Positive Data Clustering Using Finite Inverted Dirichlet Mixture Models

Taoufik BDIRI

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 in Quality Systems Engineering at

Concordia University

Montréal, Québec, Canada

August 2010

Your file Votre référence
ISBN: 978-0-494-71069-2
Our file Notre référence
ISBN: 978-0-494-71069-2

Abstract

Positive Data Clustering Using Finite Inverted Dirichlet Mixture Models

Taoufik BDIRI

In this thesis we present an unsupervised algorithm for learning finite mixture models from multivariate positive data. Indeed, this kind of data appears naturally in many applications, yet it has not been adequately addressed in the past. This mixture model is based on the inverted Dirichlet distribution, which offers a good representation and modeling of positive non gaussian data. The proposed approach for estimating the parameters of an inverted Dirichlet mixture is based on the maximum likelihood (ML) using Newton Raphson method. We also develop an approach, based on the Minimum Message Length (MML) criterion, to select the optimal number of clusters to represent the data using such a mixture. Experimental results are presented using artificial histograms and real data sets. The challenging problem of software modules classification is investigated within the proposed statistical framework, also.

Acknowledgements

First, I would like to extend my most sincere gratitude to Dr. Nizar Bouguila, my supervisor, who guided, encouraged, and supported me on my entire journey of studying in Canada in this program. His patience, his high standards, and his willingness to help me provided me with the confidence and direction which I most needed and which I appreciate. I have been very fortunate to have worked with one of the best and most talented supervisers in Concordia. Thank you!

I also express my sincere thanks to the Tunisian government and the University Mission of Tunisia in Montreal, for their endless support that enabled me to study in the most comfortable conditions.

I also am in great appreciation to Dr. A. Ben Hamza and Dr. J. Bentahar, for their assistance and encouragements.

I owe my deepest thanks to my lab mates, Sefidpour, Mohamed, Shojaee, Mosleh, Ola, Fan, Tarek, Khaled, and Dinner, for their support and the great time I spent with them during this master.

Finally, I would like to thank my father Mohamed and mother Naima for their constantly unconditional support, encouragement and endless love. I also extend my warmest thanks to my brother Hatem, to my two sisters Rym and Yosra, and to my beloved Feriel for her love, patience and support.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Introduction and Related works

Data clustering is an important technique for data analysis, and has been largely studied. Indeed, it has been shown to be a crucial step in many practical domains such as information retrieval and data mining [1]. Large corporations, hospitals, organizations and companies, all require accurate analysis of their data. Statistical-based approaches and in particular finite mixture models have long been the workhorse for data clustering, and have seen a real boost in popularity over the last decade due to the tremendous increase in available computing power [2, 3]. Its attractiveness lies in its simplicity, flexibility and in its strong statistical foundations which offer a formal model-based framework for clustering. Moreover, finite mixtures are a natural choice in many practical situations where the data of interest may be considered to consist of categories mixed in varying proportions [4]. With the technological and scientific advancement, the nature of generated data has became more rich and complex which adds more challenges when adopting finite mixture models [5]. Much of the related work, however, does not attempt to provide specific models according to the nature of the generated data. Indeed, the majority of research works related to finite mixture models have been based on the Gaussian assumption. Recently, a series of papers have shown that this assumption is generally inappropriate and that other mixture models can outperform the Gaussian mixture by discovering more efficiently useful patterns and correlations among data features. For instance, recent studies have shown that finite Dirichlet mixtures are more appropriate for proportional data (or normalized histograms) clustering and outperform significantly their Gaussian counterpart in several image processing [6], pattern

recognition [7], and data mining applications [8]. In this thesis, however, we focus on the modeling and clustering of positive data which despite the fact they occur naturally in many real-life applications, have not received much attention. We present then a clustering algorithm, based on finite mixtures of inverted Dirichlet distributions, and demonstrate that it is especially suitable for clustering this kind of data.

To the best of our knowledge finite inverted Dirichlet mixture models have never been considered in the past. In fact, compared to the Dirichlet, the inverted Dirichlet has received less attention from both a practical and theoretical point of view. Thus, the main goal of this thesis is to introduce these models and to show their usefulness in practical settings. The main motivation is the flexibility of the inverted Dirichlet distribution which, in contrast to the Gaussian, permits multiple symmetric and asymmetric mode, it may be skewed to the right, skewed to the left or symmetric as we will show in the next chapter. Given unlabeled samples, an important problem when considering finite mixtures is the learning of the model. By learning, we mean both the selection of the most optimal number of mixture components to represent the data and the estimation of the related parameters. One of the most mature approaches for parameters estimation is the maximum likelihood (ML) [9] performed generally through the expectation-maximization (EM) algorithm [10]. Another considerable part of the unsupervised learning, is to determine the proper number of clusters that represent the data which can be viewed actually as a *model search* step. Typical criteria to select the number of clusters test models by assigning to them values determined by the likelihood associated with the model, the number of free parameters in the model, the data and prior information about the model's parameters. Popular criteria include the Akaike information criterion (AIC) [11], Bayes information criterion (BIC) [12] and minimum description length (MDL) [13]. In a previous work, the authors in [8] have shown that all these criteria can be viewed as an approximation to the minimum message length criterion which generally gives the best results. Thus, we develop in this work an expression to represent the message length of a finite inverted Dirichlet mixture which gives us a statistically founded approach that outputs both the optimal number of clusters, by minimizing the message length of the data, and their parameters.

1.2 Contributions

The contributions of this thesis are as follows:

☞ **A Novel finite mixture model for efficient positive non Gaussian data clustering:** Our approach suggest a new distribution which, to the extend of our knowledge, hasn't been used before in clustering. The distribution seems to give good clustering results for positive non Gaussian data. We developed all the equations related to its parameters estimation and the algorithm to select the optimal number of clusters to represent and fit a given positive data set. We have proven that the finite inverted Dirichlet mixture can be a good candidate to cluster positive non Gaussian data.

☞ **Comparison of the inverted Dirichlet mixture model performance with finite Gaussian mixture model:** We compare the performance of our developed model with a finite Gaussian mixture model in terms of clustering, and the selection of optimal number of clusters. The comparison is based on challenging applications namely real data clustering and software modules analysis. We show that the inverted Dirichlet mixture outperforms the Gaussian one, when the data is positive and non gaussian.

1.3 Thesis Overview

The organization of this thesis is as follows:

- ❑ Chapter 1 introduces the dilemma of clustering data sets, the major role that finite mixture models play for data clustering. It also introduces methods for selecting the appropriate number of clusters for a data set.
- ❑ Chapter 2 proposes a new finite multivariate inverted Dirichlet mixture model approach, which is capable of clustering data sets. Moreover, we develop the message length expression for the inverted Dirichlet distribution to determine the number of clusters that fits the most a given data set.
- ❑ Chapter 3 is devoted to the experimental results of the application of our approach on synthetic and real data. The obtained results are compared with those of finite Gaussian mixture.
- ❑ Chapter 4 gives the summary, conclusion and potential avenues for future research.
- ❑ Chapter 5 (Appendices): presents the derivation equations of the finite mixture of multivariate Gaussian distributions, and the expression of the minimum message length in this case.

CHAPTER 2

Positive Data Clustering Using Finite Inverted Dirichlet Mixture Models

2.1 Introduction

In the previous chapter, we presented the dilemma of data clustering, and some previous work related to the use of finite mixture models in different applications that include data clustering and the selection of the optimal number of clusters for a given data set. In this chapter, we propose a new mixture model basing on the inverted Dirichlet distribution and we derivate its equations. We establish the maximum likelihood estimation, basing on the expectation-maximization (EM) algorithm, using Newton Raphson method. After the presentation of our algorithm for parameters estimation, we define the minimum message length (MML) for an inverted Dirichlet mixture to determine the number of clusters that fits the most a given data set.

2.2 Finite Inverted Dirichlet Mixture Model

If a D-dimensional vector positive vector $\vec{X} = (X_1, X_2, ..., X_D)$ follows an inverted Dirichlet distribution, the joint density function is given by [14]

$$p(\vec{X}|\vec{\alpha}) = \frac{\Gamma(|\vec{\alpha}|)}{\prod_{d=1}^{D+1}\Gamma(\alpha_d)} \prod_{d=1}^{D} X_d^{\alpha_d - 1}(1 + \sum_{d=1}^{D} X_d)^{-|\vec{\alpha}|} \quad (1)$$

where $X_d > 0, d = 1, 2, \ldots, D$, $\vec{\alpha} = (\alpha_1, \ldots, \alpha_{D+1})$ is the vector of parameters and $|\vec{\alpha}| = \sum_{d=1}^{D+1} \alpha_d$, $\alpha_d > 0, d = 1, 2, \ldots, D+1$. The inverted Dirichlet distribution was introduced for the first time in [14] where the authors derived it from the Dirichlet distribution. Another derivation based on the Gamma distribution was proposed also in [15]. It is noteworthy that like the Dirichlet, the inverted Dirichlet permits multiple symmetric and asymmetric modes and it may be skewed to the right, skewed to the left or symmetric (see Figure 2.1). Several interesting properties of the inverted Dirichlet can be found in [16]. The mean and the variance of the inverted Dirichlet distribution are given by [14]

$$E(X_d) = \frac{\alpha_d}{\alpha_{D+1} - 1} \tag{2}$$

$$Var(X_d) = \frac{\alpha_d(\alpha_d + \alpha_{D+1} - 1)}{(\alpha_{D+1} - 1)^2(\alpha_{D+1} - 2)} \tag{3}$$

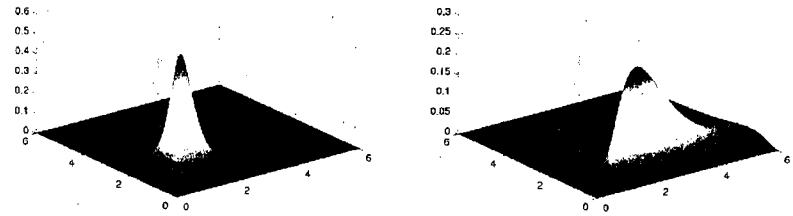
Let $\mathcal{X} = \{\vec{X}_1, \vec{X}_2, \ldots, \vec{X}_N\}$ be a data set of N D-dimensional positive vectors with a common, but unknown, probability density function $p(\vec{X}|\Theta)$. Generally $\mathcal{X}$ is composed of different, say M, clusters, thus $p(\vec{X}|\Theta)$ may be approximated with sufficient accuracy by a finite M-components mixture model:

$$p(\vec{X}|\Theta) = \sum_{j=1}^{M} p(\vec{X}|\vec{\alpha}_j)p_j \tag{4}$$

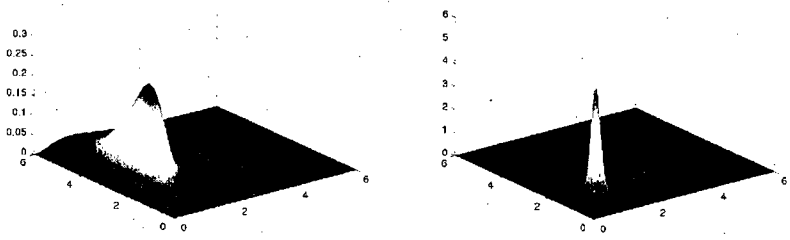
where the p_j are the mixing proportions which are positive and sum to one, and $p(\vec{X}|\vec{\alpha}_j)$ is the inverted Dirichlet distribution. The symbol Θ refers to the entire set of parameters to be estimated $\Theta = \{\vec{\alpha}_1, \vec{\alpha}_2, \ldots, \vec{\alpha}_M, p_1, p_2, \ldots, p_M\}$ with $\vec{\alpha_j} = (\alpha_{j1}, \alpha_{j2}, \cdots, \alpha_{jD+1})$ and represents the parameter vector for the j^{th} population. Figure 2.2 displays examples of finite inverted Dirichlet mixtures with different parameters. In the following sections, we shall tackle the problems of estimating Θ and also selecting the number of clusters M.

