MIXTURE-BASED CLUSTERING FOR HIGH-DIMENSIONAL COUNT DATA USING MINORIZATION-MAXIMIZATION APPROACHES

Ornela Bregu

A thesis
in
The Department
of
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
Concordia University
Montréal, Québec, Canada

September 2020

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Ornela Bregu**

Entitled: **Mixture-Based Clustering for High-Dimensional Count Data Using Minorization-Maximization Approaches**

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

Master of Applied Science in Quality Systems Engineering

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

Signed by the final examining commitee:

Dr. Zachary Patterson _____ External examiner

Dr. Fereshteh Mafakheri _____ Internal examiner

Dr. Jia Yuan Yu _____ Chair

Dr. Nizar Bouguila _____ Supervisor

Approved _____
Chair of Department or Graduate Program Director

_____ 2020 _____ _____

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

Abstract

Mixture-Based Clustering for High-Dimensional Count Data Using Minorization-Maximization Approaches

Ornela Bregu

The Multinomial distribution has been widely used to model count data. To increase clustering efficiency, we use an approximation of the Fisher Scoring as a learning algorithm, which is more robust to the choice of the initial parameter values. Moreover, we consider the generalization of the multinomial model obtained by introducing the Dirichlet as prior, which is called the Dirichlet Compound Multinomial (DCM). Even though DCM can address the burstiness phenomenon of count data, the presence of Gamma function in its density function usually leads to undesired complications. In this thesis, we use two alternative representations of DCM distribution to perform clustering based on finite mixture models, where the mixture parameters are estimated using minorization-maximization algorithm. Moreover, we propose an online learning technique for unsupervised clustering based on a mixture of Neerchal- Morel distributions. While the novel mixture model is able to capture overdispersion due to a weight parameter assigned to each feature in each cluster, online learning is able to overcome the drawbacks of batch learning in such a way that the mixture's parameters can be updated instantly for any new data instances. Finally, by implementing a minimum message length model selection criterion, the weights of irrelevant mixture components are driven towards zero, which resolves the problem of knowing the number of clusters beforehand. To evaluate and compare the performance of our proposed models, we have considered five challenging real-world applications that involve high-dimensional count vectors, namely, sentiment analysis, topic detection, facial expression recognition, human action recognition and medical diagnosis. The results show that the proposed algorithms increase the clustering efficiency remarkably as compared to other benchmarks, and the best results are achieved by the models able to accommodate over-dispersed count data.

Acknowledgments

I would like to express my deep gratitude to my supervisor, Dr. Nizar Bouguila, for giving me the opportunity to be part of his research team and for providing precious guidance and ongoing support throughout the past 2 years. I am extremely proud to be your student!

I am thankful to my friend, my lab-mate and my co-supervisor, Dr. Nuha Zamzami, for helping me every step of the way. I would not be able to make it without you!

Thank you to all my lab-mates for making this journey unforgettable!

Thank you to my best friends Anisa, Ruki and Suzana, for emotionally supporting me every day and putting up with me during hard times. You make me believe in true friendship!

The most heartfelt thank you to my aunt and uncle for believing in me and supporting my dreams! I look up to you!

Cannot find words to thank my parents enough for their unconditional understanding, love and support! You give me strength to overcome every challenge!

Finally, thank you to my brothers for being my biggest supporters! I treasure you!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Encouraged by the recent advances in technology, many companies are developing solutions to tap into the potential of enormous volumes of data generated daily, in order to derive real-time business insights. As a matter of fact, extracted knowledge from data offers a competitive edge and influences the decision making process in a wide realm of activities from customer classification in marketing research [1] to disease classification in medical science [2]. Count data is becoming more and more prevalent in a wide range of applications, with datasets growing both in size and in dimension[3]. Consider, for instance, collections of text documents, images or videos, where each object can be represented by a vector of frequencies of words [4], visual words [5] or visual objects, respectively. In this context, an increasing amount of research work is dedicated to the construction of statistical models directly accounting for the discrete nature of the data [6].

Machine learning approaches are widely employed to mine collected data for a better and cleaner representation, as well as to quickly and automatically build models for analyzing patterns or predicting future trends in large and complex data sets[7]. Therefore, clustering has been widely used to discover natural structure in data by organizing observations with similar characteristics in subgroups (aka clusters), in such a way that the similarity between the observations in a subgroup as well as the dissimilarity between subgroups is maximized [8]. Nevertheless, it is known to be a challenging task, especially when dealing with count data mainly because of its high-dimensionality and sparsity nature [9, 10], burstiness [11, 12] and overdispersion [13].

1.1 Finite Mixture Models

Finite mixture models have been widely used to provide a formal framework for clustering due to their natural capacity to represent heterogeneity and address random phenomena [14, 15]. Mixture models consider data to arise from two or more underlying groups with common distributional form but different parameters [16]. Due to their flexibility, mixture models are adopted in many domains, including, but not limited to, image processing and computer vision [17, 18, 19], social networks [20], and recommendor systems [21]. However, mixture modeling also faces some essential issues, including the choice of statistical distribution and optimal number of components that best describes and represents the data [22, 23], the learning algorithm for the mixture′s parameter estimation [14] and feature selection to extract a better, more compact representation of original dataset [24, 25, 26]. In our work, we aim to provide solutions for the above mentioned challenges by building robust models for efficient clustering of high-dimensional count data.

1.2 Discrete density functions

Multinomial distribution has been extensively used for count data modeling [4]. Although the Multinomial model is widely used in the case of count data, a serious drawback of it is its Naive Bayes independency assumption, *i.e.* features are independently distributed, equivalent with documents generated by repeatedly drawing words from a fixed distribution, which is not the case for real texts. Natural texts systematically exhibit the burstiness phenomenon; if a word appears once in a document, it is much more likely to appear again. The tendency of words to appear in bursts is not limited to text and can also be observed in images with visual words [27]. Also, it is common for some features (here words or visual words) to occur only once and many more to not occur at all, resulting in having the variance of the data exceeding its mean, which is known as the overdispersion phenomenon [13]. Multinomial distribution fails to capture any of these phenomena well, as shown in [28, 29]. Consequently, many techniques have been proposed to optimize data representation for more efficient and accurate clustering [30, 31], such as log-normalizing counts to reduce the impact of burstiness on the likelihood of a document [32], or proposing

other suitable distributions, such as zero-inflated Poisson, Negative Binomial, or Negative Multinomial distributions [33, 34, 35, 36, 37].

The most successful results, however, were reached by introducing the Dirichlet distribution as a prior to the multinomial, which is the classic approach to multinomial estimation. This model is known as the Dirichlet Compound Multinomial (DCM) [29], which has led to better clustering results that are comparable to the ones obtained with multiple heuristic changes to the multinomial models. The added value of the DCM model is the additional degree of freedom, compared to the multinomial distribution, which allows it to capture the burstiness phenomenon well [29]. The fact that Dirichlet is a natural conjugate to the Multinomial, where is based on the DCM composition, brings numerous computational advantages [38]. However, Haldane [39] stated that the presence of Gamma function in DCM density function leads to undesired complications of evaluating the function and its derivatives, which can be replaced by a series of polynomials. Indeed, replacing ratios of Gamma functions by rising polynomials considerably simplifies the learning process. Besides, an alternative parametrization of DCM in terms of proportion vector and overdispersion parameter as suggested by Bailey [40] and used by Griffiths [41], raises means to better tackle both the burstiness and overdispersion phenomena of count data. In this thesis, we consider yet another distribution, Neerchal–Morel distribution (NMD), proposed by [42] and [43], able to capture the burstiness and overdispersion phenomena of count data. Here, each document or image is modeled by different multinomial mixture models, where a different weight is assigned to each word or visual word in the vocabulary. Simply put, the motivation behind NMD can be easily supported by many real world applications where the independency assumption is violated, such as a medical dataset where correlation might exist between different samples from same patients. For more detailed explanations please refer to the work of the author in [44]. As a special case of the multinomial distribution, The Neerchal–Morel distribution, has been widely used in a number of applications including text mining, linguistics, and clustering, proving its many desirable theoretical and practical properties.

