivariate dataset that was introduced in 1936 by the British statistician and biologist Ronald Fisher as an example of linear discriminant analysis Fisher (1936). Iris dataset is mostly used for testing machine learning algorithms. The data set has 150 rows in which each represents an iris flower by 4 attributes, including its species and dimensions (length and width) of its botanical parts (sepal and petal) in centimetres. These observations are classified into 3 groups, Iris Setosa, Iris Versicolour, and Iris Virginica where each has 50 rows equally.

Secondly, **Haberman's survival** dataset (Haberman dataset for short) which was first

[3] https://archive.ics.uci.edu/ml/datasets/Immunotherapy+Dataset
[4]https://archive.ics.uci.edu/ml/datasets/Absenteeism+at+work
[5]https://archive.ics.uci.edu/ml/datasets/wholesale+customers

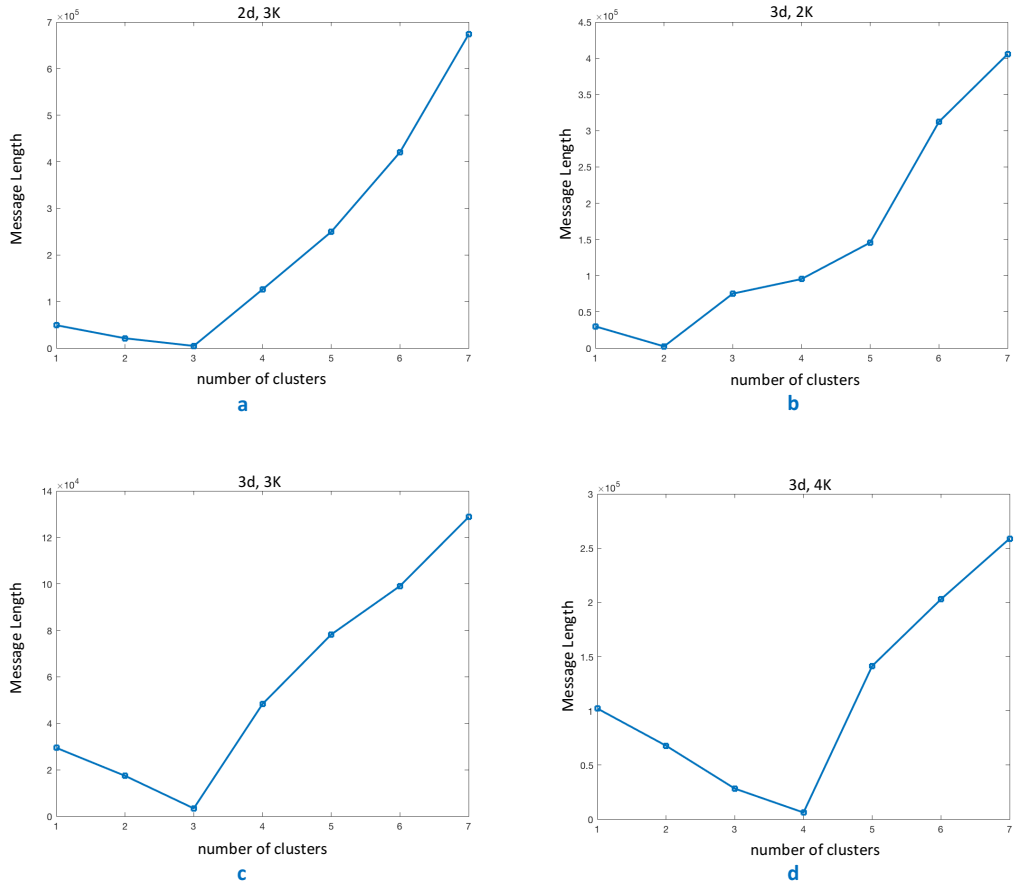

Figure 4.3: Message length plot for the generated synthetic datasets

introduced by R. A. The dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. The 306 survival patients samples are described with four features, Age of patient at the time of operation, Patient's year of operation, Number of positive axillary nodes detected, and Survival status that divide the dataset into two classes where 225 belong to first class for the patient who survived 5 years or longer , 81 belong to second class for the patient who died within 5 year.

Thirdly, **Immunotherapy** dataset is a new dataset conducted in the domain of wart treatment and collected in the dermatology clinic of Ghaem Hospital in Mashhad from January 2013 to February 2015 Khozeimeh, Alizadehsani, et al. (2017). Immunotherapy is

a new treatment method which has lately been employed where the aim of this dataset to diagnose its results and see if this treatment method has better results for each patient than other suggested methods Khozeimeh, Jabbari Azad, et al. (2017) which would help these patients spend less time and money. The dataset was collected from 90 patients with plantar and common warts, who had referred to the dermatology clinic and has 8 features. These patients are classified into 2 groups represent the Response to Treatment feature where the classes has 71 rows for positive response and 19 rows for Negative responses.

Fourthly, **Absenteeism at work** dataset (Abs@work dataset for short) which was collected during the period from July 2007 to July 2010 in a Courier company in Brazil. The high competition among the organizations in the market increases the pressure on the employees to achieve superior goals against the competitors. This compression leads some employees to acquire disturbance in the state of health which is related to the type of work activity. The aim of this dataset is to predict the reasons behind the absenteeism at work where the data has 740 instances that classified by the International Classification of Diseases into 21 categories.