2.3 Finite Inverted Dirichlet Mixture Model Estimation

In this section, we first develop the maximum likelihood estimates of the parameters of a finite inverted Dirichlet mixture. Then, we give the complete estimation algorithm.



(a) $\alpha_1 = 3.5, \alpha_2 = 3.5, \alpha_3 = 3.5$ (b) $\alpha_1 = 10, \alpha_2 = 3.5, \alpha_3 = 3.5$



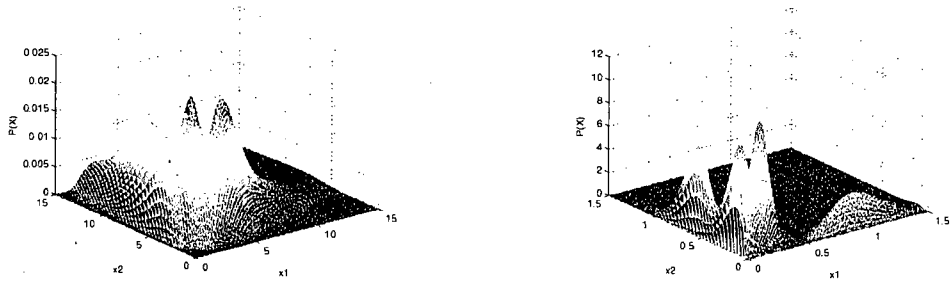
(c) $\alpha_1 = 3.5, \alpha_2 = 10, \alpha_3 = 3.5$ (d) $\alpha_1 = 3.5, \alpha_2 = 3.5, \alpha_3 = 10$

Figure 2.1: Bivariate inverted Dirichlet distributions with different parameters.

2.3.1 Maximum Likelihood Estimation

The estimation of finite mixture models has been the subject of many studies especially in the case of Gaussian data (see, for instance, [17]). The maximum likelihood approach remains, however, the most popular technique. The maximum likelihood estimate, associated with a sample of observations, is a choice of parameters which maximizes the probability density function of the sample, $\max_{\Theta} p(\mathcal{X}|\Theta)$, taking into account the constraints about the mixing parameters mentioned in the previous section. For convenience, the log-likelihood is generally maximized instead of the likelihood:

$$\Phi(\mathcal{X}|\Theta) = \log p(\mathcal{X}|\Theta) = \log \prod_{n=1}^{N} p(\vec{X}_n|\Theta) = \sum_{n=1}^{N} \log \left(\sum_{j=1}^{M} p(\vec{X}_n|\vec{\alpha_j})p_j\right) \tag{5}$$



(a) (b)

Figure 2.2: Two-dimensional inverted Dirichlet mixtures with different parameters. (a) A three-components mixture model with parameters $p_1 = 0.33$, $\alpha_{11} = 10$, $\alpha_{12} = 40$, $\alpha_{13} = 4$, $p_2 = 0.33$, $\alpha_{21} = 20$, $\alpha_{22} = 30$, $\alpha_{23} = 5$, $p_3 = 0.33$, $\alpha_{31} = 30$, $\alpha_{32} = 20$, $\alpha_{33} = 5$. (b) A four-components mixture model with parameters $p_1 = 0.2$, $\alpha_{11} = 18$, $\alpha_{12} = 14$, $\alpha_{13} = 70$, $p_2 = 0.4$, $\alpha_{21} = 2$, $\alpha_{22} = 4$, $\alpha_{23} = 17$, $p_3 = 0.2$, $\alpha_{31} = 40$, $\alpha_{32} = 5$, $\alpha_{33} = 40$, $p_4 = 0.2$, $\alpha_{41} = 5$, $\alpha_{42} = 40$, $\alpha_{43} = 60$.

The maximization of the previous equation is generally performed within the EM framework where each $\vec{X}_n$ is supposed to have arisen from one the M clusters. Thus, let $\mathcal{Z} = \{\vec{Z}_1, \ldots, \vec{Z}_N\}$ denote the missing group-indicator vectors where the jth element of $\vec{Z}_n$, Z_{nj}, is equal to one if $\vec{X}_n$ belongs to cluster j and zero, otherwise. The complete data in this case are $(\mathcal{X}, \mathcal{Z})$ and the associated complete-data log-likelihood is given by

$$\Phi_c(\mathcal{X}, \mathcal{Z}|\Theta) = \sum_{j=1}^{M}\sum_{n=1}^{N} Z_{nj}\Big(\log p_j + \log p(\vec{X}_n|\vec{\alpha_j})\Big) \tag{6}$$

The EM algorithm proceeds iteratively in two steps, The expectation (E) step and the maximization (M) step. In the E-step, we compute the conditional expectation of $\Phi_c(\mathcal{X}, \mathcal{Z}|\Theta)$ which is reduced to the computation of the posterior probabilities (i.e. the probability that a vector $\vec{X}_n$ is assigned to a cluster j):

$$p(j|\vec{X}_n, \vec{\alpha}_j) = \frac{p_j p(\vec{X}_n|\vec{\alpha}_j)}{\sum_{j=1}^{M} p_j p(\vec{X}_n|\vec{\alpha}_j)} \tag{7}$$

Then, the conditional expectation of the complete-data log likelihood given by

$$Q(\mathcal{X}, \Theta) = \sum_{j=1}^{M}\sum_{n=1}^{N} p(j|\vec{X}_n, \vec{\alpha}_j)\Big(\log p_j + \log p(\vec{X}_n|\vec{\alpha}_j)\Big) \tag{8}$$

is maximized in the M-step. To resolve this optimization problem, we must determine the solution to $\frac{\partial}{\partial\Theta}Q(\mathcal{X},\Theta)=0$. Calculating the derivative with to respect $\alpha_{jd}, d=1,\ldots,D$, we obtain

$$\begin{aligned}\frac{\partial Q(\mathcal{X},\Theta)}{\partial\alpha_{jd}} &= \sum_{n=1}^{N} p(j|\vec{X}_n,\vec{\alpha}_j)\frac{\partial}{\partial\alpha_j}\log(p(\vec{X}_n|\vec{\alpha}_j)) \\ &= \sum_{n=1}^{N} p(j|\vec{X}_n,\vec{\alpha}_j)\Big(\Psi(|\vec{\alpha}_j|)-\Psi(\alpha_{jd})+\log\big(\frac{X_{nd}}{1+\sum_{d=1}^{D}X_{nd}}\big)\Big) \end{aligned} \tag{9}$$

where $\Psi(.)$ is the digamma function. The derivative with to respect α_{jD+1} is given by

$$\frac{\partial Q(\mathcal{X},\Theta)}{\partial\alpha_{jD+1}} = \sum_{n=1}^{N} p(j|\vec{X}_n,\vec{\alpha}_j)\Big(\Psi(|\vec{\alpha}_j|)-\Psi(\alpha_{jD+1})+\log\big(\frac{1}{1+\sum_{d=1}^{D}X_{nd}}\big)\Big) \tag{10}$$

According to the previous two equations, it is clear that a closed-form solution to estimate $\vec{\alpha}_j$ does not exist. Thus, we will use an iterative approach namely the Newton-Raphson method expressed as

$$\hat{\vec{\alpha}}_j^{new} = \hat{\vec{\alpha}}_j^{old} - H_j^{-1}G_j \qquad j=1,2,\ldots,M \tag{11}$$

Where H_j is the Hessian matrix associated with $Q(\mathcal{X},\Theta)$ and G_j is the first derivatives vector, $G_j = (\frac{\partial Q(\mathcal{X},\Theta)}{\partial\alpha_{j1}},\ldots,\frac{\partial Q(\mathcal{X},\Theta)}{\partial\alpha_{jD+1}})^T$. To calculate the Hessian of $Q(\mathcal{X},\Theta)$ we have to compute the second and mixed derivatives:

$$\frac{\partial^2 Q(\mathcal{X},\Theta)}{\partial^2\alpha_{jd}} = (\Psi'(|\vec{\alpha}_j|)-\Psi'(\alpha_{jd}))\sum_{n=1}^{N} p(j|\vec{X}_n,\vec{\alpha}_j) \qquad d=1,\ldots,D+1 \tag{12}$$

$$\frac{\partial^2 Q(\mathcal{X},\Theta)}{\partial\alpha_{jd_1}\partial\alpha_{jd_2}} = \Psi'(|\vec{\alpha}_j|)\sum_{n=1}^{N} p(j|\vec{X}_n,\vec{\alpha}_j) \quad d_1\neq d_2 \quad d_1,d_2=1,\ldots,D+1 \tag{13}$$

where $\Psi'(.)$ is the trigamma function. Thus,

$$H_j = \textstyle\sum_{n=1}^{N} p(j|\vec{X}_n,\vec{\alpha}_j)\begin{pmatrix} \Psi'(|\vec{\alpha}_j|)-\Psi'(\alpha_{j1}) & \Psi'(|\vec{\alpha}_j|) & \cdots & \Psi'(|\vec{\alpha}_j|) \\ \Psi'(|\vec{\alpha}_j|) & \Psi'(|\vec{\alpha}_j|)-\Psi'(\alpha_{j2}) & \cdots & \Psi'(|\vec{\alpha}_j|) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi'(|\vec{\alpha}_j|) & \cdots & & \Psi'(|\vec{\alpha}_j|)-\Psi'(\alpha_{jD+1}) \end{pmatrix} \tag{14}$$