1.3 Optimization Approaches

A new Approximated Fisher Scoring Algorithm (AFSA) has been proposed to estimate the parameters of the Multinomial mixture model [42]. Here, the well-known Fisher Scoring (FS) algorithm is simplified by approximating the Fisher information matrix with a complete-data information matrix, which can not only mitigate computational complexity but also boosts the clustering performance. AFSA turns out to perform better than Expectation-Maximization (EM) approach, by showing higher levels of robustness to initial values and being less affected by poor local maxima. For the parameter estimates of DCM mixture model, we use a novel Minorization-Maximization (MM) algorithm framework [45]. There are two versions of the MM principle, one for iterative minimization and another for iterative maximization, respectively, known as majorization-minimization and minorization-maximization algorithms. Here, we make use of the latter one. In contrast to its special case of well-known Expectation-Maximization algorithm [46], the construction of an MM algorithm relies upon recognizing and manipulating inequalities rather than calculating conditional expectations, turning it into a powerful tool to use when dealing with nontrivial optimization problems [47]. Several complications arise during the optimization of DCM mixture models due to the non-existence of a closed-form solution, such as (1) calculating and inverting Hessian matrix, (2) solving a linear system, (3) intertwined parameters in the gamma function, (4) violation of parameter constraints, (5) dependence on the choice of initial values, etc. MM algorithms not only can avoid these complications, but they have proven to be easy to implement, amenable to acceleration and provide remarkable numerical stability [48]. Furthermore, we propose a learning approach that is robust in terms of initialization and simultaneously deals with fitting the mixture model to the observed data and selecting the optimal number of components. The proposed approach starts with a large number of components and iteratively annihilates the weak components, and redistributes the observations, where the termination criterion is based on MML criterion.

1.4 Online Learning algorithms

In this thesis, we adapt the Neerchal-Morel distribution to an online scheme. Recently, online learning has gained increasing interest given the urgent need of making machine

learning practical for real-world data analytics applications, where data are arriving at a high velocity and large volume. In contrast to traditional batch machine learning methods, online machine learning algorithms continuously integrate new information into already constructed models instead of reconstructing new models from scratch each time new data become available. Practically, the whole data set is split into batches and the parameter estimates are updated with a new batch of data points in each iteration. Consequently, the models are all-time up to date and the costs for data storage and maintenance are reduced significantly.

1.5 Thesis overview

The rest of the thesis is organized as follows: In Chapter 2, we propose mixture models for high dimensional count data based on Multinomial and Dirichlet Compound Multinomial distributions, where the mixture's parameters are to be learned by Approximated Fisher Scoring and Minorization-Maximization optimization algorithms, respectively. In Chapter 3, we build an online mixture-based model using Neerchal-Morel distribution, to capture the extra-variation of the count data in real-time applications. Finally, in Chapter 4 we summarize the experimental results and our major contributions in this thesis.

Chapter 2

Mixture-Based Clustering for Count Data Using Approximated Fisher Scoring and Minorization-Maximization Approaches

In this chapter, we use the Multinomial mixture model and introduce the Approximate Fisher Scoring algorithm for the estimation of mixture's parameters. Then, we compare it with the Dirichlet Compound Multinomial mixture model and two other alternative forms of DCM density function. Here, the learning of the mixture's parameters is achieved by using a minorization-maximization framework. Moreover, we have integrated the minimum message length criterion to our model to select the optimal number of components. Finally, we demonstrate the experimental results.

2.1 The Multinomial Mixture Model

Let $\mathcal{X} = \{X_1, \ldots, X_N\}$ be a set of N independently and identically distributed documents or images, where each can be represented as a sparse D dimensional vector of cell counts $X_i = (x_{i1}, \ldots, x_{iD})$, assumed to follow a Multinomial distribution, given

by:

$$\mathcal{M}(X_i|p) = \frac{m_i!}{\prod_{d=1}^{D} x_{id}!} \prod_{d=1}^{D} p_d^{x_{id}} \tag{1}$$

where D is the vocabulary size; $m_i = \sum_{d=1}^{D} x_{id}$ represents the length of the document; p_d is the probability of emitting a word d which is subject to the constraints $p_d > 0$ and $\sum_{d=1}^{D} p_d = 1$.

Then, a finite mixture model of K multinomial distributions is denoted as follows:

$$P(X_i|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{M}(X_i|p_k) \tag{2}$$

where π_k are the mixing weights, which must satisfy the condition $\sum_{k=1}^{K} \pi_k = 1$, $K \geqslant 1$ is number of components in the mixture, p_k are the parameters defining the kth component; and $\Theta = \{p_1, \ldots, p_K, \pi_1, \ldots, \pi_K\}$ is the set of all latent variables.

Finally, we can write the data log-likelihood for the whole dataset $\mathcal{X} = \{X_1, \ldots, X_N\}$ in the given form:

$$\mathcal{L}(\mathcal{X}|\Theta) = \prod_{i=1}^{N} \sum_{k=1}^{K} \log \left(\pi_k \mathcal{M}(X_i|p_k) \right) \tag{3}$$

In order to learn the finite mixture model, we seek to maximize the log-likelihood function $\mathcal{L}(\mathcal{X}|\Theta)$ with respect to the parameters Θ. However, the inner summation of the mixture models prevents maximum likelihood (ML) estimates to be obtained analytically. Hence, different methods, such as Fisher Scoring (FS), Expectation-Maximization or Minorization-Maximization can be used to obtain the ML estimates numerically.

To learn the parameters of the multinomial mixture model we use a novel numerical approach, the approximate Fisher Scoring algorithm [42, 43, 44], which is an approximation of the well-known Fisher Scoring method, as the name indicates. Through Fisher Scoring method the ML estimates can be found by iteratively computing Eq. (4) until the convergence criteria $|\mathcal{L}(\mathcal{X}|\Theta^{(t+1)}) - \mathcal{L}(\mathcal{X}|\Theta^{(t)})| < \epsilon$ is met for a given threshold $\epsilon > 0$.

$$\Theta^{(t+1)} = \Theta^{(t)} + F\left(\Theta^{(t)}\right)^{-1} S\left(\Theta^{(t)}\right), \quad t = 1, 2, \ldots \tag{4}$$

where $F\left(\Theta^{(t)}\right)$ is the Fisher Information Matrix (FIM), equivalent to the negative expected Hessian of the log-likelihood and $S\left(\Theta^{(t)}\right)$ is the scoring function, equivalent to the first derivative of the log-likelihood function. The calculation of the exact

FIM becomes computationally expensive or even intractable when working with high dimensional and/or huge vocabulary size data sets. Therefore, we employ the fact[49] that the FIM can be approximated by the block-diagonal matrix

$$\widetilde{F}(\Theta) = Blockdiag(\pi_1 F_1 + \ldots + \pi_K F_K, F_\pi) \tag{5}$$

where $F_k = M[D_k^{-1} + p_{kd}^{-1}\not{1}\not{1}^T]$ and $D_k = Diag(p_{k1}, \ldots, p_{k,D-1})$; $F_\pi = N[D_\pi^{-1} + \pi_k^{-1}\not{1}\not{1}^T]$ and $D_\pi = Diag(\pi_1, \ldots, p_{K-1})$; $\not{1}$ stands for the identity matrix and $M = \sum_{i=1}^{N} m_i$. Since the approximated Fisher Information Matrix (AFIM) is a block-diagonal matrix, its inverse and determinant can be obtained in closed-form, as:

$$\widetilde{F}(\Theta)^{-1} = Blockdiag\left(\pi_1^{-1}F_1^{-1}, \ldots, \pi_K^{-1}F_K^{-1}, F_\pi^{-1}\right) \tag{6}$$

$$\det\left(\widetilde{F}(\Theta)\right) = \left(\prod_{k=1}^{K} p_{kd}^{-1} \prod_{d=1}^{D-1} M\pi_k p_{kd}^{-1}\right)\left(\pi_K^{-1} \prod_{k=1}^{K-1} N\pi_k^{-1}\right) \tag{7}$$

where $F_k^{-1} = M^{-1}\left\{D_k - p_k p_k^T\right\}$ for $k = 1, \ldots, K$ and $F_\pi^{-1} = N^{-1}\left\{D_\pi - \pi\pi^T\right\}$. The determinant of the AFIM will be very useful shortly when we integrate model selection in our mixture model. Moreover, the approximated FIM is equivalent to the exact FIM of the complete-data log-likelihood, which makes the approach closely related to the EM and can be used on other finite mixture models and missing data problems. To conclude, AFSA consists on iteratively computing Eq. (8) until convergence.

$$\Theta^{(t+1)} = \Theta^{(t)} + \widetilde{F}\left(\Theta^{(t)}\right)^{-1} S\left(\Theta^{(t)}\right), \quad t = 1, 2, \ldots \tag{8}$$

By separating individual updates and further simplifying (please refer to [49] for details) we obtain the simple formulas for the weights π_k and the multinomial distribution parameter p_{kd}, as:

$$\pi_k^{(t+1)} = \pi_k^{(t)} \frac{1}{N} \sum_{i=1}^{N} \frac{\mathcal{M}(X_i|{p_k}^{(t)})}{\sum_{j=1}^{K} \pi_j^{(t)} \mathcal{M}(X_i|{p_j}^{(t)})} \tag{9}$$

$$p_{kd}^{(t+1)} = \frac{1}{M} \sum_{i=1}^{N} x_{id} \frac{\mathcal{M}(X_i|{p_k}^{(t)})}{\sum_{j=1}^{K} \pi_j^{(t)} \mathcal{M}(X_i|{p_j}^{(t)})} - p_{kd}^{(t)} \left[1 - \frac{1}{M} \sum_{i=1}^{N} m_i \frac{\mathcal{M}(X_i|{p_k}^{(t)})}{\sum_{j=1}^{K} \pi_j^{(t)} \mathcal{M}(X_i|{p_j}^{(t)})}\right] \tag{10}$$

The simplicity of the new algorithm brings numerous advantages, especially in computation demands, for the unsupervised clustering of high-dimensional count data.