Finally, **Wholesale customers** dataset (Sales dataset for short). This kind of application is widely seen in marketing where the inference would help companies in making better decisions regarding budget, amount/ type of goods to supply to serve a particular customer segment that would increase market share and bottom line for such businesses. The data set source from Lisbon, Portugal Abreu et al. (2011) and it concerns the annual customers' expenses (in monetary units) on product categories: grocery, fresh/frozen/delicatessen products, milk products, detergents and paper products. It has 440 customers of wholesale grouped into two segments based on their spending patterns. The first group, 298 customers from the Horeca (Hotel/Restaurant/Cafe) channel and the second is 142 customers from the Retail channel. As mentioned in Baudry, Cardoso, Celeux, Amorim, and Ferreira (2012), they are distributed into two large Portuguese cities regions (Lisbon and Oporto)

and a complementary region. The wholesale data also includes a questionnaire responses evaluating possible managerial actions with the potential impact on sales such as improving the store layout, offering discount tickets or extending products assortment. The customers answers were registered about whether those actions have impacts on their purchases.

Before comparing our model performance with others, we use the MML to select the number of clusters. As it is presented in Fig.(4.4) a,b,c, d and e for Iris, Haberman, Immunotherapy, Abs@work, and Wholesale customers respectively, the MML was able to determine the optimal number of clusters for all datasets.

After that, we run each algorithm, GMM, DMM, SDMM, and SSDMM 100 times and report the overall accuracy, average accuracy, Precision and Recall at Macro level, and false alarm each with standard errors. The results for the four datasets are shown in the Table (4.3). As we can see, for Iris dataset, the SSDMM outperforms other models with accuracy of 95.33% compared to 94.67% for SDMM, 89.33% for DMM and 82.00% for GMM. Then, for Haberman dataset, the SSDMM performs better than other models with accuracy of 75.49% compared to 73.53% for SDMM, 62.75% for DMM and 66.67% for GMM. Whereas for Immunotherapy dataset, over again the SSDMM together with DMM are better than others with overall accuracy of 88.89% compared with the equal overall accuracy of 74.44% for both SDMM and GMM; however, SDMM has less false alarm which is considered better. Then, for Abs@work dataset, GMM outperforms others with 98.38% followed by a competitive accuracy value for SSDMM with 96.89%. Finally, for Sales dataset, the SSDMM performs again better than other models with accuracy of 84.32% compared to 81.82% for SDMM, 77.27% for DMM and 78.18% for GMM.