Thus, H_j can be written as follows:

$$H_j = D_j + \gamma_j A_j^T A_j \tag{15}$$

where $D_j = diag[-\sum_{n=1}^{N} p(j|\vec{X}_n, \vec{\alpha}_j)\Psi'(\alpha_{j1}), \ldots, -\sum_{n=1}^{N} p(j|\vec{X}_n, \vec{\alpha}_j)\Psi'(\alpha_{jD+1})]$ is a diagonal matrix, $\gamma_j = \sum_{n=1}^{N} p(j|\vec{X}_n, \vec{\alpha}_j)\Psi'(|\vec{\alpha}_j|)$, and $A_j^T = (a_1, a_2, \ldots, a_{D+1})$ with $a_i = 1 \ \ \forall i$. Then, by the theorem of matrix inverse given by Graybill [18, Theorem 8.3.3], we have:

$$H_j^{-1} = D_j^{-1} + \gamma_j^* A_j^{*T} A_j^* \tag{16}$$

where D_j^{-1} can be easily computed and

$$A_j^* = \frac{-1}{\sum_{n=1}^{N} p(j|\vec{X}_n, \vec{\alpha}_j)} \left(\frac{1}{\Psi'(\alpha_{j1})}, \ldots, \frac{1}{\Psi'(\alpha_{jD+1})}\right) \tag{17}$$

$$\gamma_j^* = \left[(\Psi'(|\vec{\alpha}_n|) \sum_{d=1}^{D+1} \frac{1}{\Psi'(\alpha_{jd})}) - 1\right] \Psi'(|\vec{\alpha}_j|) \sum_{n=1}^{N} p(j|\vec{X}_n, \vec{\alpha}_j) \tag{18}$$

Once we have H_j^{-1} and G_j we can apply the Newton Raphson step in Eq. 11 to update the parameters estimation of the inverted Dirichlet mixture. Concerning the parameters p_j, it is straightforward to show that a closed-form solution does exist and is given by :

$$p_j = \frac{\sum_{n=1}^{N} p(j|\vec{X}_n, \vec{\alpha}_j)}{N} \tag{19}$$

2.3.2 Initialization and Complete Estimation Algorithm

The EM algorithm needs initial estimates as starting values which have crucial importance for successful mixture estimation. Our initialization scheme is based on both the well-known K-Means algorithm and the method of moments. The method of moments relies on low order statistics of the inverted Dirichlet namely the mean and the variance given by Eqs. 2 and 3, respectively, from which can straightforwardly find the following initial estimates for each mixture component j:

$$\alpha_{jD+1} = \frac{E(X_d)^2 + E(X_d)}{Var(X_d)} + 2 \tag{20}$$

$$\alpha_{jd} = E(X_d)(\alpha_{jD+1} - 1) \quad d = 1, \ldots, D+1 \tag{21}$$

The initialization algorithm can be described as follows:

Initialization Algorithm

1. Apply the k-means on the N D-dimensional vectors to obtain initial M clusters.

2. Calculate $p_j = \frac{Number\ of\ elements\ in\ calss\ j}{N}$

3. Apply the moments method for each component j to obtain a vector of parameter $\vec{\alpha}_j$ using Eqs. 20 and 21.

Then, the algorithm of the inverted Dirichlet mixture estimation can be summarized as follows

Estimation Algorithm

1. INPUT: D-dimensional data $\vec{X}_n$, $n = 1, \ldots, N$ and a specified number of clusters M.

2. Initialization algorithm

3. E-Step: Compute the posterior probabilities $p(j|\vec{X}_n, \vec{\alpha}_j)$ using Eq. 5.

4. M-Step:

   - Update the $\vec{\alpha_j}$ using Eq. 11, $j = 1, \ldots, M$.
   - Update p_j using Eq. 19, $j = 1, \ldots, M$.

5. If $p_j < \epsilon$, discard component j go to 3.

6. If the convergence test ($\Delta \exp(Q(\mathcal{X}, \Theta)) < \epsilon$) is passed terminate, else go to 3.

2.4 MML for Inverted Dirichlet Mixture

The goal of this section is to develop a criterion that automatically determines the number of components of an inverted Dirichlet mixture given a data set of positive vectors. Previous studies (see, for instance, [8]) have shown that the MML criterion allows principled model selection and that it generalizes many other criteria. The MML method codes at the same time the model and its associated parameters, and the data given the model. The message length for a mixture of distributions is given by [19]

$$MessLen \simeq -\log(h(\Theta)) - \log(p(\mathcal{X}|\Theta)) + \frac{1}{2}\log(|F(\Theta)|) + \frac{N_p}{2}(1 - \log(12)) \tag{22}$$

where $h(\Theta)$ is the prior probability, $p(\mathcal{X}|\Theta)$ is the likelihood, $F(\Theta)$ is the expected Fisher information matrix, $|F(\Theta)|$ is its determinant, and N_p is the number of free parameters to be estimated which is equal to $M(D+2)-1$. The selection of the number of clusters is carried out by finding the minimum with regards to Θ of the message length $Messlen$. In the following, we first develop the Fisher information for a mixture of inverted Dirichlet distribution and then we propose a prior distribution.

The expected Fisher information matrix is generally approximated by complete-data Fisher information matrix in the case of finite mixture models [20]. The complete-data Fisher information matrix has block-diagonal structure and then:

$$|F(\Theta)| \simeq |F(p_1,\ldots,p_M)| \prod_{j=1}^{M} |F(\vec{\alpha}_j)| \tag{23}$$

where $|F(p_1,\ldots,p_M)|$ is the Fisher information with regards to the mixing parameters vector and we can show that [8]:

$$|F(p_1,\ldots,p_M)| = \frac{N^{M-1}}{\prod_{j=1}^{M} p_j} \tag{24}$$

and $|F(\vec{\alpha}_j)|$ the Fisher information with regards to $\vec{\alpha}_j$ of a single inverted Dirichlet distribution. For $F(\vec{\alpha}_j)$, let us consider the jth cluster of the mixture $\mathcal{X}_j = (\vec{X}_l,\ldots,\vec{X}_{l+n_j-1})$, where $l \leq N$ and n_j is the number of elements in cluster j, with parameter $\vec{\alpha}_j$. We can write the negative of the log-likelihood function as follows

$$-\log(p(\mathcal{X}_j|\vec{\alpha}_j)) = -\log(\prod_{n=l}^{l+n_j-1} p(\vec{X}_n|\vec{\alpha}_j)) = -\sum_{n=l}^{l+n_j-1} \log(p(\vec{X}_n|\vec{\alpha}_j)) \tag{25}$$

We have

$$-\frac{\partial \log(p(\mathcal{X}_j|\vec{\alpha}_j))}{\partial \alpha_{jd}} = -n_j(\Psi(|\vec{\alpha}_j|) - \Psi(\alpha_{jd})) - \sum_{n=1}^{N} \log(\frac{X_{nd}}{1+\sum_{d=1}^{D} X_{nd}}) \quad d=1,\ldots,D \tag{26}$$

$$-\frac{\partial \ln(p(\mathcal{X}_j|\vec{\alpha}_j))}{\partial \alpha_{jD+1}} = -n_j(\Psi(|\vec{\alpha}_j|) - \Psi(\alpha_{jD+1})) - \sum_{n=1}^{N} \log(\frac{1}{1+\sum_{d=1}^{D} X_{nd}}) \tag{27}$$

Then,

$$-\frac{\partial^2 \log(p(\mathcal{X}_j|\vec{\alpha}_j))}{\partial \alpha_{jd_1} \partial \alpha_{jd_2}} = -n_j \Psi'(|\vec{\alpha}_j|) \quad d_1,d_2 = 1,\ldots,D+1 \quad d_1 \neq d_2 \tag{28}$$

$$-\frac{\partial^2 \log(p(\mathcal{X}_j|\vec{\alpha}_j))}{\partial^2 \alpha_{jd}} = -n_j(\Psi'(|\vec{\alpha}_j|) - \Psi'(\alpha_{jd})) \quad d=1,\ldots,D+1 \tag{29}$$

We remark that $F(\vec{\alpha}_j)$ can be written as

$$F(\vec{\alpha}_j) = D_j + \gamma \vec{A}\vec{A}^T \tag{30}$$

where $D_j = diag[n_j\Psi'(\alpha_{j1}), ..., n_j\Psi'(\alpha_{j(D+1)})]$, $\gamma_j = -n_j\Psi'(|\vec{\alpha}_j|)$, $A^T = (a_1, a_2, \ldots, a_{D+1})$ with $a_i = 1 \ \forall i$, then we have [18, Theorem 8.4.3]:

$$\begin{aligned} |F(\vec{\alpha}_j)| &= (1 + \gamma \sum_{d=1}^{D+1} \frac{a_d^2}{D_{dd}}) \prod_{d=1}^{D+1} D_{dd} \\ &= (1 - \Psi'(|\vec{\alpha_j}|) \sum_{d=1}^{D+1} \frac{1}{\Psi'(\alpha_{jd})}) n_j^{D+1} \prod_{d=1}^{D+1} \Psi'(\alpha_{jd}) \end{aligned} \tag{31}$$

By substituting the previous equation and Eq. 24 into Eq. 23, we obtain

$$|F(\Theta)| = \frac{N}{\prod_{j=1}^{M} p_j} \prod_{j=1}^{M} [(1 - \Psi'(|\vec{\alpha}_j|) \sum_{d=1}^{D+1} \frac{1}{\Psi'(\alpha_{jk})}) n_j^{D+1} \prod_{d=1}^{D+1} \Psi'(\alpha_{jd})] \tag{32}$$

Regarding $h(\Theta)$, we make a common assumption in the case of finite mixture models by supposing that $\vec{\alpha}_j$ and the vector $(p_1, \ldots, p_M)$ are independent:

$$h(\Theta) = h(p_1, \ldots, p_M) \prod_{j=1}^{M} h(\vec{\alpha}_j) \tag{33}$$

We will now define the densities $h(\vec{\alpha}_j)$ and $h(p_1, \ldots, p_M)$. We know that the vector $h(p_1, \ldots, p_M)$ is defined on the simplex $\{p_1, \ldots, p_M | \sum_{j=1}^{M-1} p_j < 1\}$, then a natural choice, as a prior, for this vector is a symmetric Dirichlet Distribution

$$h(p_1, \ldots, p_M) = \frac{\Gamma(M\eta)}{\Gamma(\eta)^M} \prod_{j=1}^{M} p_j^{\eta-1} \tag{34}$$

The choice of $\eta = 1$ gives a uniform prior:

$$h(p_1, \ldots, p_M) = (M-1)! \tag{35}$$

As for $h(\vec{\alpha}_j)$ and in the absence of other knowledge about the α_{jk}, $d = 1, \ldots, D+1$, we choose the following uniform prior, which has been found appropriate according to our experimental results, over the range $[0, e^6|\vec{\hat{\alpha}}_j|/\hat{\alpha}_{jk}]$ where $\vec{\hat{\alpha}}_j$ is the estimated vector:

$$h(\alpha_{jk}) = \frac{e^{-6}\hat{\alpha}_{jk}}{|\vec{\hat{\alpha}}_j|} \tag{36}$$

By substituting Eqs. 36 and 35 in Eq. 33, we obtain the following:

$$\log(h(\Theta)) = \sum_{j=1}^{M-1} \log(j) - 6M(D+1) - (D+1)\sum_{j=1}^{M} \log(|\vec{\hat{\alpha}}_j|) + \sum_{j=1}^{M}\sum_{d=1}^{D+1} \log(\hat{\alpha}_{jd}) \qquad (37)$$

The expression of the message length for a finite mixture of inverted Dirichlet distributions is obtained by substituting Eqs. 37 and 32 into Eq. 22. Having this expression in hand, the complete learning algorithm is then as follows:

Complete Learning Algorithm

For each candidate value of M

1. Estimate the parameters of the inverted Dirichlet Distribution using estimation algorithm in the previous subsection.
2. Calculate the associated criterion $MML(M)$ using Eq. 22.
3. Select the optimal model M^* such that $M^* = \arg\min_M MML(M)$.