However, several limitations and technical problems associated with the multinomial independency assumption have been discussed in the literature [29, 32], especially when dealing with datasets prone to burstiness or overdispersion phenomena. An appropriate and efficient solution to address this issue is the hierarchical Bayesian modeling approach that introduces the prior information into the construction of the statistical model.

2.2 The Dirichlet Compound Multinomial (DCM) Mixture Model

Authors in [29] used the fact that the Dirichlet distribution is a natural conjugate to the Multinomial distribution to propose the DCM distribution, which can not only accommodate burstiness phenomenon but shows potential to outperform the multinomial distribution with all heuristics applied.

2.2.1 DCM Distribution

Relying upon hierarchical Bayesian modeling, the Dirichlet distribution carries prior information for the multinomial parameter, which is then used to model the document or image. Consequently, different documents or images in the same class are modeled by different multinomial distributions, so the words or visual words that appear once will have a higher probability of appearing again as represented by the multinomial parameter.

Let $\mathcal{X} = \{X_1, \ldots, X_N\}$ be a set of N independently and identically distributed documents or images, where each can be represented as a sparse D dimensional vector of cell counts $X_i = (x_{i1}, \ldots, x_{iD})$, assumed to follow a Dirichlet Compound Multinomial (DCM) distribution, given by:

$$\begin{aligned}
\mathcal{DCM}(X_i|\alpha) &= \int_p \mathcal{M}(X_i|p)\mathcal{D}(p|\alpha)dp \\
&= \int_p \left(\frac{m_i!}{\prod_{d=1}^{D} x_{id}!} \prod_{d=1}^{D} p_d^{x_{id}} \right) \left(\frac{\Gamma\left(\sum_{d=1}^{D} \alpha_d\right)}{\prod_{d=1}^{D} \Gamma\left(\alpha_d\right)} \prod_{d=1}^{D} p_d^{\alpha_d - 1} \right) dp \qquad (11) \\
&= \frac{m_i!}{\prod_{d=1}^{D} x_{id}!} \frac{\Gamma(|\alpha|)}{\Gamma(m_i + |\alpha|)} \prod_{d=1}^{D} \frac{\Gamma(x_{id} + \alpha_d)}{\Gamma(\alpha_d)}
\end{aligned}$$

where $\mathcal{M}(X_i|p)$ and $\mathcal{D}(p|\alpha)$ are the Multinomial and the Dirichlet distribution, respectively; D is the vocabulary size; p_d is the probability of emitting a word d; $m_i = \sum_{d=1}^{D} x_{id}$ represents the length of the document; α_d is the positive Dirichlet distribution parameter for each word, and $|\alpha| = \sum_{d=1}^{D} \alpha_d$ is the sum of parameters. From a practical point of view, each Dirichlet represents a general topic that compounds a set of documents whether each Multinomial is linked to specific sub-topics and gives higher probability to some words to appear than others, for a specific document. This model can also be interpreted as bag-of-bags-of words or bag-of-scaled documents [29]. Note that DCM has one extra degree of freedom, as compared to the multinomial, since its parameters do not bear the unit-sum constraint, which accommodates burstiness and makes it more practical [28, 32].

2.2.2 Alternative DCM parametrizations

Although the DCM composition has led to better clustering results, Haldane (1941) [39] stated that the occurrence of Gamma function in DCM density function leads to undesired complications of evaluating the function and its derivatives which can be replaced by a series of polynomials. The alternative representation of the density function is given by:

$$DCM(X_i|\alpha) = \frac{m_i!}{\prod_{d=1}^{D} x_{id}!} \frac{\prod_{d=1}^{D} \alpha_d(\alpha_d+1)\dots(\alpha_d+x_{id}-1)}{|\alpha|(|\alpha|+1)\dots(|\alpha|+m_i-1)} \tag{12}$$

Moreover, an alternative parametrization in terms of proportion vector $\beta = (\beta_1, \dots, \beta_D)$ and overdispersion parameter θ, proposed by Bailey(1957)[40], raises means to better tackle the burstiness and overdispersion phenomena of count data [41]. It is given by:

$$DCM(X_i|\beta,\theta) = \frac{m_i!}{\prod_{d=1}^{D} x_{id}!} \frac{\prod_{d=1}^{D} \beta_d(\beta_d+\theta)\dots[\beta_d+(x_{id}-1))\theta]}{(1+\theta)\dots[1+(m_i-1)\theta]} \tag{13}$$

where $\beta_d = \frac{\alpha_d}{|\alpha|}$, $\sum_d \beta_d = 1$ and $\theta = \frac{1}{|\alpha|}$. Indeed, replacing ratios of Gamma functions by rising polynomials will considerably simplify the learning process.

2.2.3 DCM Mixture Learning Approaches

The Expectation Maximization Algorithm

The Expectation Maximization algorithm is one of the most familiar approaches used to find maximum likelihood solutions for probabilistic models with missing data. A membership vector $Z_i = (z_{i1}, \ldots, z_{iK})$ is assigned to each observation X_i such that $z_{ik} = 1$ if the object i belongs in the cluster k and all other elements equal to 0. Therefore, the membership vector is a K-dimensional binary random variable whose values must satisfy the conditions $z_{ik} \in \{0, 1\}$ and $\sum_{k=1}^{K} z_{ik} = 1$. The conditional distribution of latent variable $\mathcal{Z}$, given the mixing coefficients π, can be written as:

$$P(\mathcal{Z}|\pi) \sim Multi(\pi) = \prod_{i=1}^{N}\prod_{k=1}^{K} \pi_k^{z_{ik}} \tag{14}$$

Similarly, from Eq.(3) we can write the conditional distribution of data vectors $\mathcal{X}$, given the latent variables $\mathcal{Z}$ and the component parameters Θ. Thus, we can rewrite the complete data likelihood in the given form:

$$P(\mathcal{X}, \mathcal{Z}|\Theta) = \prod_{i=1}^{N}\prod_{k=1}^{K} \Big(\pi_k DCM(X_i|\alpha_k)\Big)^{z_{ik}} \tag{15}$$

Based on Eq.(11) and Eq.(15), the complete data log-likelihood corresponding to a K-component mixture of DCM distributions is given by:

$$\begin{aligned}
\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta) &= \sum_{i=1}^{N}\sum_{k-1}^{K} z_{ik} \left(\log \pi_k + \log DCM(X_i|\alpha_k)\right) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik} log \pi_k + \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik} \left(\log \Gamma\left(|\alpha|\right) - \log \Gamma\left(m_i + |\alpha|\right)\right) \\
&+ \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik} \sum_{d=1}^{D} \left(\log \Gamma\left(x_{id} + \alpha_d\right) - \log \Gamma\left(\alpha_d\right)\right)
\end{aligned} \tag{16}$$

The optimization of the complete-data likelihood function $\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)$ is significantly easier than the direct optimization of the likelihood function $\mathcal{L}(\mathcal{X}|\Theta)$. In EM, the learning of the parameters of a mixture model is done by a two-step iteration (Expectation-step and Maximization-step). In the E-step, the posterior probabilities of the missing variables $P(\mathcal{Z}|\mathcal{X}, \Theta^{(t)})$ are evaluated using the current values of the parameters, as:

$$\hat{z}_{ik}^{(t)} = P(\mathcal{Z}|\mathcal{X}, \Theta^{(t)}) = \frac{\pi_k^{(t)} DCM\left(X_i|\alpha_k^{(t)}\right)}{\sum_{j=1}^{K} \pi_j^{(t)} DCM\left(X_i|\alpha_j^{(t)}\right)} \tag{17}$$