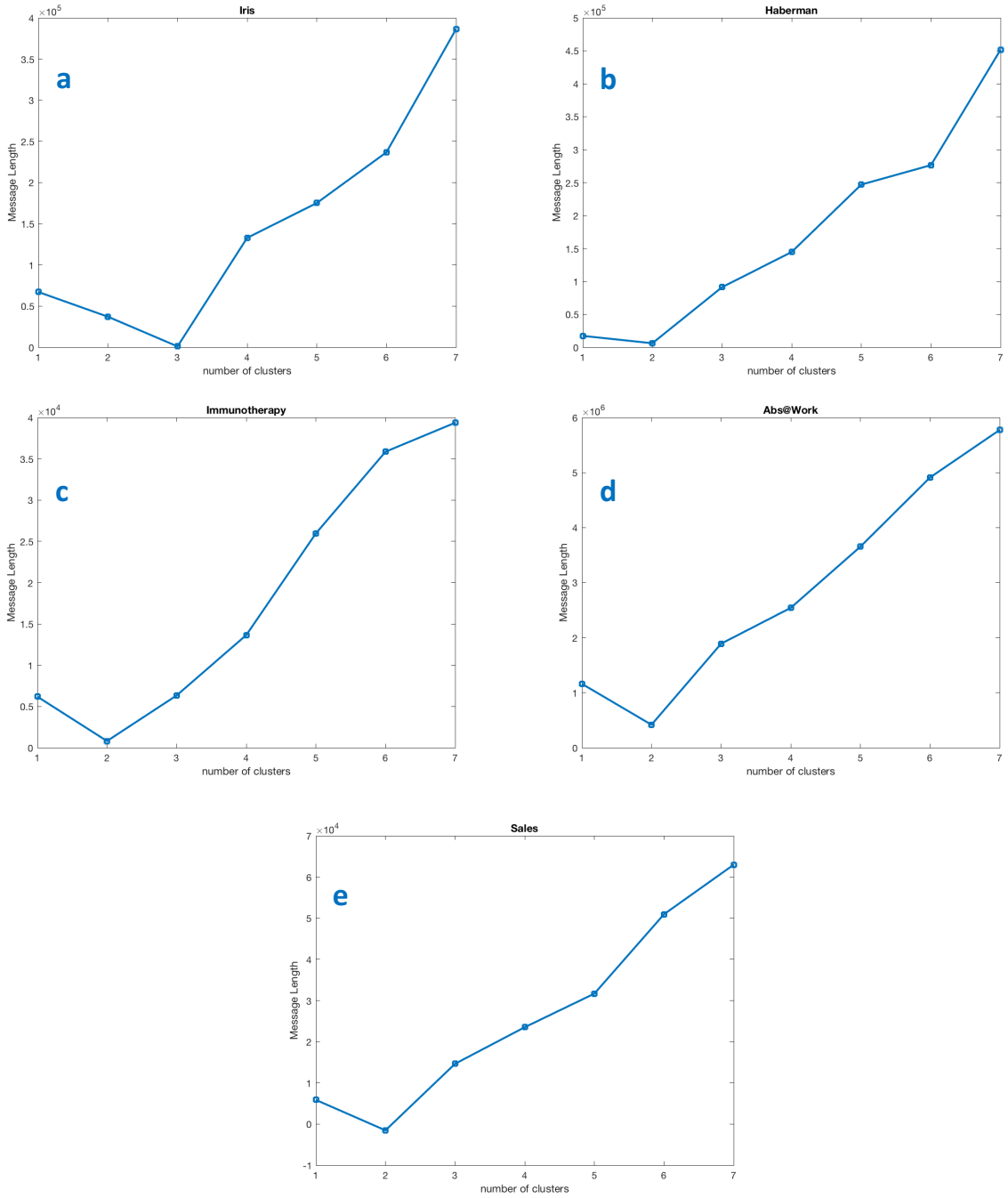

Figure 4.4: Message length plot for the five real datasets

Dataset	Model	Overall Acc.	Avg. Acc.	Macro Precision	Macro Recall	False Alarm
Iris	SSDMM	95.33%±0.0077	95.33%±0.0143	95.33%±0.0077	95.34%±0.0090	0.0467
	SDMM	94.67%±0.0083	94.67%±0.0144	94.67%±0.0083	94.71%±0.0114	0.0533
	DMM	89.33%±0.0063	89.33%±0.0146	89.33%±0.0063	91.92%±0.0087	0.1067
	GMM	82.00%±0.0304	82.00%±0.0321	82.00%±0.0304	82.06%±0.0303	0.1800
Haberman	SSDMM	75.49%±0.0228	62.00%±0.0388	62.00%±0.0048	67.66%±0.0167	0.2093
	SDMM	73.53%±0.0226	63.83%±0.0350	63.83%±0.0062	65.25%±0.0139	0.1949
	DMM	62.75%±0.0190	57.08%±0.0263	47.41%±0.0098	46.67%±0.0173	0.2771
	GMM	66.67%±0.0167	66.67%±0.0214	66.67%±0.0167	63.30±0.0134	0.1525
Immunotherapy	SSDMM	88.89%±0.0144	88.89%±0.0084	88.89%±0.0084	88.97%±0.0208	0.1111
	SDMM	74.44%±0.0244	60.67%±0.0142	60.67%±0.0142	61.66%±0.0187	0.1549
	DMM	88.89%±0.0168	88.98%±0.0083	88.98%±0.0083	89.14%±0.0216	0.1102
	GMM	74.44%±0.0173	49.11%±0.0067	49.11%±0.0025	47.62%±0.0045	0.5089
Abs@work	SSDMM	96.89%±0.0037	96.82%±0.0034	96.82%±0.0034	97.10%±0.0138	0.0318
	SDMM	91.89%±0.0065	89.46%±0.0025	89.46%±0.0025	92.79%±0.0147	0.1054
	DMM	78.65%±0.0089	79.16%±0.0088	79.16%±0.0088	84.78%±0.0166	0.2084
	GMM	98.38%±0.0006	98.14%±0.0006	98.14%±0.0006	98.60%±0.0005	0.0186
Sales	SSDMM	84.32%±0.0176	84.00%±0.0100	84.00%±0.0100	81.86%±0.0173	0.1568
	SDMM	81.82%±0.0281	77.73%±0.0055	77.73%±0.0055	79.66%±0.0118	0.0281
	DMM	77.27%±0.0304	77.88%±0.0076	77.88%±0.0076	75.04%±0.0138	0.0304
	GMM	78.18%± 0.0242	74.68%±0.0229	74.68%±0.0292	75.06%±0.0256	0.0242

Table 4.3: Classification results for Real datasets

4.5 Software Modules Defect-prone Prediction

Due to the increasing number of software errors and defects, many researchers have tackled the challenging problem of predicting errors Catal (2011); Jiang, Cukic, and Menzies (2007); Najadat and Alsmadi (2012); Shihab (2012), where the predictions help to figure out the potential future defects Shihab (2014). It is most likely that when we say a faulty software program, the fault is located in some of the modules[6] not all.

The authors in, Shihab (2012) discuss the importance of historical datasets in detecting fault prone software modules. Therefore the unavailability of these kind of datasets makes the process more difficult. On the other hand, it is important to select the appropriate metrics which explain the attributes of these software modules. Hence, it would help to effectively classify the fault-prone software modules.

The datasets used in this study are the four mission critical NASA software projects, which are obtained from NASA public MDP (Modular toolkit for Data Processing) repository that has 13 projects which are publicly accessible[7]. Each dataset contains 21 software metrics (independent variables) which are 5 different lines of code measure, 3 McCabe metrics, 4 base Halstead measures, 8 derived Halstead measures, a branch-count, and 1 associated dependent Boolean variable for predicting whether the module is defective or not, rather than how many defects it contains. The Halsteads and McCabes complexity measures are useful metrics that can be computed early during the software program design and implementation stages. They are based on the characteristics of the software modules as explained in McCabe (1976).

The McCabes metric includes the following:

(1) Essential complexity.

(2) Cyclomatic complexity.

[6] A module is the smallest independent unit of a software that performs a certain function

[7] http://promise.site.uottawa.ca/SERepository/datasets-page.html

(3) Design complexity.

(4) Number of lines of code.

While the Halsteads complexity metric contains:

(1) Base measures.

(2) Derived measures.

(3) Line of code (LOC) measures.