CHAPTER 3

Experimental Results

3.1 Introduction

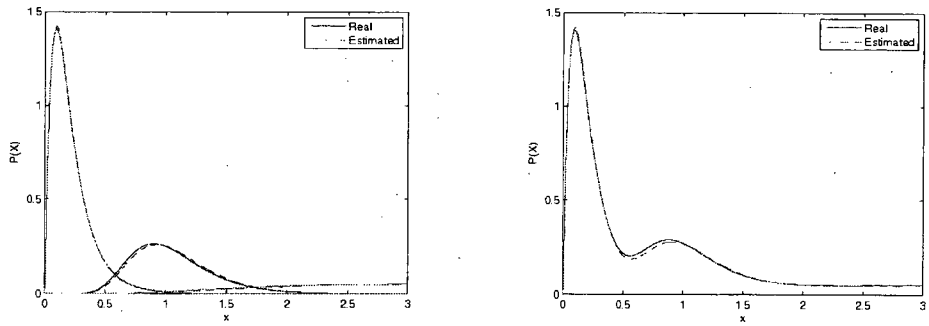
In this section, we first validate our algorithm using synthetic data. The second subsection is intended to show how our model performs on some widely used real data sets. A real-life application which concerns the challenging problem of software modules categorization is investigated in the third subsection.

3.2 Synthetic Data

In this subsection, we report on experiments using one-dimensional and multi-dimensional synthetic data. In [15], a method has been proposed to generate inverted Dirichlet data. Let $X_1, X_2, \ldots, X_{D+1}$ be independent variables which follow Gamma distributions having the same scale but with different parameters $\alpha_1, \alpha_2, \ldots, \alpha_{D+1}$, respectively. Let $Y_d = \frac{X_d}{X_{D+1}}, \quad d = 1, 2, \ldots, D$, then the vector $\vec{Y} = (Y_1, Y_2, \ldots, Y_D)$ has a D-variate inverted Dirichlet distribution with parameter vector $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_{D+1})$. Using this property we can generate M clusters having each n_j D-dimensional vectors which follow inverted Dirichlet distribution with parameter vector $\vec{\alpha}_j = (\alpha_{j1}, \alpha_{j2}, \ldots, \alpha_{jD+1})$. In the following we use this approach to generate both artificial histograms (one-dimensional data) and multi-dimensional data that we shall use to investigate our learning algorithm capabilities.

One-Dimensional Data

We generated artificial histograms from artificial inverted Beta [1] mixture models. Then, we tried to learn the parameters of these artificial histograms (i.e. estimate the parameters of the mixture components and select the number of modes). Figures 3.1 and 3.2 show examples of these artificial histograms. The first histogram represents an inverted Beta mixture of three well separated components while the second one displays overlapped inverted Beta components. The real and estimated parameters of both artificial histograms are shown in tables 3.1 and 3.2. Figure 3.3 shows the number of clusters selected by our algorithm for both histograms and we can observe that in both cases the exact number of clusters ($M = 3$) was favored. For these examples, we can conclude that our algorithm performs well on synthetic data, as there is not a huge difference between real and estimated histograms and their respective parameters, whether the distributions are overlapped or well separated.



(a) Different inverted Beta components.

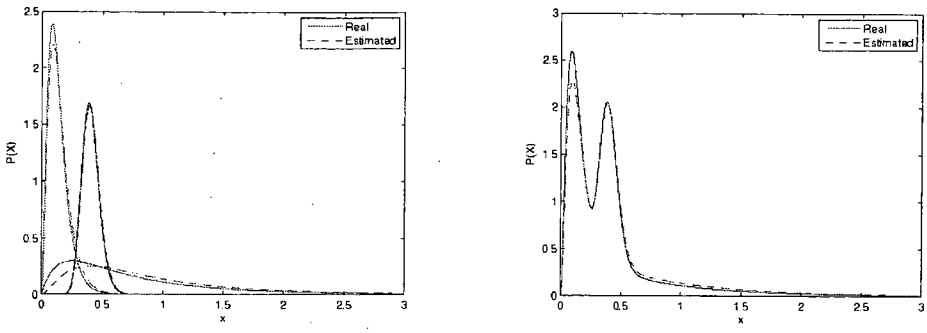
(b) Inverted Beta mixture.

Figure 3.1: First artificial histogram.

Multi-Dimensional Data

We also tested our algorithm on generated multi-dimensional data. In the following, we show some examples of two-dimensional data sets that we have generated to investigate our approach. We use $D = 2$ only for

[1] The inverted Beta is the one-dimensional special case of the inverted Dirichlet obtained when $D = 1$.



(a) Different inverted Beta components.

(b) Inverted Beta mixture.

Figure 3.2: Second artificial histogram.

Table 3.1: Real an estimated parameters of the first artificial histogram (see Figure 3.1) where the mixture components are well separated. The hat denotes the estimated parameters.

	Real parameters	Estimated parameters
Mode 1	$p_1 = 0.33$	$\hat{p}_1 = 0.33$
	$\alpha_{11} = 10$	$\hat{\alpha}_{11} = 10.32$
	$\alpha_{12} = 2$	$\hat{\alpha}_{12} = 2.03$
Mode 2	$p_2 = 0.33$	$\hat{p}_2 = 0.33$
	$\alpha_{21} = 20$	$\hat{\alpha}_{21} = 19.22$
	$\alpha_{22} = 20$	$\hat{\alpha}_{22} = 19.28$
Mode 3	$p_3 = 0.34$	$\hat{p}_3 = 0.34$
	$\alpha_{31} = 2$	$\hat{\alpha}_{31} = 1.92$
	$\alpha_{32} = 10$	$\hat{\alpha}_{32} = 9.70$

ease of representation. In the first example, data were generated from three inverted Dirichlet densities (see Figure 3.4.a) with different parameters as shown in table 3.3. A total of 100 samples for each of the two first densities and a total of 50 samples for the third distribution were taken. The message length values as a function of the number of clusters are presented in table 3.4, where we can see clearly that the MML

Table 3.2: Real an estimated parameters of the second artificial histogram (see Figure 3.2) where the mixture components are overlapped.

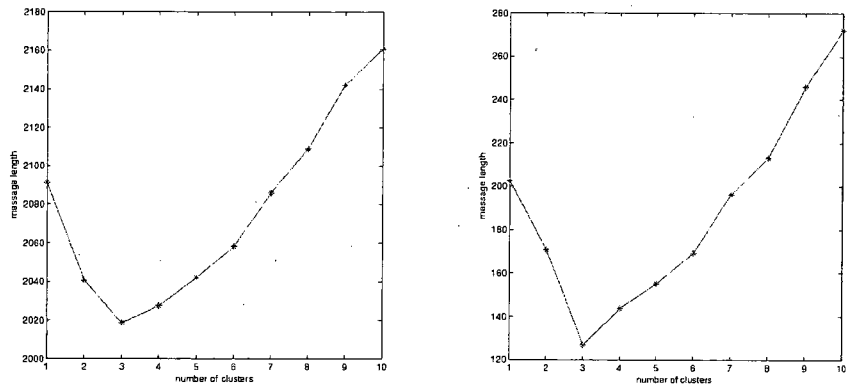
	Real parameters	Estimated parameters
Mode 1	$p_1 = 0.33$	$\hat{p}_1 = 0.33$
	$\alpha_{11} = 3$	$\hat{\alpha}_{11} = 3.12$
	$\alpha_{12} = 23$	$\hat{\alpha}_{12} = 24.24$
Mode 2	$p_2 = 0.33$	$\hat{p}_2 = 0.34$
	$\alpha_{21} = 43$	$\hat{\alpha}_{21} = 39.19$
	$\alpha_{22} = 108$	$\hat{\alpha}_{22} = 99.70$
Mode 3	$p_3 = 0.34$	$\hat{p}_3 = 0.33$
	$\alpha_{31} = 2$	$\hat{\alpha}_{31} = 2.15$
	$\alpha_{32} = 3$	$\hat{\alpha}_{32} = 3.01$

criterion found the exact number of clusters. In the second example, data were generated from five inverted

Table 3.3: Real and estimated mixture parameters for the first two-dimensional generated data set.

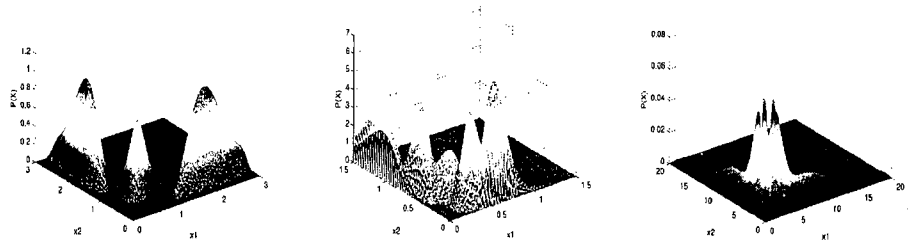
Cluster 1	$p_1 = 0.4$	$\alpha_{11} = 15$	$\alpha_{21} = 65$	$\alpha_{31} = 30$
	$\hat{p}_1 = 0.4$	$\hat{\alpha}_{11} = 14.83$	$\hat{\alpha}_{21} = 64.38$	$\hat{\alpha}_{31} = 29.58$
Cluster 2	$p_2 = 0.4$	$\alpha_{12} = 65$	$\alpha_{22} = 15$	$\alpha_{32} = 30$
	$\hat{p}_2 = 0.4$	$\hat{\alpha}_{12} = 64.06$	$\hat{\alpha}_{22} = 14.90$	$\hat{\alpha}_{32} = 29.69$
Cluster 3	$p_3 = 0.2$	$\alpha_{13} = 30$	$\alpha_{23} = 34$	$\alpha_{33} = 35$
	$\hat{p}_3 = 0.2$	$\hat{\alpha}_{13} = 30.54$	$\hat{\alpha}_{23} = 34.32$	$\hat{\alpha}_{33} = 35.51$

Dirichlet densities (see Figure 3.4.b) with different parameters as shown in table 3.5. A total of 100 samples for each of densities were taken. According to table 3.6, we can see that the MML found the exact number of clusters. In the third example, data were generated from six inverted Dirichlet densities (see Figure 3.4.c) with different parameters as shown in table 3.7. A total of 100 samples were taken from the four first densities and 50 samples from the two last densities. Again the MML criterion was able to find the exact



(a) (b)

Figure 3.3: Number of clusters automatically selected using MML criterion for the: (a) first artificial histogram and (b) second artificial histogram.