Then, in the M-step, the expectation of the complete-data log likelihood with respect to the missing variables is maximized. The motivation behind this solution relies on the nonnegativity of the Kullback-Leibler divergence of two conditional probability densities. The divergence inequality in turn depends on Jensen's inequality and the concavity of the function ln x. [50] It can be shown that estimates on each iteration increase the log-likelihood function on the data [51]. The parameter estimates and the weights get updated using the posterior probabilities calculated in the previous step according to :

$$\Theta^{(t+1)} = \arg\max_{\Theta} \mathcal{Q}\left(\Theta|\Theta^{(t)}\right) \tag{18}$$

where

$$\mathcal{Q}\left(\Theta|\Theta^{(t)}\right) = \sum_{\mathcal{Z}} \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta) P(\mathcal{Z}|\mathcal{X}, \Theta^{(t)}) \tag{19}$$

By setting the derivative of the log-likelihood function equal to zero, we obtain:

$$\pi_k^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} \hat{z}_{ik}^{(t)} \tag{20}$$

Here, we do not obtain a closed-form solution for the α_k parameters since they are intertwined in the $log\Gamma(|\alpha|)$ term. We therefore use the Newton-Raphson method, where the parameter estimates iterate according to:

$$\alpha_{kd}^{(t+1)} = \alpha_{kd}^{(t)} - \left(\frac{\partial^2 \mathcal{Q}\left(\Theta|\Theta^{(t)}\right)}{\partial \alpha_{kd} \alpha_{kd}} \right)^{-1} \frac{\partial \mathcal{Q}\left(\Theta|\Theta^{(t)}\right)}{\partial \alpha_{kd}} \tag{21}$$

where:

$$\frac{\partial \mathcal{Q}\left(\Theta|\Theta^{(t)}\right)}{\partial \alpha_{kd}} = \sum_i \hat{z}_{ik} \left[\Psi\left(x_{id} + \alpha_{kd}^{(t)}\right) - \Psi\left(m_i + \left|\alpha_k^{(t)}\right|\right) + \Psi(|\alpha_k^{(t)}|) - \Psi\left(\alpha_{kd}^{(t)}\right) \right] \tag{22}$$

$$\begin{aligned} \frac{\partial^2 \mathcal{Q}\left(\Theta|\Theta^{(t)}\right)}{\partial \alpha_{kd} \alpha_{kd}} = \sum_i \hat{z}_{ik} & \left[\left(\Psi'\left(x_{id} + \alpha_{kd}^{(t)}\right) - \Psi'\left(\alpha_{kd}^{(t)}\right) \right) 1_{\{d=d'\}} \right] \\ & - \left[\Psi'\left(m_i + \left|\alpha_k^{(t)}\right|\right) + \Psi'(|\alpha_k^{(t)}|) \right] \end{aligned} \tag{23}$$

where $\Psi(z)$ and $\Psi'(z)$ are the digamma and trigamma functions, respectively. Calculating and inverting the Hessian matrix at each iteration is computationally expensive. Moreover, the EM approach has several other drawbacks, such as violation of parameter's constraints and dependence on the choice of initial values. In the next section

we introduce the MM framework, as a general case of the EM approach which not only can avoid these complications, but is also easy to implement.

The Minorization-Maximization (MM) Algorithm

The key to the construction of an MM algorithm for calculating MLE of the model parameters is to carefully choose an appropriate surrogate function minorizing the log-likelihood function, which must satisfy two properties, mathematically written as:

$$\begin{aligned} \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta^{(t)}) &= \mathcal{G}(\Theta^{(t)}|\Theta^{(t)}), \\ \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta) &\geqslant \mathcal{G}(\Theta|\Theta^{(t)}), \Theta \neq \Theta^{(t)} \end{aligned} \tag{24}$$

In other words, the surface of the surrogate function lies below the surface of the objective function and they are tangent at the point $\Theta = \Theta^{(t)}$, where $\Theta^{(t)}$ represents the current iterate. Given the definition of $\Theta^{(t)}$ and Eq.(24), one can prove that if the surrogate function reaches its maximum value for $\Theta^{(t+1)}$, then MM procedure drives the likelihood uphill. This is also known as the ascent property and is based on the inequalities:

$$\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta^{(t+1)}) \geqslant \mathcal{G}(\Theta^{(t+1)}|\Theta^{(t)}) \geqslant \mathcal{G}(\Theta^{(t)}|\Theta^{(t)}) \equiv \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta^{(t)}) \tag{25}$$

The ascent property holds true even if $\mathcal{G}(\Theta|\Theta^{(t)})$ is increased rather than maximized, leading to significant levels of numerical stability and proving to be exceptionally beneficial in case the maximum of the surrogate function can not be found. Therefore, the surrogate function is maximized during the second step of the MM algorithm in order to produce the next iterate Θ^{t+1}.

As we have already emphasized, MM relies on recognizing and manipulating inequalities after close examination of the log-likelihood. In this work, we strategically minorize parts of the overall objective function while leaving the other parts untouched. Thus, to construct an MM algorithm under the parametrization in Eq.(12), we need to minorize terms such as $\ln(\alpha_{kd} + l)$ for $l = (0, \ldots, x_{id})$ and $-\ln(|\alpha_k| + l)$ for $l = (0, \ldots, m_i)$, making use of Jensen inequality and the supporting hyperplane property. Similarly, under parametrization in Eq.(13), we minorize terms such as $-\ln(1 + k\theta)$ and $\ln(\pi_j + k\theta)$, once again making use of the previously mentioned inequalities.

As previously stated, we aim to minorize only parts of the overall objective function. Under parametrization in Eq.(12), the complete log-likelihood is written as:

$$\begin{aligned}\mathcal{L}(\mathcal{X},\mathcal{Z}|\Theta) &= \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik} log\mathcal{DCM}(X_{id}|\alpha_{kd}) \\ &= \sum_{k=1}^{K}\sum_{i=1}^{N} z_{ik}\sum_{d=1}^{D}\sum_{l=0}^{x_{id}-1}\ln(\alpha_{kd}+l) - \sum_{k=1}^{K}\sum_{i=1}^{N} z_{ik}\sum_{l=0}^{m_i-1}(\ln(|\alpha_k|+l)\end{aligned} \tag{26}$$

For computational benefits, we simplify the two terms further using the inequalities proposed in [52]. By close observation of the function, one can tell that the terms $\ln(\alpha_{kd}+l)$ and $-\ln(|\alpha_k|+l)$ occur in the data log likelihood only when the conditions $x_{id} \geqslant l+1$ and $m_i \geqslant l+1$ are satisfied, respectively. Therefore,

$$\begin{aligned}\sum_{k=1}^{K}\sum_{i=1}^{N}\hat{z}_{ik}\sum_{d=1}^{D}\sum_{l=0}^{x_{id}-1}\ln(\alpha_{kd}+l) &= \sum_{k=1}^{K}\sum_{d=1}^{D}\sum_{l=0}^{\max_i x_{id}-1}\ln(\alpha_{kd}+l)\sum_{i=1}^{N}\hat{z}_{ik(x_{id}-1\geqslant l)} \\ &= \sum_{k=1}^{K}\sum_{d=1}^{D}\sum_{l=0}^{\max_i x_{id}-1} S_{dlk}\ln(\alpha_{kd}+l)\end{aligned} \tag{27}$$

and

$$\begin{aligned}-\sum_{k=1}^{K}\sum_{i=1}^{N}\hat{z}_{ik}\sum_{l=0}^{m_i-1}(\ln(|\alpha_k|+l) &= -\sum_{k=1}^{K}\sum_{l=0}^{\max_i m_i-1}\ln(|\alpha_k|+l)\sum_{i=1}^{N}\hat{z}_{ik(m_i-1\geqslant l)} \\ &= -\sum_{k=1}^{K}\sum_{l=0}^{\max_i m_i-1} N_{lk}\ln(|\alpha_k|+l)\end{aligned} \tag{28}$$

where $N_{lk} = \sum_{i=1}^{N}\hat{z}_{ik(m_i-1\geqslant l)}$ represents the sum of responsibilities the of data points x_{id} where the batch size is bigger than the variable l. The batch size is the total number of the frequencies for each document or image. On the other hand, $S_{dlk} = \sum_{i=1}^{N}\hat{z}_{ik(x_{id}-1\geqslant l)}$ is the sum of the responsibilities of data points x_{id} with d-th coord bigger than the variable l. Applying the Jensen inequality to the $ln(\alpha_{kd}+l)$ terms

$$\begin{aligned}\ln(\alpha_{kd}+l) &\geq \frac{\alpha_{kd}^{(t)}}{\alpha_{kd}^{(t)}+l}\ln\left(\frac{\alpha_{kd}^{(t)}+l}{\alpha_{kd}^{(t)}}\cdot\alpha_{kd}\right) + \frac{l}{\alpha_{kd}^{(t)}+l}\ln\left(\frac{\alpha_{kd}^{(t)}+l}{l}\cdot l\right) \\ &= \frac{\alpha_{kd}^{(t)}}{\alpha_{kd}^{(t)}+l}\ln\alpha_{kd} + c^{(t)}\end{aligned} \tag{29}$$

and the supporting hyperplane inequality to the $-ln(|\alpha_k|+l)$ terms

$$-\ln(|\alpha_k|+l) \geq -\frac{|\alpha_k| - \left|\alpha_k{}^{(t)}\right|}{|\alpha_k{}^{(t)}|+l} - \ln\left(\left|\alpha_k{}^{(t)}\right|+l\right) = -\frac{|\alpha|}{|\alpha_k{}^{(t)}|+l} + c^{(t)} \tag{30}$$

yields the surrogate function:

$$\mathcal{G}\left(\alpha|\alpha^{(t)}\right) = -\sum_{k}\sum_{l}\frac{N_{lk}}{|\alpha_k{}^{(t)}|+l}|\alpha_k| + \sum_{k}\sum_{d}\sum_{l}\frac{S_{dlk}\alpha_{dk}^{(t)}}{\alpha_{dk}^{(t)}+l}\ln\alpha_{dk} + c^{(t)} \tag{31}$$