Two of the datasets are CM1 (NASA spacecraft instrument) and PC1 (Flight software for an earth orbiting satellite) which are from software projects written in a procedural language (C), where a module in this case is a function. The other two datasets are KC1 (which is system implementing storage management for receiving and processing ground data) and KC3 (Collection, processing and delivery of satellite metadata) which are from projects written in object-oriented languages (C++ and Java) where a module in this case is a method. Table 4.4 summarizes the main properties of the considered datasets.

Data set	language	samples	non-defects	defects
CM1	C	498	449	49
KC1	C++	2109	1783	326
KC3	JAVA	458	415	43
PC1	C	1109	1032	77

Table 4.4: Summarized NASA Datasets Properties

For evaluation, we have used some common performance measures such as Average accuracy, precision and recall averaged at Macro level, and False alarm to assess and compare different prediction models quantitatively. Table (4.5) presents the results SSDMM, SDMM, and DMM. Each algorithm was relatively run 100 times with different random initializations and for each dataset, the average metrics with standard errors are reported.

Dataset	Model	Overall Accc.	Avg. Acc.	Precision	Recall	FA
CM1	SSDMM	80.32%±0.0309	61.82%±0.0204	61.82%±0.0204	57.27%±0.0301	0.0787
	SDMM	80.30%±0.0339	61.80%±0.0585	61.80%±0.0056	57.24%±0.0155	0.0730
	DMM	29.92%±0.0337	60.23%±0.0496	60.23%±0.0053	55.57%±0.0111	0.0988
KC1	SSDMM	72.04%±0.0185	72.44%±0.0172	71.53%±0.0172	62.12%±0.0279	0.0658
	SDMM	70.51%±0.0271	71.53%±0.0261	71.53%±0.0106	62.12%±0.0130	0.0658
	DMM	72.02%±0.0254	70.98%±0.0257	70.98%±0.0069	62.24%±0.0088	0.0717
KC3	SSDMM	62.45%±0.0309	73.02%±0.0426	73.02%±0.0077	57.94%±0.0193	0.0235
	SDMM	87.48%±0.0313	73.02%±0.0329	73.02%±0.0118	57.94%±0.0176	0.0235
	DMM	65.28%±0.0342	74.59%±0.0155	74.59%±0.0177	58.62%±0.0086	0.1368
PC1	SSDMM	75.65%±0.0331	61.68%±0.0183	61.68%±0.0020	54.17%±0.0205	0.0496
	SDMM	75.65%±0.0340	50.47%±0.0884	50.47%±0.0044	50.31%±0.0077	0.0688
	DMM	72.32%±0.0311	61.69%±0.0850	61.69%±0.0080	53.78%±0.0080	0.0486

Table 4.5: Classification results for NASA four datasets

As we can see in table 4.4 and for each dataset, we have more non-defective modules than the defectives ones, *i.e.,* we encounter imbalanced classes. Therefore, using "overall accuracy" is not effective in our case. However, the average accuracy gives us better assessment measure. Table (4.5) shows that SSDMM performs almost as good as SDMM in CM1 and KC3 datasets, but for PC1 SSDM is comparable to DMM, while for KC1 the SSDMM performs the best.

For CM1, the average accuracy gives the result of 61.82% for SSDMM and 61.80% for SDMM which is better than 60.23% in case of DMM. While for KC1, the average accuracy is 72.44% for SSDMM and 71.53% SDMM which is better than 70.98% in case of DMM. Moreover, for KC3 dataset, the same result of 73.02% is obtained from both SSDMM and SDMM against 74.59%. For PC1, SSDMM and DMM provide almost the same result of 61.68% and 61.69% respectively which are better than 50.47% for SDMM. According to the false alarm, SSDMM once again achieved comparable results indicated by the small values presents (where the smaller is the better), as (0.0787) for CM1, (0.0658) for KC1, (0.0235) for KC3, and (0.0496) for PC1.

Generally, we can say the experimental results show that SSDMM is better than or at

least is comparable to other generative models and more flexible to predict the defects. Although the focus of this work is on software defect prediction, we believe that the proposed model can be efficiently used in many other applications where the data is in proportional form.

4.6 Writer identification

The importance of handwritten documents has still retained its place in this paperless world; however, the problem lies in identifying the writers' authentication. A certain degree of stability exists behind each writing style of an individual, that makes it possible to identify the personality of the person who has written. For this reasons, several researches in the recent years have investigated the Writer identification problem and many approaches have been proposed to distinguish the author of a document Christlein, Gropp, Fiel, and Maier (2017); He, Wiering, and Schomaker (2015); Wu, Tang, and Bu (2014)). The necessity to identify the author is a widespread problem that emerges often in some fields more than others. For example, the field of medicine where the prescription should come from an authorized doctor, the court of justice where a document authenticity has to be concluded, the library where ancient documents can be analyzed for indexing and retrieval, and in banks for the verification of signatures.

In this section, we use SSDMM to model a persons handwriting where the objective is to identify the writer of a sample among N given writers. Firstly, each handwriting image is segmented into lines regions where for each writer, the lines were splitted randomly into two halves; one for training and one for identification. Second, SIFT is used in the training stage to detect the key points and extract the descriptors Lowe (2004). The features are formed by computing the gradient at each pixel in a 16×16 window around the detected key points. In each 4×4 quadrant, a gradient orientation histogram is formed by

adding the weighted gradient value to one of eight orientation histograms resulting in 128-dimensional descriptor vector. Third, after extracting the features from the training dataset, they are used to generate a codebook by quantizing resulting distribution of descriptors into a number of homogeneous clusters using unsupervised clustering approach, typically a k-means algorithm as proposed in Elkan (2003), where the centroid of each cluster is treated as a visual word. Then, in the identification stage, the extracted descriptors are assigned to the closest visual word (Euclidean distance) resulting in a histogram of frequencies that can be normalized and used for the identification task based on the different tested methods.