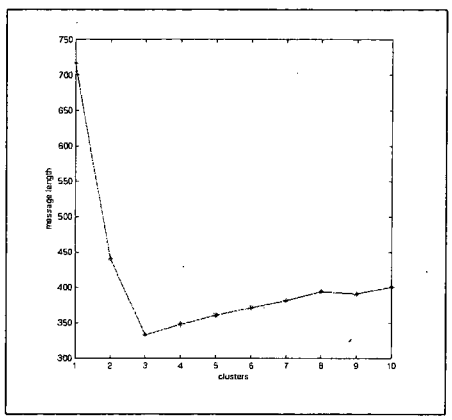
(a) (b) (c)

Figure 3.4: Three two-dimensional artificial mixture models with: (a) three, (b) five and (c) six components.

number of clusters as shown in table 3.8.

Finally table 3.9 shows an example of real and estimated parameters in the case of a 4-dimensional data set generated from a 3-components finite inverted Dirichlet mixture. Again our algorithm was able to estimate accurately the parameters and to select the exact number of clusters (see table 3.10). In the next section we validate our algorithm with real data.

Table 3.4: Message length values as a function of the number of clusters for the first two-dimensional generated data set.



M	Message Length
1	716.51
2	440.24
3	**333.05**
4	347.61
5	360.78
6	371.36
7	381.71
8	393.98
9	390.93
10	400.61

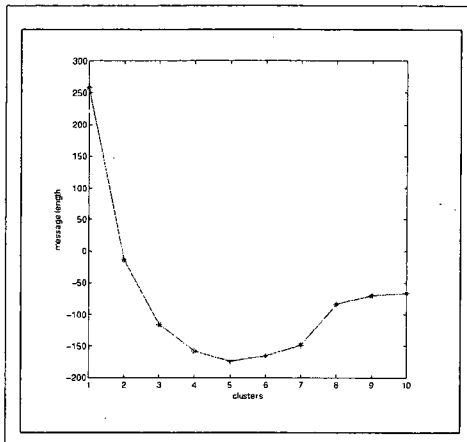
Table 3.5: Real and estimated mixture parameters for the second two-dimensional generated data set.

Cluster 1	$p_1 = 0.2$	$\alpha_{11} = 30$	$\alpha_{21} = 4$	$\alpha_{31} = 50$
	$\hat{p}_1 = 0.2$	$\hat{\alpha}_{11} = 28.84$	$\hat{\alpha}_{21} = 4.00$	$\hat{\alpha}_{31} = 48.50$
Cluster 2	$p_2 = 0.2$	$\alpha_{12} = 2$	$\alpha_{22} = 40$	$\alpha_{32} = 34$
	$\hat{p}_2 = 0.2$	$\hat{\alpha}_{12} = 2.08$	$\hat{\alpha}_{22} = 41.10$	$\hat{\alpha}_{32} = 34.84$
Cluster 3	$p_3 = 0.2$	$\alpha_{13} = 15$	$\alpha_{23} = 20$	$\alpha_{33} = 40$
	$\hat{p}_3 = 0.2$	$\hat{\alpha}_{13} = 15.54$	$\hat{\alpha}_{23} = 20.87$	$\hat{\alpha}_{33} = 41.17$
Cluster 4	$p_4 = 0.2$	$\alpha_{14} = 10$	$\alpha_{24} = 40$	$\alpha_{34} = 60$
	$\hat{p}_4 = 0.2$	$\hat{\alpha}_{14} = 10.28$	$\hat{\alpha}_{24} = 41.17$	$\hat{\alpha}_{34} = 62.11$
Cluster 5	$p_5 = 0.2$	$\alpha_{15} = 20$	$\alpha_{25} = 10$	$\alpha_{35} = 50$
	$\hat{p}_5 = 0.2$	$\hat{\alpha}_{15} = 19.32$	$\hat{\alpha}_{25} = 9.93$	$\hat{\alpha}_{35} = 48.51$

3.3 Real Data

In this section we investigate the performance of our algorithm and compare the modeling capabilities of inverted Dirichlet and Gaussian mixtures using two well-known data sets. The classification was performed

Table 3.6: Message length values as a function of the number of clusters for the second two-dimensional generated data set.

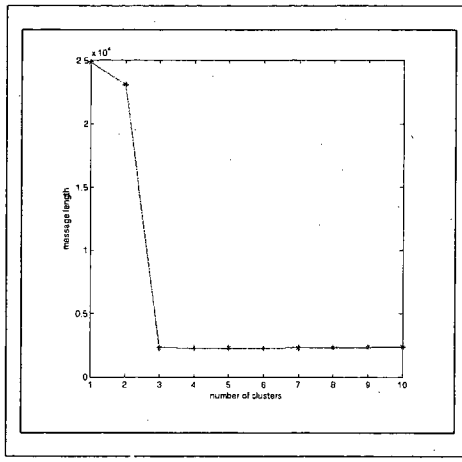


M	Message Length
1	258.10
2	−13.72
3	−115.70
4	−157.76
5	**-174.47**
6	−165.64
7	−148.34
8	−84.08
9	−70.12
10	−66.17

Table 3.7: Real and estimated mixture parameters for the third two-dimensional generated data set.

Cluster 1	$p_1 = 0.2$	$\alpha_{11} = 20$	$\alpha_{21} = 30$	$\alpha_{31} = 10$
	$\hat{p}_1 = 0.2$	$\hat{\alpha}_{11} = 20.01$	$\hat{\alpha}_{21} = 30.20$	$\hat{\alpha}_{31} = 10.15$
Cluster 2	$p_2 = 0.2$	$\alpha_{12} = 10$	$\alpha_{22} = 2$	$\alpha_{32} = 3$
	$\hat{p}_2 = 0.2$	$\hat{\alpha}_{12} = 9.99$	$\hat{\alpha}_{22} = 2.00$	$\hat{\alpha}_{32} = 3.06$
Cluster 3	$p_3 = 0.2$	$\alpha_{13} = 2$	$\alpha_{23} = 10$	$\alpha_{33} = 3$
	$\hat{p}_3 = 0.19$	$\hat{\alpha}_{13} = 2.05$	$\hat{\alpha}_{23} = 10.69$	$\hat{\alpha}_{33} = 3.08$
Cluster 4	$p_4 = 0.2$	$\alpha_{14} = 30$	$\alpha_{24} = 20$	$\alpha_{34} = 10$
	$\hat{p}_4 = 0.21$	$\hat{\alpha}_{14} = 30.10$	$\hat{\alpha}_{24} = 20.17$	$\hat{\alpha}_{34} = 10.01$
Cluster 5	$p_5 = 0.1$	$\alpha_{15} = 70$	$\alpha_{25} = 6$	$\alpha_{35} = 7$
	$\hat{p}_5 = 0.11$	$\hat{\alpha}_{15} = 71.85$	$\hat{\alpha}_{25} = 6.11$	$\hat{\alpha}_{35} = 7.18$
Cluster 6	$p_6 = 0.1$	$\alpha_{16} = 6$	$\alpha_{26} = 70$	$\alpha_{36} = 7$
	$\hat{p}_6 = 0.09$	$\hat{\alpha}_{16} = 6.45$	$\hat{\alpha}_{26} = 74.07$	$\hat{\alpha}_{36} = 7.14$

Table 3.8: Message length values as a function of the number of clusters for the third two-dimensional generated data set.



M	Message Length
1	24839
2	23077
3	2298
4	2289
5	2290
6	**2284**
7	2299
8	2324
9	2340
10	2370

using the Bayesian decision rule after the classes densities were estimated and the number of clusters was selected.

The Haberman dataset

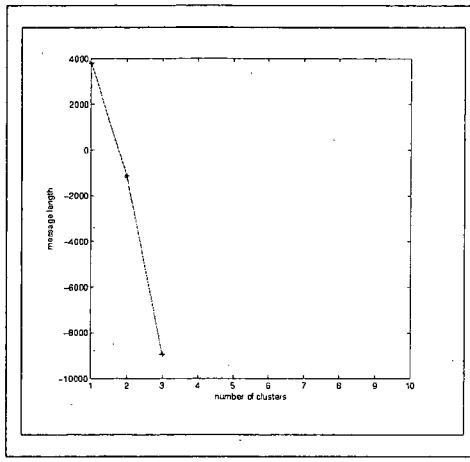
The first data set, called Haberman dataset [21], contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. The dataset contains 306 instances, and four attributes including the class attribute. These attributes are: age of patient at time of operation, patient's year of operation, number of positive axillary nodes detected, survival status (1 = the patient survived 5 years or longer, 2 = the patient died within 5 year). It has 225 instances from class 1, and 81 instances from class 2. By applying our algorithm to this dataset, the MML criterion found that $M = 2$ leads us to the minimum message length. So we have two classes, which meets the specification of our dataset (see table 3.11). Using Gaussian mixture, however, we failed to obtain the exact number of clusters (i.e. $M = 3$ was wrongly favored) as shown in table 3.12.
In our classification, we consider a true positive a patient who survived 5 years or longer attributed to class 1, a false negative a patient who died within 5 years attributed to class 1, a false positive a patient who survived

Table 3.9: Real and estimated parameters in the case of a 4-dimensional data set generated from a 3-components finite inverted Dirichlet mixture