The resulting surrogate function is less than or equal the complete log-likelihood function and can be solved analytically in the maximization step. By maximizing the surrogate function, we obtain the update for the α_{kd} parameter:

$$\alpha_{kd}^{(t+1)} = \left(\sum_{l}\frac{\alpha_{kd}^{(t)}S_{dlk}}{\alpha_{kd}^{(t)}+l}\right) / \left(\sum_{l}\frac{N_{lk}}{|\alpha_k^{(t)}|+l}\right) \tag{32}$$

Under the parametrization in Eq.(13), the complete log-likelihood is written as:

$$\begin{aligned}\mathcal{L}(\mathcal{X},\mathcal{Z}|\Theta) &= \sum_{i=1}^{N}\sum_{k=1}^{K}\hat{z}_{ik}log\mathcal{DCM}(X_{id}|\beta_{kd},\theta_k)\\ &= \sum_{k=1}^{K}\sum_{i=1}^{N}\hat{z}_{ik}\sum_{d=1}^{D}\sum_{l=0}^{x_{id}-1}\ln(\beta_{kd}+l\theta_k) - \sum_{k=1}^{K}\sum_{i=1}^{N}\hat{z}_{ik}\sum_{l=0}^{m_i-1}\ln(1+l\theta_k)\end{aligned} \tag{33}$$

Following the same approach, we constructed a surrogate function for the second parametrization. Following the same logic, we simplify the terms:

$$\begin{aligned}\sum_{k=1}^{K}\sum_{i=1}^{N}\hat{z}_{ik}\sum_{d=1}^{D}\sum_{l=0}^{x_{id}-1}\ln(\beta_{kd}+l\theta_k) &= \sum_{k=1}^{K}\sum_{d=1}^{D}\sum_{l=0}^{\max_i x_{id}-1}\ln(\beta_{kd}+l\theta_k)\sum_{i=1}^{N}\hat{z}_{ik(x_{id}-1\geqslant l)}\\ &= \sum_{k=1}^{K}\sum_{d=1}^{D}\sum_{l=0}^{\max_i x_{id}-1}S_{dlk}\ln(\beta_{kd}+l\theta_k)\end{aligned} \tag{34}$$

and

$$\begin{aligned}-\sum_{k=1}^{K}\sum_{i=1}^{N}\hat{z}_{ik}\sum_{l=0}^{m_i-1}\ln(1+l\theta_k) &= -\sum_{k=1}^{K}\sum_{l=0}^{\max_i m_i-1}\ln(1+l\theta_k)\sum_{i=1}^{N}\hat{z}_{ik(m_i-1\geqslant l)}\\ &= -\sum_{k=1}^{K}\sum_{l=0}^{\max_i m_i-1}N_{lk}\ln(1+l\theta_k)\end{aligned} \tag{35}$$

Then we minorize the terms such as $-\ln(1+k\theta)$ and $\ln(\pi_j+k\theta)$, once again making use of the Jensen inequality

$$\begin{aligned}\log(\beta_{kd}+k\theta_k) &\geq \frac{\beta_{kd}^n}{\beta_{kd}^n+l\theta_k^n}\log\left(\frac{\beta_{kd}^n+l\theta_k^n}{\beta_{kd}^n}\beta_{kd}\right) + \frac{l\theta_k^n}{\beta_{kd}^n+l\theta_k^n}\log\left(\frac{\beta_{kd}^n+l\theta_k^n}{l\theta_k^n}l\theta_k\right)\\ &= \frac{\beta_{kd}^n}{\beta_{kd}^n+l\theta_k^n}\log\beta_{kd} + \frac{l\theta_k^n}{\beta_{kd}^n+l\theta_k^n}\log\theta_k\end{aligned} \tag{36}$$

and the supporting hyperplane property

$$-\log(1+k\theta_k) \geq -\log\left(1+l\theta_k^n\right) - \frac{1}{1+l\theta_k^n}\left(l\theta_k - l\theta_k^n\right) = \frac{l\theta_k}{1+l\theta_k^n} \tag{37}$$

which lead to the surrogate function

$$-\sum_k \sum_l N_{kl} \frac{l}{1+l\theta_k^n}\theta_k + \sum_k \sum_d \sum_l S_{dlk} \left\{ \frac{\beta_{kd}^n}{\beta_{kd}^n + l\theta_k^n} \log \beta_{kd} + \frac{l\theta_k^n}{\beta_{kd}^n + l\theta_k^n} \log \theta \right\} \tag{38}$$

Setting the partial derivatives of the surrogate function (Eq.38) with respect to θ_k and β_{kd} equal to 0 yields the MM updates:

$$\theta_k^{n+1} = \left(\sum_d \sum_l \frac{S_{dlk} l\theta_k^n}{\beta_{kd}^n + l\theta_k^n} \right) / \left(\sum_l \frac{N_{lk} l}{1+l\theta_k^n} \right) \tag{39}$$

$$\beta_{kd}^{n+1} = \left(\sum_l \frac{S_{dlk}\beta_{kd}^n}{\beta_{kd}^n + l\theta_k^n} \right) / \left(\sum_j \sum_l \frac{S_{jlk}\beta_{jk}^n}{\beta_{jk}^n + l\theta_k^n} \right) \tag{40}$$

where the constraint $\sum_d \beta_{kd} = 1$ must be satisfied and therefore finding the proportion parameter update has been treated as a Langrange multiplier problem.

2.3 Model selection

Previously, we assumed the number of components in the mixture was known beforehand, which is not the case in unsupervised learning. Different approaches have been proposed in the literature to deal with the unknown number of components. Deterministic approaches such as Minimum Message Length (MML) [14, 53] Akaike's Information Criterion (AIC) [54], Minimum Description Length (MDL) [55] or Mixture MDL (MMDL) [14] are widely used since they appear to be less computationally demanding, as compared to other stochastic approaches [56, 57].

MML has proven to be efficient with mixture models [58, 59]. Therefore, we choose it to find the optimal number of components that best describe and represent the data. Minimum Message Length selection criterion consists of evaluating the statistical model's ability to compress a message containing the data. MML's philosophy, based on information theory, states that the best statistical model has the ability to achieve a high compression of its data [60]. The message includes two parts where the first part encodes the model using only prior information about the model and no

information about the data, and the second part encodes only the data. Finally, the optimal number of clusters K is the candidate value which minimizes the message length, given by:

$$\hat{\Theta} = \arg\min_{\Theta} \left\{ -\log(h(\Theta)) - \log(P(\mathcal{X}|\Theta)) + \frac{1}{2}\log|F(\Theta)| + \frac{N_p}{2}\left(1 + \log\frac{N_p}{12}\right)\right\} \tag{41}$$

where $h(\Theta)$ is the prior probability, $P(\mathcal{X}|\Theta)$ is the likelihood for the complete data set, $|F(\Theta)|$ is the determinant of the expected Fisher information matrix and N_p is the number of free parameters to be estimated, which is $KD-1$ and $K(D+1)-1$ for the Multinomial and DCM mixture models, respectively.