Our experiments are based on handwritten text pages from public datasets: two English datasets, IAM Marti and Bunke (2002), and Firemaker Bulacu, Schomaker, and Vuurpijl (2003), and two Arabic datasets, KHATT Mahmoud et al. (2012), and IFN/ENIT database Pechwitz et al. (2002). The IAM dataset includes $1,539$ English handwriting document images written by 657 writers, with 158 writers owning 3 or more handwriting samples. The Firemaker dataset contains $1,000$ handwriting pages written by 250 writers, four pages for each. The KHATT database is composed of unconstrained handwritten Arabic texts written by $1,000$ different writers developed jointly by research groups from KFUPM, Saudi Arabia, TUDortmund, Germany, and TU-Braunschweig, Germany. Finally, IFN/ENIT database composed of $26,549$ images of Tunisian town/village names written by 411 writers and was developed by the Institute of Communications Technology (IFN) at Technical University Braunschweig in Germany and The National School of Engineers of Tunis (ENIT). Figure (4.5) presents sample handwriting images from each dataset used in our experiments.

As we can see in table 4.6, SSDMM almost outperforms other models with accuracy of 98.17% for IAM dataset, 91.46% for Firemaker dataset, and 87.53% for KHATT dataset. Finally, for IFN/ENIT dataset, GMM is slightly better with 79.82% as compared to SSDMM with 78.95%.

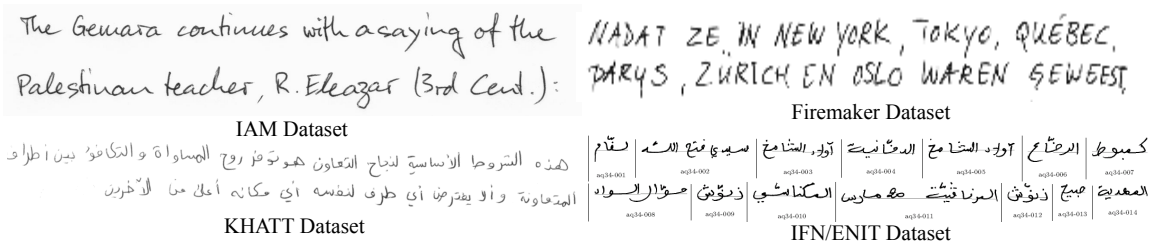

Figure 4.5: Written samples from the used datasets

Dataset	GMM	DMM	SDMM	SSDMM
IAM	83.21%	74.36%	94.87%	98.17%
Firemaker	64.36%	65.62%	91.46%	91.46%
KHATT	77.56%	20.71%	87.53%	87.53%
IFN/ENIT	79.82%	45.61%	70.45%	78.95%

Table 4.6: Performance of different models for writer identification on the considered datasets.

Chapter 5

Conclusion

Our work is related to the area of model-based clustering by proposing the shifted scaled Dirichlet mixture model in a purpose of extending the research work that has concerned modeling multivariate proportional data. Our choice of the shifted scaled Dirichlet distribution was motivated by its extra parameters that add flexibility to data modeling as compared to the Dirichlet distribution.

After that we discussed the maximum likelihood approach through implementing expectation maximization algorithm for our model parameters estimation. Note that, in real-world application we need a predefined the number of components that a dataset is generated from. For this purpose, we implement the minimum message length as a model selection criterion that help in determining the optimal number of clusters.

Thereafter, we used different datasets to evaluate our model and show its capability to cluster the chosen datasets with widely used performance measures. We first tested the model with synthetic data generated from the Dirichlet density and then compared the estimated model parameters with the real mixture model parameters. After that, we went further to execute the tests on real datasets with different applications related to life science, the medical field, business field, and retailers sales. We also considered a very popular application in software engineering about predicting defects-prone software modules which

has become very critical and expensive in case of large software projects. We ended up our experiments with consideration of the writer identification application which is a task of associating a handwriting sample with its writer identity. The manual writer identification is very time consuming that requires an exhaustive comparison for the details. Leaving the task for the computer is very useful to automatically confirm the authenticity of a document or to link together documents written by the same author.

We experience a number of challenges and limitations in all stages of this work.

- The limitation to handle a very high dimensional and sparse dataset.

  Due to the difficultly in computing the inverse of the high-dimensional Hessian matrix when estimating model parameters which requires more work in Bayesian methods to tackle that.

- The convergence to the global maximum is very difficult.

  because of the initialization step in EM, particularly while using the K-means algorithm. A better initialization methods could solve this problem.

- We experience an issue with imbalanced classes in the application of the software defect prediction.

  This issue makes it difficult for the algorithm to find the optimal parameters that define the defect group we are interested to know. Indeed, a small fraction of defects limits our detection ability. A solution could be the need of developing metrics or feature suitable for early defects detection for software modules.

Future works should tackle all the limitations we have faced. Moreover, finding efficient optimization techniques for estimating parameter vectors could be a promising future work. Also, online learning is an interesting direction.

Appendix A

We show here the possibility of obtaining the classic Dirichlet density model from the shifted scaled Dirichlet when we set $\tau = 1$ and the vector $\beta = C(1, \ldots, 1)$.