	Real parameters	Estimated parameters
Cluster 1	$p_1 = 0.33$	$\hat{p}_1 = 0.33$
	$\alpha_{11} = 3$	$\hat{\alpha}_{11} = 3.12$
	$\alpha_{12} = 23$	$\hat{\alpha}_{12} = 23.29$
	$\alpha_{13} = 98$	$\hat{\alpha}_{13} = 100.59$
	$\alpha_{14} = 23$	$\hat{\alpha}_{14} = 23.51$
	$\alpha_{15} = 199$	$\hat{\alpha}_{15} = 203.46$
Cluster 2	$p_2 = 0.33$	$\hat{p}_2 = 0.33$
	$\alpha_{21} = 43$	$\hat{\alpha}_{21} = 43.63$
	$\alpha_{22} = 108$	$\hat{\alpha}_{22} = 108.69$
	$\alpha_{23} = 23$	$\hat{\alpha}_{23} = 23.38$
	$\alpha_{24} = 34$	$\hat{\alpha}_{24} = 34.28$
	$\alpha_{25} = 92$	$\hat{\alpha}_{25} = 93.27$
Cluster 3	$p_3 = 0.34$	$p_3 = 0.34$
	$\alpha_{31} = 2$	$\hat{\alpha}_{31} = 2.02$
	$\alpha_{32} = 3$	$\hat{\alpha}_{32} = 3.00$
	$\alpha_{33} = 9$	$\hat{\alpha}_{33} = 9.02$
	$\alpha_{34} = 90$	$\hat{\alpha}_{34} = 91.13$
	$\alpha_{35} = 23$	$\hat{\alpha}_{35} = 23.52$

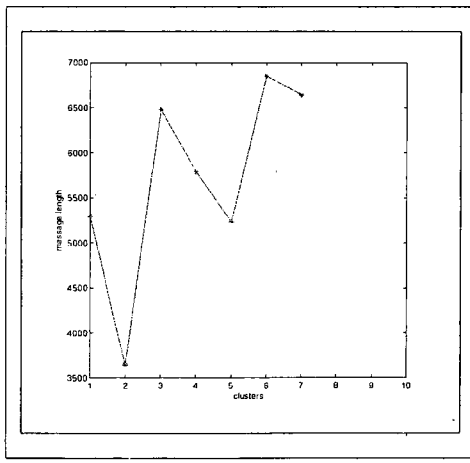
5 years or longer attributed to class 2, a true negative a patient who died within 5 years attributed to class 2. In this case, there are two types of misclassification type I and type II. Type II misclassification occurs when a patient who survived 5 years or longer is wrongly classified as a patient who died within 5 years and type II misclassification occurs when a patient who died within 5 years is mistakenly classified as a patient who survived 5 years or longer (see table 3.13). The classification was performed using the Bayesian decision rule after the classes densities were estimated. Table 3.14 displays the confusion matrices for Haberman

Table 3.10: Message length values as a function of the number of clusters for the four-dimensional generated data set.



M	Message Length
1	3810.5
2	-1130.5
3	**-8933**
4	N/A
5	N/A
6	N/A
7	N/A
8	N/A
9	N/A
10	N/A

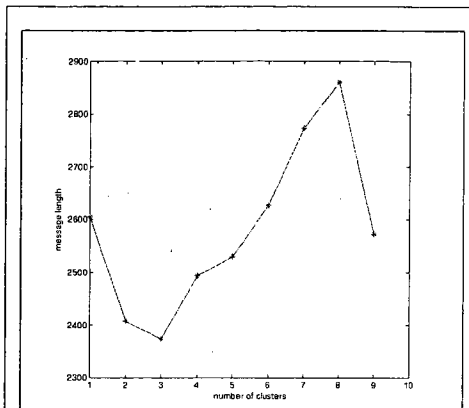
Table 3.11: The message length as a function of the number of cluster in the case of the Haberman dataset when using inverted Dirichlet mixture.



M	Message Length
1	5297.5
2	**3645.6**
3	6480.6
4	5791.3
5	5234.1
6	6851.5
7	6640.4
8	N/A
9	N/A
10	N/A

dataset classification when using both the inverted Dirichlet mixture and the Gaussian mixture which we have forced to consider $M = 2$. Having these confusion matrices in hand, we can use them to compute

Table 3.12: The message length as a function of the number of cluster in the case of the Haberman dataset when using Gaussian mixture.



M	Message Length
1	2604.7
2	2407.4
3	**2372.6**
4	2492.9
5	2529.8
6	2627.0
7	2773.2
8	2860.3
9	2572.7
10	N/A

Table 3.13: Confusion matrix representation for the Haberman dataset. S: patient who survived 5 years or longer, D: patient who died within 5 years.

	S	**D**
S	True positive (TP)	False positive (FP)
D	False negative (FN)	True negative (TN)

widely used measures such as the accuracy, precision, false positive rate, false negative rate, specificity, and sensitivity as we can see in table 3.15. From this table we notice that the classification based on the inverted Dirichlet mixture is significantly more accurate and precise than the one based on the Gaussian mixture.
Let p (positive) stands for a patient who survived for 5 years or longer, and n (negative) stands for a patient who died within 5 years. Let $+$ be the test event of a patient who survived 5 years or longer, and $-$ be the test event a patient who died within 5 years. We need to know the following:

- $P(p)$, or the probability that the patient survived 5 years or longer regardless of any other information. In our case this is 0.7353.
- $P(n)$, or the probability that the patient died within 5 years. This is $1 - P(p)$ or 0.2647.

Table 3.14: Confusion matrices for Haberman dataset classification using both the inverted Dirichlet and the Gaussian mixture models.

inverted Dirichlet

	S	**D**
S	195	30
D	51	30

Gaussian

	S	**D**
S	150	75
D	27	54

Table 3.15: Test results of inverted Dirichlet and Gaussian mixtures for Haberman dataset classification.

	Expression	Inverted Dirichlet	Gaussian
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$	0.74	0.66
Precision	$\frac{TP}{TP+FP}$	0.87	0.66
False positive rate	$\frac{FP}{TN+FP}$	0.50	0.58
False negative rate	$\frac{FN}{TP+FN}$	0.21	0.15
specificity	$\frac{TN}{TN+FP}$	0.50	0.41
sensitivity	$\frac{TP}{TP+FN}$	0.79	0.84

- $P(+|p)$, or the probability that the test is positive, given that the patient survived 5 years or longer. This is equal to the sensitivity of our test.
- $P(-|p)$, or the probability that the test is negative, given that the patient survived 5 years or longer. This is equal to Type I error of our test.
- $P(+|n)$, or the probability that the test is positive, given that the patient died within 5 years. This is equal to Type II error of our test.
- $P(-|n)$, or the probability that the test is negative, given that the patient died within 5 years. This is equal to the specificity of our test.

Let us now compute the following probabilities:

- $P(p|+)$, or the probability that the patient survived 5 years or longer, given that the test is positive.
- $P(p|-)$, or the probability that the patient survived 5 years or longer, given that the test is

negative.

- $P(n|-)$, or the probability that the patient died within 5 years, given that the test is negative.
- $P(n|+)$, or the probability that the patient died within 5 years, given that the test is positive.

From table 3.16 we notice that the inverted Dirichlet mixture significantly outperforms the Gaussian mixture in its credibility while giving a negative test result, telling that the patient died within 5 years. Concerning the positive test result event telling the patient survived 5 years or longer, inverted Dirichlet mixture and Gaussian mixture give close results.

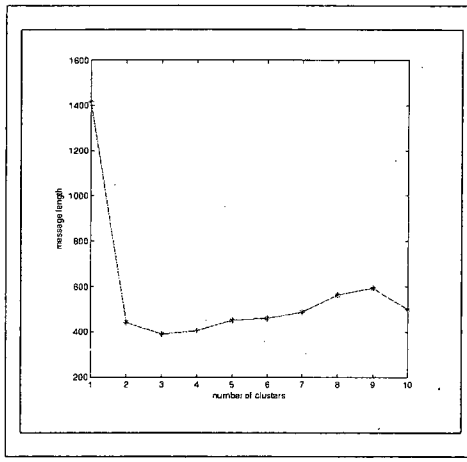
Table 3.16: Posterior results of inverted Dirichlet and Gaussian mixtures for Haberman dataset classification problem.

	Expression	Inverted Dirichlet	Gaussian
$P(p\|+)$	$\frac{P(+\|p)P(p)}{P(+\|p)P(p)+P(+\|n)P(n)}$	0.91	0.93
$P(p\|-)$	$\frac{P(-\|p)P(p)}{P(-\|p)P(p)+P(-\|n)P(n)}$	0.73	0.79
$P(n\|-)$	$\frac{P(-\|n)P(n)}{P(-\|n)P(n)+P(-\|p)P(p)}$	0.26	0.20
$P(n\|+)$	$\frac{P(+\|n)P(n)}{P(+\|n)P(n)+P(+\|p)P(p)}$	0.08	0.06

Iris Dataset

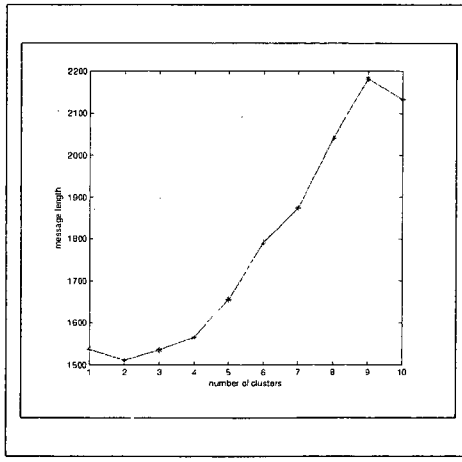
The second dataset, called Iris data set [21], contains 3 classes having 50 instances each, where each class refers to a type of Iris plant (Iris setosa, Iris virginica and Iris versicolor). One class is linearly separable from the other two; the latter are not linearly separable from each other. For each instance we have four attributes: sepal length in cm, sepal width in cm, petal length in cm, and petal width in cm. By applying our algorithm to the Iris dataset, the MML criterion found the exact number of clusters (i.e. $M = 3$ as shown in table 3.17) which was not the case when using Gaussian mixtures since $M = 2$ was favored as shown in table 3.18. The confusion matrices for Iris dataset classification using inverted Dirichlet and Gaussian, forced to consider 3 clusters, mixtures are given in tables 3.19 and 3.20, respectively. In this confusion matrices, the cell ($class_i$, $class_j$) represents the number of instances from $class_j$ which are classified as $class_i$. We notice that the two mixture give similar results. We can see that the errors are generated in

Table 3.17: The message length as a function of the number of cluster in the case of the Iris dataset using inverted Dirichlet mixture.



M	Message Length
1	1416.3
2	442.5
3	**389.1**
4	404.5
5	451.2
6	459.7
7	486.4
8	563.4
9	593.2
10	499.4

Table 3.18: The message length as a function of the number of cluster in the case of the Iris dataset using using Gaussian Mixture.