The capability of the MML criterion is directly dependent on the choice of prior distribution $h(\Theta)$ for the parameters of the mixture models. By assuming that the parameters p_d of the different components as a prior are independent from the mixing probabilities π, and the components of $h(p_d)$ are independent as well [61], we get:

$$h(\Theta) = h(\pi)\prod_{k=1}^{K} h\left(p_k\right) = h(\pi)\prod_{k=1}^{K}\prod_{d=1}^{D} h\left(p_{kd}\right) \tag{42}$$

Considering that both parameters π, p belong on the the probability simplex: $\{(\pi_1, \ldots, \pi_K) : \sum_{k=1}^{K}\pi_k = 1, \pi_k > 0 \text{ for } k = 1, \ldots, K\}$ and $\{(p_{k1}, \ldots, p_{KD}) : \sum_{d=1}^{D} p_{kd} = 1, p_{kd} > 0 \text{ for } d = 1, \ldots, D\}$, the Dirichlet distribution becomes a natural choice as a prior. The choice of a flat Dirichlet distribution (all parameters equal to 1) gives a uniform prior as follows [22, 23]:

$$h(\pi) = \Gamma(K) = (K-1)! \tag{43}$$

$$h(p) = \Gamma(D) = (D-1)! \tag{44}$$

Thus, substituting (43) and (44) into (42), and taking the log we obtain:

$$log(h(\Theta)) = log\Gamma(K) + KDlog\Gamma(D) \tag{45}$$

As a result of the approximation we introduced in the previous section, we are able to compute the determinant of the AFIM after the data vectors have been assigned to their respective clusters, as in Eq (7). Thus, by substituting (45) and (7) into (41), we obtain the expression of MML for a finite mixture of Multinomial distributions,

given a candidate value for K.

On the other hand, the information matrix for the DCM model is difficult to obtain analytically. Even though by using MM algorithms, we solve many of the weaknesses of the EM algorithm for mixture models, the requirement of knowing the number of clusters during the initialization step still remains. To overcome this challenge, we adopt the approach in [24], which is based on the MML criterion [23], where we avoid the problem that might emerge with running the EM algorithm multiple times to obtain the whole set of candidates. Instead, with each iteration we will run the component-wise EM until convergence, where the irrelevant components, with $\hat{\pi}_k^{(t+1)} = 0$ are annihilated, and the parameters are updated accordingly. Then, the MML criterion is re-evaluated for non-zero components only.

We perform similar transformations as in [24] (please refer for more details), and gain the final form:

$$\hat{\Theta} = \arg\max_{\Theta} \left\{ \log p(\mathcal{X}|\Theta) - \frac{RD}{2} \sum_{k=1}^{K} \log \pi_k - \frac{S}{2} \sum_{d=1}^{D} \log\left(1 - \rho_d\right) - \frac{RK}{2} \sum_{d=1}^{D} \log \rho_d \right\} \tag{46}$$

where R and S are the number of parameters in the probability densities. Thus, $R = S = 1$ for the first DCM parametrization in Eq.(12) and $R = S = 2$ for the second one based on Eq.(13).

The model can be initialized with a large value of K, thus surpassing the limitation of initialization dependency. Starting with a large value of K may lead to several empty components and there will be no need to estimate, and transmit, their parameters. Thus, we need to update the component's weight in the M-step as:

$$\widehat{\pi}_k = \frac{\max\left(\sum_i \hat{z}_{ik} - \frac{RD}{2}, 0\right)}{\sum_j \max\left(\sum_i \hat{z}_{ij} - \frac{RD}{2}, 0\right)} \tag{47}$$

The advantage of the new update formula is its pruning behavior, that when some of the π_k go to zero they will be removed.

2.4 Experimental Results

In this section, we aim to prove the effectiveness of our proposed models via three real-world applications; sentiment analysis, facial expression recognition, and human action recognition. The experiments aim to compare the accuracy of the proposed

mixture models based on two different parameterizations of DCM, to the original parameterization of DCM as in [29] and to the approximate Fisher Scoring algorithm [49] with multinomial mixture model serving as a baseline . We evaluate the clustering performance of the different models across different data sets.

2.4.1 Sentiment Analysis

Sentiment Analysis is the process of determining the attitude of a subject towards a particular topic by a given text written in natural languages. Typically, it has been used in the past to classify the opinions in product or movie reviews. Therefore, we have chosen two widely used datasets in the past, namely IMBD dataset for movie reviews and the Amazon dataset for product reviews [62].

IMDB movie reviews dataset categorizes the reviews into positive and negative sentiments. Ratings on IMDB are given as star values $\in \{1, 2, \ldots, 10\}$, Following previous work on polarity classification, we have considered only highly polarized reviews where a review is assigned label 0 (negative) if its score is less than 5, and label 1 (positive) if its score is 6, or greater. Here, we used a mix of the training and testing sets having around 25,000 samples from each positive/negative group with a vocabulary size of $76,340$ unique words. **The Amazon reviews full score dataset** contains 600,000 training samples and 130,000 testing samples for each review score from 1 to 5. Similarly to IMDB preprocessing step, they were linearly mapped to $[0, 1]$ to use as document labels, negative and positive, respectively. Here, we used a mix of 50,000 sample reviews from the training and testing sets, with a vocabulary size of 55,383 unique words.

Note that we do not separate the data set into training and testing sets. The Rainbow package [63] was used to read the text files and perform the feature selection considering words with the highest average mutual information after removing all rare words (less than 50 occurrences in our experiments). Since for sentiment analysis, certain stop words (e.g., negating words such as no, not, and never) are indicative; traditional stop word removal was not used. Each text file is then represented as a vector containing the occurrence frequency for each word from the vocabulary.

Table 1: Average accuracy (in %), and estimated number of components of DCM algorithms over different runs for sentiment analysis.

	MN	**MN-AFSA**	**DCM**	**DCM1**	**DCM2**	$\hat{K}$	K
Amazon	50.83	65.28	62.66	68.23	75.76	2.2	2
IMDB	64.18	78.93	82.19	82.35	82.85	2.0	2

Table 2: Average time (in seconds) of algorithms over different runs for IMDB dataset.

	MN-AFSA	**DCM**	**DCM1**	**DCM2**
Average Time	129.608170	1120.387617	105.988919	65.842693

Table (1) presents the results of all the experiments for both datasets. The baseline method gives an overall accuracy of 50.83% for the Amazon dataset and 64.18% for IMDB dataset. As shown in the table, the accuracy has been considerably improved using the approximation to the FIM, by almost 15% for both datasets. Moreover, it has outperformed the accuracy of DCM mixture model with EM algorithm by 2.5% for Amazon dataset. However, the best accuracy is achieved in using the other parmeterizations, for the model DCM1 based on Eq.(12) and DCM2 based on Eq.(13). As shown in the table, a great improvement is achieved for the second proposed model DCM2, 75.76% for Amazon dataset and 82.85% for IMDB dataset, owing to the extra parameter capability of capturing the overdispersion phenomenon which is extremely problematic in the case of review datasets since the text is usually very short (only a few sentences) and the vocabulary size is huge.

Even though the improvement for the IMDB dataset seems less significant, by taking into consideration the huge size of the dataset, there are around 600 and 2500 more reviews classified correctly for DCM2 as compared to DCM1 and MN-AFSA, respectively. The efficiency of DCM models in sentiment analysis is increased even more by considering the robustness and the requested time until convergence of the algorithms. From the experiments, we noticed that the DCM models tend to be more robust to the random initialization than MN-AFSA. Therefore, we have represented in the table (1) the average accuracy over 5 different runs. Also, the proposed models perform 10 times faster than the original model DCM, as shown in table (2). The gain in simplicity from using the approximation to the FIM makes the multinomial mixture model perform much faster (10x) and be much more accurate (15%) compared to its baseline model. The confusion matrices for all the models are given in

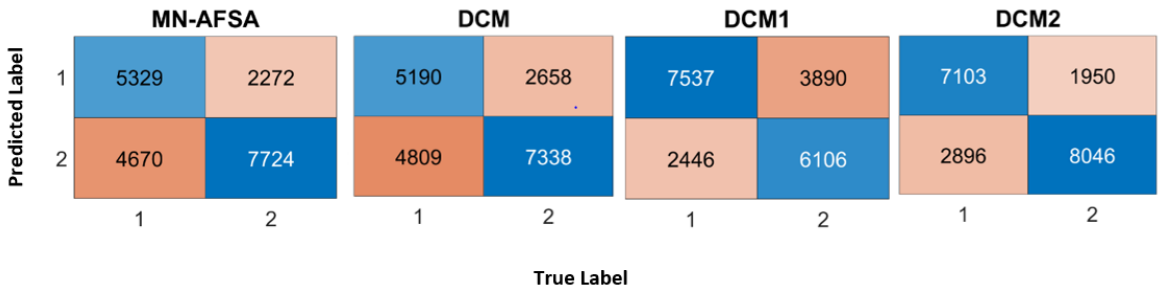

Figure 1: Confusion matrices for sentiment analysis in Amazon dataset using different approaches.