The shifted scaled Dirichlet distribution defined as follows,

$$\mathcal{SSD}(\mathbf{X}|\theta) = \frac{\Gamma(\alpha_+)}{\prod_{d=1}^{D}\Gamma(\alpha_d)}\frac{1}{\tau^{D-1}}\frac{\prod_{d=1}^{D}\beta_d^{-\frac{\alpha_d}{\tau}}x_d^{(\frac{\alpha_d}{\tau}-1)}}{(\sum_{d=1}^{D}\frac{x_d}{\beta_d}\frac{1}{\tau})^{\alpha_+}} \tag{45}$$

We show a simple decomposition of the Dirichlet from the shifted scaled Dirichlet.

$$\mathcal{D}(\mathbf{X}|\theta) = \frac{\Gamma(\alpha_+)\prod_{d=1}^{D}x_d^{(\alpha_d-1)}}{\prod_{d=1}^{D}\Gamma(\alpha_d)} \rightarrow \text{Dirichlet portion} \tag{46}$$

$$\mathcal{SS}(\mathbf{X}|\theta) = \frac{1}{\tau^{D-1}}\frac{\prod_{d=1}^{D}\beta_d^{-\frac{\alpha_d}{\tau}}x_d^{(\frac{1}{\tau})}}{(\sum_{d=1}^{D}\frac{x_d}{\beta_d}\frac{1}{\tau})^{\alpha_+}} \rightarrow \text{Shifted scale portion} \tag{47}$$

Which means that in the case of a Dirichlet density, the shifted-scaled portion is equal to 1.

$$\frac{1}{\tau^{D-1}}\frac{\prod_{d=1}^{D}\beta_d^{-\frac{\alpha_d}{\tau}}x_d^{(\frac{1}{\tau})}}{(\sum_{d=1}^{D}\frac{x_d}{\beta_d}\frac{1}{\tau})^{\alpha_+}} = 1 \tag{48}$$

Appendix B

In order to ensure all β constraints $0 \leq \beta_{jd} \leq 1; \sum_{d=1}^{D} \beta_{jd} = 1$ are satisfied, Lagrange Multiplier is introduced while estimating β_{jd}. Therefore, the augmented log likelihood is:

$$\phi(\Theta, Z, X, \Lambda) = \sum_{n=1}^{N} \sum_{j=1}^{K} z_{nj}(\log(\mathbf{X_n}|\boldsymbol{\alpha_j}, \boldsymbol{\beta_j}, \tau_j)) + \Lambda(1 - \sum_{d=1}^{D} \beta_{jd}) \tag{49}$$

Taking the derivative of the previous equation with respect to β_{jd}, Λ respectively, we obtain:

$$\frac{\partial \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)}{\partial \beta_{jd}} = \frac{1}{\Lambda}[\sum_{n=1}^{N} z_{nj} \frac{\alpha_+ X_{nd}}{\tau_j * \beta_{jd} \sum_{d=1}^{D} (\frac{X_{nd}}{\beta_{jd}})} - \frac{\alpha_{jd}}{\tau_j}] \tag{50}$$

$$\begin{aligned} &\frac{\partial \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)}{\partial \Lambda} = 1 - \sum_{d=1}^{D} \beta_{jd} \\ &\text{that gives us: } => \sum_{d=1}^{D} \beta_{jd} = 1 \end{aligned} \tag{51}$$

Then, by substituting Eq. 51 in Eq. 50 and solve them in order to end up with Λ equation as following:

$$\Lambda = \sum_{n=1}^{N} \hat{z}_{nj} \sum_{d=1}^{D} \frac{\alpha_+ X_{nd}}{\tau_j \beta_{jd} \sum_{d=1}^{D} (\frac{X_{nd}}{\beta_{jd}})} - \sum_{d=1}^{D} \frac{\alpha_{jd}}{\tau_j}. \tag{52}$$

Thereafter, we get the final β_{jd} equation after we plug Eq. 52 in Eq. 50 as following:

$$\beta_{jd} = \frac{\sum_{n=1}^{N} z_{nj} \left(\frac{\alpha_{+} X_{nd}}{\tau_j \beta_{jd} \sum_{d=1}^{D} (\frac{X_{nd}}{\beta_{jd}})} - \frac{\alpha_{jd}}{\tau_j} \right)}{\sum_{n=1}^{N} \hat{Z_{nj}} \left(\sum_{d=1}^{D} \frac{\alpha_{+} X_{nd}}{\tau_j \beta_{jd} \sum_{d=1}^{D} (\frac{X_{nd}}{\beta_{jd}})} - \sum_{d=1}^{D} \frac{\alpha_{jd}}{\tau_j} \right)} \tag{53}$$

References

Abreu, N. G. C. F. M., et al. (2011). *Analise do perfil do cliente recheio e desenvolvimento de um sistema promocional* (Unpublished doctoral dissertation).

Baudry, J.-P., Cardoso, M., Celeux, G., Amorim, M. J., & Ferreira, A. S. (2012). Enhancing the selection of a model-based clustering with external qualitative variables. *arXiv preprint arXiv:1211.0437*.

Baxter, R. A. (1996). Minimum message length inference: Theory and applications. In *Monash university, australia.*

Baxter, R. A., & Oliver, J. J. (2000). Finding overlapping components with mml. *Statistics and Computing*, *10*(1), 5–16.