M	Message Length
1	1536.3
2	**1510.6**
3	1535.4
4	1565.6
5	1655.6
6	1791.3
7	1873.8
8	2040.9
9	2182.6
10	2132.6

the second class classification whose density is overlapping with the third class density, and then we had a misclassification of 6 and 5 plants from the second class, classified as being from the third class, respectively

Table 3.19: Confusion matrix for Iris Dataset using Inverted Dirichet Mixture.

	Class1	Class2	Class3
Class1	50	0	0
Class2	0	44	0
Class3	0	6	50

Table 3.20: Confusion matrix for Iris Dataset using Gaussian Mixture.

	Class1	Class2	Class3
Class1	50	0	0
Class2	0	45	0
Class3	0	5	50

for the inverted Dirichlet and gaussian mixtures.

3.4 Complexity-Based Classification of Software Modules

There is currently much interest in the problem of providing good data-based models for quality improvement in systems engineering. The problem has been the subject of extensive research in the past (see, for instance, [22]) and has been extended to the important domain of software engineering [23]. Indeed, probability models and statistical methods are now a popular technique for evaluating the reliability of computer software and quantifying its performance before its release into the marketplace. Developing and maintaining a given software system is a challenging problem that has a lot of difficulties. Software is composed of a great number of relatively independent units called modules (i.e. a set of source-code files) which perform certain functions. One way to test software quality is to determine the number of faults in each module.

These faults may be related, for instance, to changes [2] to source code happening while the software is executing [25] and are in general in a small portion [3] of the modules [27]. Most of the time, people are not concerned about the exact number of changes, rather than setting a threshold. If the number of faults (i.e. defects in a program that can cause incorrect execution [28]) found in certain module exceeds this previously set criterion, it is regarded as fault-prone, otherwise non fault-prone [29]. For example, if a threshold of two faults is set, each module having two or more changes will be assigned to the fault-prone group and considered unstable, with high-risk and might cause failure. A software prediction model is viewed as an empirical tool using a certain algorithm to forecast modules types (i.e fault-prone or non fault-prone) and should be easy to interpret. A key common characteristic of these prediction models is that they establish a relationship between the measures of modules attributes and the types [30]. The fundamental construction of the predictive models is based upon the faults and corresponding measures collected from past similar program development and maintenance scenarios. When the model is built, we can determine the quality and reliability of new modules, if the measures of their attributes are in hand. The understanding of the modules through prediction models helps to target high-risk modules which need priority attention, extensive testing, redesign and improvement in early life cycle [31], which is very valuable, cost-effective, and improve the efficiency of inspection efforts.

In this section, we investigate our algorithm on the well-known MIS data set [28]. MIS is a widely used commercial software system consisting of about 4500 routines written in approximate 400,000 lines of Pascal, FORTRAN, and PL/M assembly code. The practical number of changes (faults) as well as 11 software complexity metrics of each module in this program were determined during three-years system testing and maintenance. Basically, the MIS data set used in this paper, is composed of 390 modules and each module is described by 11 complexity metrics acting as variables:

- LOC is the number of lines of code, including comments.
- CL is the number of lines of code, excluding comments.
- TChar is the number of characters
- TComm is the number of comments.

[2] See [24] for a discussion about the types and classes of changes that may occur.

[3] According to the 80/20 rule (i.e Pareto rule), about 20 percent of a software system is responsible for 80 percent of its errors, costs and rework [26].

- MChar is the number of comment characters.
- DChar is the number of code characters
- $N = N_1 + N_2$ is the program length, where N_1 is the total number of operators and N_2 is the total number of operands.
- $\hat{N} = \eta_1 \log_2 \eta_1 + \eta_2 \log_2 \eta_2$ is an estimated program length, where η_1 is the number of unique operators and η_2 is the number of unique operands.
- $N_F = (\log_2 \eta_1)! + (\log_2 \eta_2)!$ is Jensen's estimator of program length.
- V(G), McCabe's cyclomatic number, is one more than the number of decision nodes in the control flow graph.
- BW is Belady's bandwidth metric, where $BW = \frac{1}{n} \sum_i i L_i$ and L_i represents the number of nodes at level i in a nested control flow graph of n nodes. This metric indicates the average level of nesting or width of the control flow graph representation of the program.

In documented MIS data set, modules 1 to 114 are regarded as non fault-prone (number of faults less than 2), and the others 276 instances are considered to be fault-prone. In our classification, we consider a true positive a non fault-prone module classified as a non fault prone module, a true negative a fault-prone module classified as a fault-prone module, a false negative a fault-prone module classified as a non fault-prone module, and a false positive a non fault-prone module classified as a fault-prone module. In the case of our problem, there are two types of misclassification, type I and type II. Type II misclassification occurs when a non fault-prone module is wrongly classified as fault-prone and type II misclassification occurs when a fault-prone modules is mistakenly classified as non fault-prone (see table 3.21). The main goal of

Table 3.21: Confusion Matrix Representation for the Software Modules

	Non fault-prone (NF)	**Fault-prone (F)**
Non fault-prone (NF)	True Positive (TP)	False positive (FP)
Fault-prone (F)	False negative (FN)	True negative (TN)

this application is to compare the performances of inverted Dirichlet and Gaussian mixtures. The confusion matrices computed for both mixtures are shown in table 3.22. These confusion matrices were used then to compute the accuracy, precision, false positive rate, false negative rate, specificity, and sensitivity (see

Table 3.22: Confusion matrices for the software modules classification problem.

Inverted Dirichlet

	NF	F
NF	91	23
F	85	191

Gaussian

	NF	F
NF	104	10
F	160	116

table 3.23). From table 3.23 we notice that the classification based on the inverted Dirichlet mixture is significantly more accurate but a little bit less precise than the classification based on the Gaussian mixture.

Table 3.23: Test results of inverted Dirichlet and Gaussian mixtures for the software modules classification.

	Inverted Dirichlet	Gaussian
Accuracy	0.7231	0.5641
Precision	0.7982	0.9123
False Positive rate	0.1075	0.0794
False Negative rate	0.4830	0.6061
Specificity	0.8925	0.9206
Sensitivity	0.5170	0.3939

We can use the Bayes's theorem to analyze our results basing on the samples of data that we have. Let p (positive) stands for being non fault prone module, and n (negative) stands for being fault prone module. Let $+$ be the event of a non fault prone module test, and $-$ be the event a fault prone module test. We need to know the following:

- $P(p)$, or the probability that the software is non fault prone regardless of any other information. In our case this is 0.2923.
- $P(n)$, or the probability that the software is a fault prone. This is $1 - P(p)$ or 0.7077
- $P(+/p)$, or the probability that the test is positive, given that the module is non fault prone.

This is equal to the sensitivity of our test.

- $P(-/p)$, or the probability that the test is negative, given that the module is non fault prone. This is equal to Type I error of our test.
- $P(+/n)$, or the probability that the test is positive, given that the module is fault prone. This is equal to Type II error of our test.
- $P(-/n)$, or the probability that the test is negative, given that the module is fault prone. This is equal to the specificity of our test.

Let us now compute the following probabilities:

- $P(p/+)$, or the probability that the module is non fault prone, given that the test is positive.
- $P(p/-)$, or the probability that the module is non fault prone, given that the test is negative.
- $P(n/-)$, or the probability that the module is fault prone, given that the test is negative.
- $P(n/+)$, or the probability that the module is fault prone, given that the test is positive.

Table 3.24: Posterior results of inverted Dirichlet and Gaussian mixture for the software modules classification problem.

	Inverted Dirichlet	Gaussian
$P(p/+)$	0.3065	0.2116
$P(p/-)$	0.0474	0.0344
$P(n/-)$	0.9526	0.9656
$P(n/+)$	0.6935	0.7884

From table 3.24 we notice that inverted Dirichlet mixture significantly outperforms the Gaussian mixture in its credibility while giving a positive test result, telling that a module is non fault-prone. Concerning the negative test result event telling the module is fault-prone, inverted Dirichlet mixture and Gaussian mixture provide comparable good results.

CHAPTER 4

Conclusions

The main goal of this work is to find meaningful structure in a set of unlabeled non Gaussian positive vectors through inverted Dirichlet mixture-based modeling. The specific choice of the inverted Dirichlet is motivated by its excellent properties namely its flexibility to approximate many shapes since, in contrast to the Gaussian, it can be symmetric, skewed to the right or skewed to the left. Given unlabeled data that is generated from this mixture we have two related tasks, one is to estimate the parameters of the mixture distribution, and the other usually referred as mixture selection, is to determine the appropriate number of clusters. For the first task we have used the maximum likelihood approach performed via a hybrid of EM and Newton Raphson. The second task has been based on the MML criterion that we have developed for inverted Dirichlet mixture. Through extensive experiments involving synthetic data, real data and the challenging problem of software modules categorization we have shown that our unsupervised learning algorithm leads us to promising results by searching efficiently the space of cluster locations and the number of clusters and by optimizing the message length of the data. The key to the success of the MML criterion is the introduction of prior information about the mixture model's parameters. Moreover, we have proved that the inverted Dirichlet distribution can be a powerful tool to analyze non gaussian data. There are many avenues for future work. We believe that the integration of feature selection, using for instance a similar approach as in [32], could improve further the accuracy of the model learning. Indeed, when there are too many features, recent studies have shown that it is highly desirable to discard weak irrelevant features which may compromise the classification results. A promising extension of this work that we intend to pursue would be also the investigation of online learning techniques and Bayesian approaches as done previously, in the

case of Dirichlet mixtures, in [33] and [34], respectively. Finally, and most importantly, it must be stressed that the model that we proposed may have many other potential applications since positive data are naturally generated in many other domains such as computer vision, image processing, bioinformatics and natural language processing.