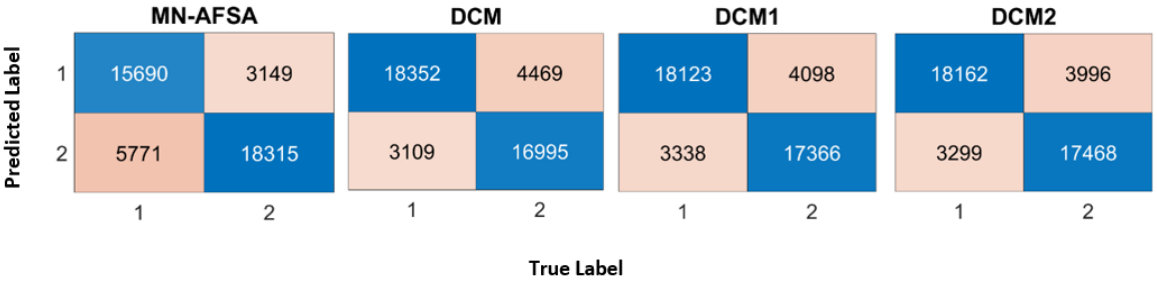

Figure 2: Confusion matrices for sentiment analysis in IMDB dataset using different approaches.

Figure (1) and Figure (2). We can notice that the accuracy of both classes has been greatly improved using the second parameterization DCM2.

Since the considered datasets contain an even number of positive and negative reviews, the improvement from randomly guessing, which yields around 50% accuracy, is significantly improved, by almost 30%. Our models show superior performance to the baseline, and perform the best when taking overdispersion into consideration. Moreover, both proposed frameworks successfully selected the optimal number of components that agrees with the true number of clusters for both text datasets. As shown in Table (1) the model selection approach proposed with DCM gives that the average number of classes are 2.2 and 2.0 for Amazon and IMDB datasets, respectively. Besides, Figure (3) shows the results using the model selection approach based on the multinomial model with approximated FIM. We can see that the number of clusters that minimizes the message length is 2 for both datasets.

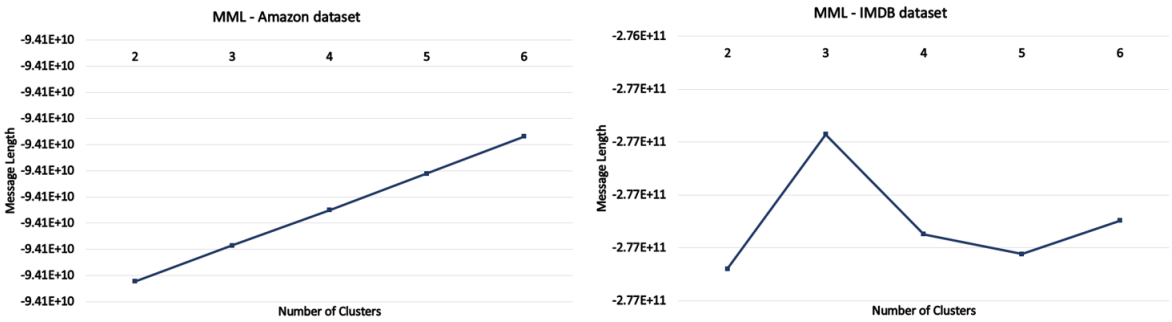

Figure 3: Optimal number of clusters from MML criteria for Amazon and IMDB datasets.

2.4.2 Facial Expression Recognition

Facial expression recognition is one of the most attractive and growing topics nowadays in various research areas, such as human-computer interaction, healthcare systems, and computer graphics. Indeed, using facial expression recognition to understand the emotional state of users can drastically improve the interaction between users and computers or can allow robots to detect the mental states of users on healthcare systems. Also, technologies such as virtual reality (VR) and augmented reality (AR) employ facial expression recognition to deliver a natural and friendly experience to users. To test our models' performance, we have chosen two challenging datasets: MMI [64] and the extended Cohn-Kanade (CK+) [65]. The considered datasets are split into two parts, where half of the images are used to build the visual vocabulary, and the other half is for representation and clustering.

The MMI database contains 1,140 images with a size of 720 x 576 pixels, where each of them belongs in one of 6 basic categories of facial recognition: Anger, Disgust, Fear, Happiness, Sadness, and Surprise). Sample images from the MMI database with different facial expressions are shown in Fig.(4). Participants, where 66% are male, range in age from 19 to 62, having either a European, Asian, or South American ethnic background. **The Extended Cohn-Kanade (CK+) Dataset** is one of the most widely used benchmark databases for testing recognition algorithms due to its high level of difficulty and challenge it represents. It consists of facial behavior of 210 adults (69% female), from 18 to 50 years old, where 81% Euro-American, 13% Afro-American, and 6% other groups. The type of emotion states in CK+ are anger, disgust, fear, happiness, sadness, surprise, as shown in Fig. (5). The duration of

Table 3: Average accuracy (in %), and estimated number of components of DCM algorithms over different runs for facial expression recognition.

	MN	**MN-AFSA**	**DCM**	**DCM1**	**DCM2**	$\hat{K}$	K
MMI	67.74	75.91	76.99	78.28	79.79	6.3	6
CK+	70.94	75.11	76.45	76.35	76.74	6.3	6

image sequences varies from 10 to 60 frames, beginning at the neutral frame and ending at the peak expression frame. Image sequences were digitized into either 640 x 490 or 640 x 480-pixel arrays with an 8-bit gray-scale value. We included all posed expressions that could be labeled as one of the six basic emotion categories, which is about 4,000 images. The recognition accuracy of the facial expression, obtained by applying the different approaches to the considered data sets is shown in Table (3).

As observed in Table (3), the model MN-AFSA achieves higher accuracy than its baseline model and the model DCM2 (DCM with the second parametrization) outperforms all other models, with an overall average clustering accuracy of 79.79% and 76.74% for MMI and CK+, respectively. The reason behind the excellent performance of the DCM2 is two-fold: the capability of its parameters to capture the overdispersion and burstiness phenomena of the data and the great simplification granted by the Minorization-Maximization (MM) framework to our algorithms for high dimensional data. The simplicity granted by the approximation of the Fisher Information Matrix (FIM) makes MN-AFSA more powerful as compared to the DCM model which requires the FIM to be computed in each iteration and linear systems to be solved. Its computation does not only consume time and memory, it also can get intractable as the vocabulary size increases. Also, in some cases the exact FIM is computationally singular, so its inverse cannot be computed and the conditions for identifiability are not satisfied. Moreover, the updates in MN-AFSA and DCM usually violate the parameter constraints, which is perfectly solved by the MM principle as explained theoretically and proved by our experiments.

Lastly, we can observe from Fig.(6) and (7) that all six facial expressions in the MMI dataset are overall distinguished with high accuracy. However, happiness, disgust, sadness, and surprise achieve better results than the rest, due to their distinctive features in eye and mouth parts. On the other hand, expressions of anger and fear

Figure 4: Different sample frames on facial expressions in MMI database.

Figure 5: Different sample frames on facial expressions in CK+ database.

are easily confused with sadness or disgust. The improvement achieved on CK+ is not as good as the previous ones, which might be because of the quality of the image sequences as well as the small number of sample images in some of the classes. It is worth reminding the performance improvement is not the only added value of our algorithms. Computation efficiency, simplicity, independence from initialization, etc. make our proposed models a powerful weapon in the clustering algorithms artillery. Furthermore, as shown in Table (3) and Figure (8) the proposed approaches based on DCM and MN successfully selected the optimal number of components, that agrees with the prespecified one, in both datasets.

2.4.3 Human Action Recognition

Recognizing human activities from video sequences, or images, has been one of the most challenging problems in computer vision due to several issues, such as occlusion, background clutter, changes in scale, viewpoint, lighting conditions, shadows, appearance, frame resolution and the enormous volume of data. Moreover, intra-class dissimilarities and inter-class similarities can increase the challenge. Since different people have different body movements, based on their habits, then actions between different classes may be difficult to distinguish as they may be represented by similar

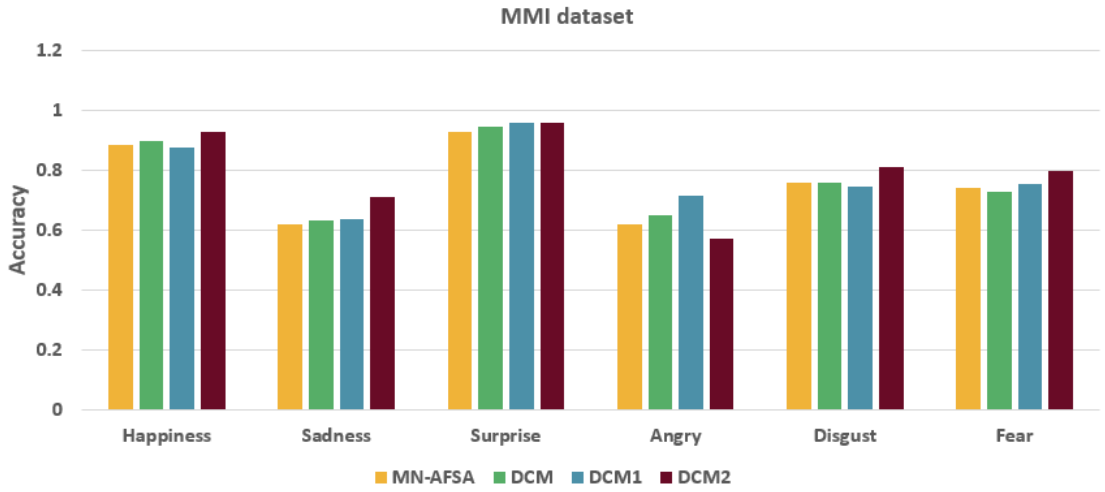

Figure 6: Class recognition accuracy for MMI database.