Bdiri, T., & Bouguila, N. (2012). Positive vectors clustering using inverted dirichlet finite mixture models. *Expert Systems with Applications*, *39*(2), 1869–1882.

Bouguila, N., & Ziou, D. (2005a). Mml-based approach for finite dirichlet mixture estimation and selection. In *International workshop on machine learning and data mining in pattern recognition* (pp. 42–51).

Bouguila, N., & Ziou, D. (2005b). On fitting finite dirichlet mixture using ecm and mml. In *International conference on pattern recognition and image analysis* (pp. 172–182).

Bouguila, N., & Ziou, D. (2006a). A hybrid sem algorithm for high-dimensional unsupervised learning using a finite generalized dirichlet mixture. *IEEE Transactions on Image Processing*, *15*(9), 2657–2668.

Bouguila, N., & Ziou, D. (2006b). Unsupervised selection of a finite dirichlet mixture

model: an mml-based approach. *IEEE Transactions on Knowledge and Data Engineering*, *18*(8), 993–1009.

Bouguila, N., & Ziou, D. (2007). High-dimensional unsupervised selection and estimation of a finite generalized dirichlet mixture model based on minimum message length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(10).

Bouguila, N., Ziou, D., & Vaillancourt, J. (2004). Unsupervised learning of a finite mixture model based on the dirichlet distribution and its application. *IEEE Transactions on Image Processing*, *13*(11), 1533–1543.

Brock, G., Pihur, V., Datta, S., Datta, S., et al. (2011). clvalid, an r package for cluster validation. *Journal of Statistical Software (Brock et al., March 2008)*.

Bulacu, M., Schomaker, L., & Vuurpijl, L. (2003). Writer identification using edge-based directional features. In *null* (p. 937).

Catal, C. (2011). Software fault prediction: A literature review and current trends. *Expert systems with applications*, *38*(4), 4626–4636.

Christlein, V., Gropp, M., Fiel, S., & Maier, A. (2017). Unsupervised feature learning for writer identification and writer retrieval. In *Document analysis and recognition (icdar), 2017 14th iapr international conference on* (Vol. 1, pp. 991–997).

Conway, J. H., & Sloane, N. J. A. (2013). *Sphere packings, lattices and groups* (Vol. 290). Springer Science & Business Media.

Costa Filho, I. G. (2008). *Mixture models for the analysis of gene expression: integration of multiple experiments and cluster validation* (Unpublished doctoral dissertation). Citeseer.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1–38.

Dowe, D., & Farr, G. (1997). An introduction to mml inference. *Technical Report*.

Elkan, C. (2003). Using the triangle inequality to accelerate k-means. In *Proceedings of the 20th international conference on machine learning (icml-03)* (pp. 147–153).

Erman, J., Arlitt, M., & Mahanti, A. (2006). Traffic classification using clustering algorithms. In *Proceedings of the 2006 sigcomm workshop on mining network data* (pp. 281–286).

Figueiredo, M. A. T., & Jain, A. K. (2002). Unsupervised learning of. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(3), 381–396.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of human genetics*, *7*(2), 179–188.

Gentle, J. (1998). The em algorithm and extensions. *Biometrics*, *54*(1), 395.

Giordan, M., & Wehrens, R. (2015). A comparison of computational approaches for maximum likelihood estimation of the dirichlet parameters on high-dimensional data. *SORT*, *39*, 109–126.

Graybill, F. A. (1983). Matrices with applications in statistics.

Gupta, R. D., & Richards, D. S. P. (2001). The history of the dirichlet and liouville distributions. *International Statistical Review*, *69*(3), 433–446.

Hankin, R. K., et al. (2010). A generalization of the dirichlet distribution. *Journal of Statistical Software*, *33*(11), 1–18.

He, S., Wiering, M., & Schomaker, L. (2015). Junction detection in handwritten documents and its application to writer identification. *Pattern Recognition*, *48*(12), 4036–4048.

Hu, Z. (2015). *Initializing the em algorithm for data clustering and sub-population detection* (Unpublished doctoral dissertation). The Ohio State University.

Huang, J. (2005). Maximum likelihood estimation of dirichlet distribution parameters. *CMU Technique Report*.

Jefferys, W. H., & Berger, J. O. (1992). Ockham's razor and bayesian analysis. *American Scientist*, *80*(1), 64–72.

Jiang, Y., Cukic, B., & Menzies, T. (2007). Fault prediction using early lifecycle data. In *Software reliability, 2007. issre'07. the 18th ieee international symposium on* (pp. 237–246).

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, *32*(3), 241–254.

Kaufman, L., & Rousseeuw, P. (1987). *Clustering by means of medoids*. North-Holland.

Khozeimeh, F., Alizadehsani, R., Roshanzamir, M., Khosravi, A., Layegh, P., & Nahavandi, S. (2017). An expert system for selecting wart treatment method. *Computers in biology and medicine*, *81*, 167–175.

Khozeimeh, F., Jabbari Azad, F., Mahboubi Oskouei, Y., Jafari, M., Tehranian, S., Alizadehsani, R., & Layegh, P. (2017). Intralesional immunotherapy compared to cryotherapy in the treatment of warts. *International journal of dermatology*, *56*(4), 474–478.

Kirkos, E., Spathis, C., & Manolopoulos, Y. (2007). Data mining techniques for the detection of fraudulent financial statements. *Expert systems with applications*, *32*(4), 995–1003.