CHAPTER 5

Appendices

5.1 Finite Gaussian Mixture Model

The multivariate Gaussian probability density function is the common assumption when using finite mixture models and is given by

$$p(\vec{X}_n|\theta_j) = \frac{exp(\frac{1}{2}(\vec{X}_n - \vec{\mu}_j)^T \sum_j^{-1}(\vec{X}_n - \vec{\mu}_j))}{(2\pi)^{d/2}|\sum_j|^{1/2}} \tag{1}$$

Thus, in the case of a finite Gaussian mixture model, we have $\theta_j = (\vec{\mu}_j, \sum_j)$ with $\vec{\mu}_j$ the dim-dimensional mean vector, and $\sum_j$ the dim x dim covariance matrix. The estimation of these parameters is done through the maximum likelihood (ML) estimate:

$$\hat{\Theta}_{ML} = arg\max_{\Theta} L(\mathcal{X}|\Theta) \tag{2}$$

where $L(\mathcal{X}|\Theta)$ is the log-likelihood corresponding to a M-components and N vectors:

$$L(\mathcal{X}|\Theta) = \log \prod_{n=1}^{N} p(\vec{X}_n|\Theta) = \sum_{n=1}^{N} \log \left(\sum_{j=1}^{M} p(\vec{X}_n|\theta_j)p_j\right) \tag{3}$$

The maximization defining the ML estimates is subject to the constraints over the mixing parameters and cannot be found analytically. Typically the the ML estimates of the mixture parameters can be obtained using expectation maximization (EM) and related techniques. The maximization of Eq. 3 is generally performed within the EM framework where each $\vec{X}_n$ is supposed to have arisen from one the M clusters. Thus, let $\mathcal{Z} = \{\vec{Z}_1, \ldots, \vec{Z}_N\}$ denote the missing group-indicator vectors where the jth element of $\vec{Z}_n$, Z_{nj}, is equal

to one if $\vec{X}_n$ belongs to cluster j and zero, otherwise. The complete data in this case are $(\mathcal{X}, \mathcal{Z})$ and the associated complete-data log likelihood is given by

$$\Phi_c(\mathcal{X}, \mathcal{Z}|\Theta) = \sum_{j=1}^{M}\sum_{n=1}^{N} Z_{nj}\Big(\log p_j + \log p(\vec{X}_n|\theta_j)\Big) \tag{4}$$

The EM algorithm proceeds iteratively in two steps, The expectation (E) step and the maximization (M) step. In the E-step, we compute the conditional expectation of $\Phi_c(\mathcal{X}, \mathcal{Z}|\Theta)$ which is reduced to the computation of the posterior probabilities (i.e. the probability that a vector $\vec{X}_n$ is assigned to a cluster j):

$$p(j|\vec{X}_n, \theta_j) = \frac{p_j p(\vec{X}_n|\theta_j)}{\sum_{j=1}^{M} p_j p(\vec{X}_n|\theta_j)} \tag{5}$$

Then, the conditional expectation of the complete-data log-likelihood given by

$$Q(\mathcal{X}, \Theta) = \sum_{j=1}^{M}\sum_{n=1}^{N} p(j|\vec{X}_n, \theta_j)\Big(\log p_j + \log p(\vec{X}_n|\theta_j)\Big) \tag{6}$$

is maximized in the M-step. To resolve this optimization problem, we must determine the solution to $\frac{\partial}{\partial\Theta}Q(\mathcal{X}, \Theta) = 0$. The EM algorithm produces a sequence of estimates $\Theta_t, t = 0, 1, 2...$ by applying two steps in alternation until some convergence criterion is satisfied:

1. E-Step : Compute $p(j|\vec{X}_n, \theta_j)$ given the parameter estimates from the initialization

$$p(j|\vec{X}_n, \theta_j) = \frac{p_j p(\vec{X}_n|\theta_j)}{\sum_{j=1}^{M} p_j p(\vec{X}_n|\theta_j)} \tag{7}$$

2. M-Step : Update the parameter estimates according to $\hat{\Theta}_{ML} = arg\max_{\Theta} L(\mathcal{X}|\Theta)$:

$$p_j^{(t+1)} = \frac{1}{N}\sum_{n=1}^{N} p(j|\vec{X}_n, \theta_j) \tag{8}$$

$$\vec{\mu}_j^{(t+1)} = \frac{\sum_{n=1}^{N} p(j|\vec{X}_n, \theta_j)\vec{X}_n}{N p_j} \tag{9}$$

$$\sum_j{}^{(t+1)} = \frac{\sum_{n=1}^{N} p(j|\vec{X}_n, \theta_j)[(\vec{X}_n - \vec{\mu}_j^{(t)})(\vec{X}_n - \vec{\mu}_j^{(t)})^T]}{N p_j} \tag{10}$$

5.2 MML for Gaussian Mixture

The message length for a mixture of distributions is given by [19]

$$MessLen \simeq -\log(h(\Theta)) - L(\mathcal{X}|\Theta) + \frac{1}{2}\log(|F(\Theta)|) + \frac{N_p}{2}(1 - \log(12)) \tag{11}$$

where $h(\Theta)$ is the prior probability, $p(\mathcal{X}|\Theta)$ is the likelihood, $F(\Theta)$ is the expected Fisher information matrix, $|F(\Theta)|$ is its determinant, and N_p is the number of free parameters to be estimated. To use Eq. 11 for a gaussian mixture, we must first choose a prior distribution $h(\Theta)$ and derive an expression for the determinant of Fisher Information matrix, $F(\Theta)$. Olivier, Baxter and Wallace [35], demonstrated that a convenient prior for the gaussian mixture is :

$$h(\Theta) = \frac{(M-1)!}{2^{Mdim}} \prod_{k=1}^{dim} (\frac{1}{\sigma_{popk}})^{2M} \tag{12}$$

where $\sigma_{pop} = (\sigma_{pop1}, \sigma_{pop2}, ..., \sigma_{popdim})$ is the standard deviation of the entire population. The determinant of the Fisher Information matrix is

$$\frac{1}{2}\log(|F(\Theta)|) \simeq \sum_{k=1}^{dim}\sum_{j=1}^{M} \log\frac{\sqrt{2}N_j}{\sigma_{jk}^2} + \frac{1}{2}\log(N) - \frac{1}{2}\sum_{j=1}^{M}\log(p_j) \tag{13}$$

with

$$N_j = p_j * N \tag{14}$$

and

$$\sigma_{jk} = \sqrt{\frac{\sum_{n=1}^{N} p(j|\vec{X}_n, \theta_j)(X_{nk} - \mu_{jk})^2}{N_j}} \tag{15}$$

By substituting Eq. 12 and Eq. 13 in Eq. 11 we have :

$$MessLen \simeq -\log[\frac{(M-1)!}{2^{Mdim}} \prod_{k=1}^{dim} (\frac{1}{\sigma_{popk}})^{2M}] - L(\mathcal{X}|\Theta) + \sum_{k=1}^{dim}\sum_{j=1}^{M} \log\frac{\sqrt{2}N_j}{\sigma_{jk}^2} + \frac{1}{2}\log(N) - \frac{1}{2}\sum_{j=1}^{M}\log(p_j) + \frac{N_p}{2}(1-\log(12)) \tag{16}$$

Complete Learning Algorithm

For each candidate value of M

1. Estimate the parameters of the Gaussian mixture using the estimation algorithm in the previous subsection.

2. Calculate the associated criterion $MML(M)$ using Eq. 16.

3. Select the optimal model M^* such that $M^* = \arg\min_M MML(M)$.

List of References

[1] A. K. Jain, M. Murty and P. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[2] C. Glymour, D. Madigan, D. Pregibon and P. Smyth. Statistical Inference and Data Mining. *Communications of the ACM*, 39(11):35–41, 1996.

[3] C. Glymour, D. Madigan, D. Pregibon and P. Smyth. Statistical Themes and Lessons for Data Mining. *Data Mining and Knowledge Discovery*, 1:11–28, 1997.

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

[5] U. Fayyad, D. Haussler and P. Stolorz. Mining Scientific Data. *Communications of the ACM*, 39(11):51–57, 1996.

[6] N. Bouguila, D. Ziou and J. Vaillancourt. Unsupervised Learning of a Finite Mixture Model Based on the Dirichlet Distribution and its Application. *IEEE Transactions on Image Processing*, 13(11):1533–1543, 2004.

[7] N. Bouguila and D. Ziou. Using unsupervised learning of a finite Dirichlet mixture model to improve pattern recognition applications. *Pattern Recognition Letters*, 26(12):1916–1925, 2005.

[8] N. Bouguila and D. Ziou. Unsupervised Selection of a Finite Dirichlet Mixture Model: An MML-Based Approach. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):993–1009, 2006.

References

[9] S. Ganesaligman. Classification and Mixture Approaches to Clustering via Maximum Likelihood. *Applied Statistics*, 38(3):455–466, 1989.

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

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

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

[13] J. Rissanen. Modeling by Shortest Data Description. *Automatica*, 14:465–471, 1987.

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

[15] H. Yassaee. Inverted Dirichlet Distribution and Multivariate Logistic Distribution. *The Canadian Journal of Statistics*, 2(1):99–105, 1974.

[16] M. Ghorbel. On the Inverted Dirichlet Distribution. *Communications in Statistics-Theory and Methods*, 39:21–37, 2010.

[17] T. Y. Young and G. Coraluppi. Stochastic Estimation of a Mixture of Normal Density Functions Using an Information Criterion. *IEEE Transactions on Information Theory*, 16(3):258–263, 1970.

[18] F. A. Graybill. *Matrices with Applications in Statistics*. Wadsworth, California, 1983.

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

[20] M. A. T. Figueiredo and A. K. Jain. Unsupervised Learning of Finite Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):4–37, March 2002.

[21] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

References

[22] G. Box. Statistics and Quality Improvement. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 157(2):209–229, 1994.

[23] N. D. Singpurwalla and S. P. Wilson. Software Reliability Modeling. *International Statistical Review*, 62(3):289–317, 1994.

[24] Pressman, R.S. Software Engineering: A Practioners Approach. *5th edn. McGraw-Hill, New York*, 2001.

[25] Basili, V.R., Hutchens, D.H. An Empirical Study of a Syntactic Complexity Family. *IEEE Transactions on Software Engineering SE*, 9(6):664–672, 1983.

[26] Porter, A.A., Selby, R.W. Empirically guided software development using metric-based classification trees. *IEEE Software*, 7(2):4654, 1990.

[27] Khoshgoftaar, T.M., Allen, E.B. Classification of Fault-Prone Software Modules: Prior Probabilities, Costs, and Model Evaluation. *Empirical Software Engineering*, 3(3):275298, 1998.

[28] M. R. Lyu, Ed. *Handbook of Software Reliability Engineering*. IEEE Computer Society Press and McGraw-Hill Book Company, 1996.

[29] Khoshgoftaar, T.M., Allen, E.B. A Practical Classification-Rule for Software-Quality Models. *IEEE Transactions on Reliability*, 49(2):209216, 2000.

[30] Troster, J., Tian, J. Measurement and Defect Modeling for a Legacy Software System. *Annals of Software Engineering*, 1(1):95118, 1995.

[31] Briand, L.C., Basili, V.R., Hetmanski, C.J. Developing Interpretable Models with Optimized Set Reduction for Identifying High-Risk Software Components. *IEEE Transactions on Software Engineering*, 19(11):10281044, 1993.

[32] S. Boutemedjet, N. Bouguila and D. Ziou. A Hybrid Feature Extraction Selection Approach for High-Dimensional Non-Gaussian Data Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1429–1443, 2009.

References

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

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

[35] Jonathan J. Oliver , Rohan A. Baxter , Chris S. Wallace. Unsupervised Learning Using MML. *in Proc of the 13th Int. Conf. on Machine Learning (San Francisco)*, pages 364–372, 1996.