Li, M., & Zhang, L. (2008). Multinomial mixture model with feature selection for text clustering. *Knowledge-Based Systems*, *21*(7), 704–708.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Mahmoud, S. A., Ahmad, I., Alshayeb, M., Al-Khatib, W. G., Parvez, M. T., Fink, G. A., ... El Abed, H. (2012). Khatt: Arabic offline handwritten text database. In *2012 international conference on frontiers in handwriting recognition (icfhr 2012)* (pp. 449–454).

Marti, U.-V., & Bunke, H. (2002). The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, *5*(1), 39–46.

McCabe, T. J. (1976). A complexity measure. *IEEE Transactions on software Engineering*(4), 308–320.

McLachlan, G., & Krishnan, T. (2007). *The em algorithm and extensions* (Vol. 382). John Wiley & Sons.

McLachlan, G., & Peel, D. (2004). *Finite mixture models*. John Wiley & Sons.

McLachlan, G. J., & Basford, K. E. (1988). *Mixture models: Inference and applications to clustering* (Vol. 84). Marcel Dekker.

McLachlan, G. J., & Peel, D. (2000). *Finite mixture models, volume 299 of probability and statistics–applied probability and statistics section.* Wiley, New York.

Medasani, S., & Krishnapuram, R. (1999). A rison of gaussian and pearson mixture modeling for pattern recognition and computer vision applications. *Pattern recognition letters*, *20*(3), 305–313.

Meila, M., & Shi, J. (2001). Learning segmentation by random walks. In *Advances in neural information processing systems* (pp. 873–879).

Minka, T. (2000). *Estimating a dirichlet distribution.* Technical report, MIT.

Monti, G., Mateu i Figueras, G., Pawlowsky-Glahn, V., Egozcue, J. J., et al. (2011). The shifted-scaled dirichlet distribution in the simplex.

Monti, G. S., Mateu-Figueras, G., & Pawlowsky-Glahn, V. (2011). *Notes on the scaled dirichlet distribution*. John Wiley & Sons, Chichester.

Najadat, H., & Alsmadi, I. (2012). Enhance rule based detection for software fault prone modules. *International Journal of Software Engineering and Its Applications*, *6*(1), 75–86.

Nath, S. V. (2006). Crime pattern detection using data mining. In *Web intelligence and intelligent agent technology workshops, 2006. wi-iat 2006 workshops. 2006 ieee/wic/acm international conference on* (pp. 41–44).

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an

algorithm. In *Advances in neural information processing systems* (pp. 849–856).

Ng, K. W., Tian, G.-L., & Tang, M.-L. (2011). *Dirichlet and related distributions: Theory, methods and applications* (Vol. 888). John Wiley & Sons.

Oboh, B. S., & Bouguila, N. (2017). Unsupervised learning of finite mixtures using scaled dirichlet distribution and its application to software modules categorization. In *Industrial technology (icit), 2017 ieee international conference on* (pp. 1085–1090).

Oboh, E. S. (2016). *Cluster analysis of multivariate data using scaled dirichlet finite mixture model* (Unpublished doctoral dissertation). Concordia University.

Ongaro, A., & Migliorati, S. (2013). A generalization of the dirichlet distribution. *Journal of Multivariate Analysis*, *114*, 412–426.

Ongaro, A., Migliorati, S., Monti, G. S., et al. (2008). A new distribution on the simplex containing the dirichlet family.

Pawlowsky-Glahn, V., & Buccianti, A. (2011). *Compositional data analysis: Theory and applications*. John Wiley & Sons.

Pechwitz, M., Maddouri, S. S., Märgner, V., Ellouze, N., Amiri, H., et al. (2002). Ifn/enit-database of handwritten arabic words. In *Proc. of cifed* (Vol. 2, pp. 127–136).

Powell, P. D. (2011). Calculating determinants of block matrices. *arXiv preprint arXiv:1112.4379*.

Ronning, G. (1989). Maximum likelihood estimation of dirichlet distributions. *Journal of statistical computation and simulation*, *32*(4), 215–221.

Shao, J. (1993). Linear model selection by cross-validation. *Journal of the American statistical Association*, *88*(422), 486–494.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(8), 888–905.

Shihab, E. (2012). *An exploration of challenges limiting pragmatic software defect prediction* (Unpublished doctoral dissertation). Queen's University (Canada).

Shihab, E. (2014). Practical software quality prediction. In *Software maintenance and evolution (icsme), 2014 ieee international conference on* (pp. 639–644).

Sneath, P. H. (1957). The application of computers to taxonomy. *Microbiology*, *17*(1), 201–226.

Sun, Z.-L., Choi, T.-M., Au, K.-F., & Yu, Y. (2008). Sales forecasting using extreme learning machine with applications in fashion retailing. *Decision Support Systems*, *46*(1), 411–419.

Theodoridis, S., Koutroumbas, K., et al. (2008). Pattern recognition. *IEEE Transactions on Neural Networks*, *19*(2), 376.

Wallace, C. S., & Dowe, D. L. (2000). Mml clustering of multi-state, poisson, von mises circular and gaussian distributions. *Statistics and Computing*, *10*(1), 73–83.

Wallace, C. S., & Freeman, P. R. (1987). Estimation and inference by compact coding. *Journal of the Royal Statistical Society. Series B (Methodological)*, 240–265.

Wu, X., Tang, Y., & Bu, W. (2014). Offline text-independent writer identification based on scale invariant feature transform. *IEEE Transactions on Information Forensics and Security*, *9*(3), 526–536.