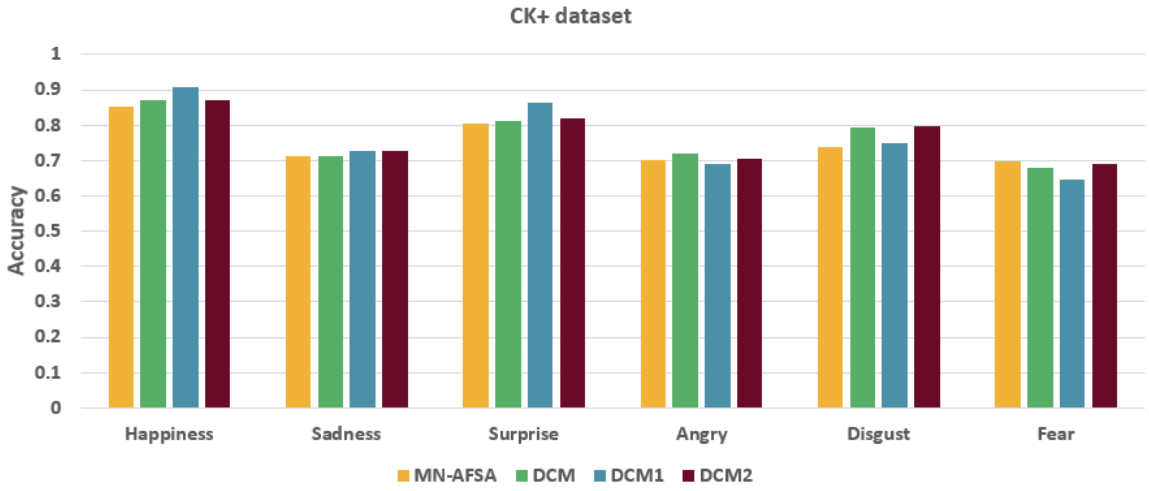

Figure 7: Class recognition accuracy for CK+ database.

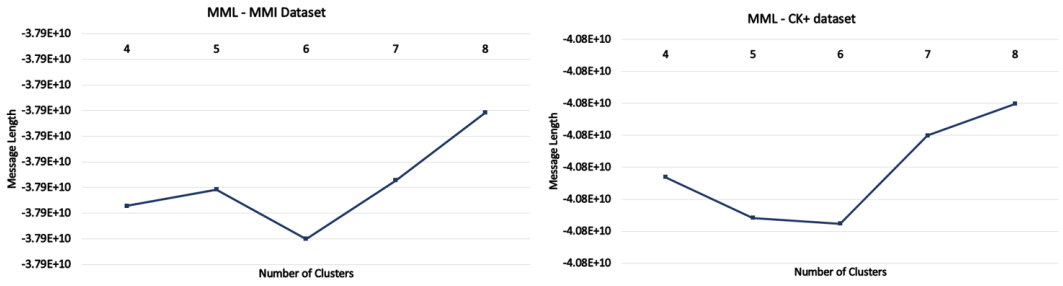

Figure 8: Optimal number of clusters from MML criteria for MMI and CK+ datasets

information. Video surveillance and security systems, action-based human-computer interaction, intelligent robots for human behavior characterization are just a few out of many more applications where human action recognition plays an essential role. We have carefully chosen two challenging datasets, namely KTH [66] and Ballet dataset [67], to test the performance of our models on recognizing multiple high level activities from video sequences composing several actors performing different movements.

KTH Human Action Dataset contains single-action video sequences of 25 actors who perform six different types of human actions; walking, jogging, running, boxing, hand waving, and hand-clapping in 4 different scenarios: outdoors s1, outdoors with scale variation s2, outdoors with different clothes s3 and indoors s4. Representative frames are shown in Fig.(9). It contains a total of 2391 sequences, all taken over homogeneous backgrounds with a static camera with a 25 fps frame rate. The spatial resolution of the sequences is 160x120 pixels, and their average length is around 4 seconds. This dataset has been used at video level, i.e., each video sequence has been represented as a histogram using the bag of features approach.

The Ballet Dataset contains multiple actions in a video sequence, so in that case we perform per-frame classification instead of per-video classification. It consists of 44 labelled video sequences with 8 different ballet dancing activities; standing hand opening, standing still, turning, left-to-right hand opening, leg swinging, jumping, hopping, and right-to-left hand opening. The activities are performed by one woman and two men and only one actor is performing in each video at a particular time. The example sequences from video of Ballet dataset are shown in Fig.(10). This dataset has been used at the frame level, i.e., we extracted the frames and treated each frame as an image.

We can see from Table (4), that our models perform significantly better (more than 15% improvement in the accuracy) as compared to the baseline model. Clearly, the DCM models outperform the MN-AFSA for both tested datasets. Once more, the second parametrization proves itself more accurate and efficient compared to the first parametrization DCM1. However, by closely observing the class accuracy given in Fig.(11), and Fig. (12) we can further discuss where our algorithms fall short. Indeed, a lot of the mistakes made by our algorithm make intuitive sense and can also be generalized for other algorithms. For example, hopping is easily confused with

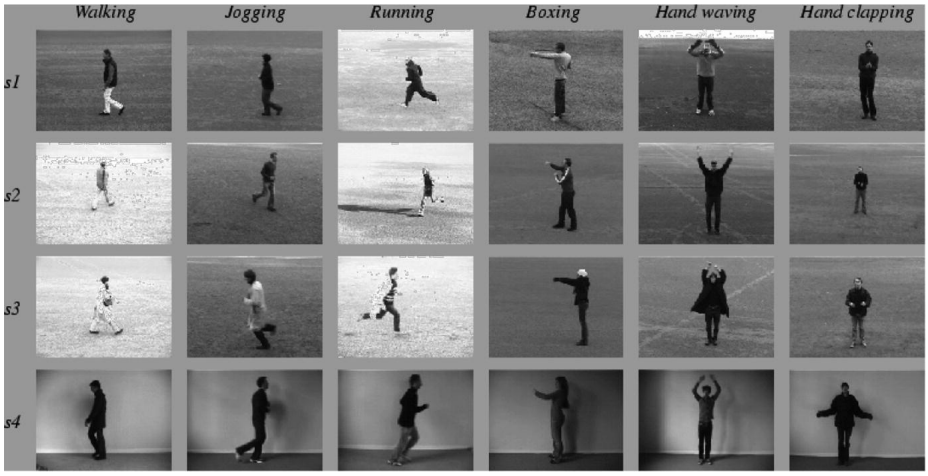

Figure 9: Different sample frames on human actions in KTH database.

Figure 10: Different sample frames on human actions in Ballet database.

jumping or standing still and right-to-left hand opening is easily confused with left-to-right hand opening in the Ballet dataset. Similarly, walking, running and jogging are similar actions and our algorithm tends to confuse them with each other. This is represented in the graphs by their lower recognition accuracy as compared to the other classes. In addition, the performance of our model selection approach based on DCM was evaluated on both datasets, and the average numbers of clusters found over the 10 runs were 6.1 and 8.8 for KTH and Ballet datasets, respectively as shown in Table (4). Moreover, the number of clusters selected by the proposed approach based on MN agrees with the true ones for the two considered datasets as shown in Figure (13). Both clustering accuracy and the selected optimal number of components confirmed that our proposed frameworks are capable of providing promising results in

Table 4: Average accuracy (in %), and estimated number of components of DCM algorithms over different runs for human action recognition.

	MN	**MN-AFSA**	**DCM**	**DCM1**	**DCM2**	$\hat{K}$	K
KTH	58.17	72.76	73.56	76.28	78.37	6.1	6
Ballet	64.95	78.33	81.23	82.82	85.04	8.8	8

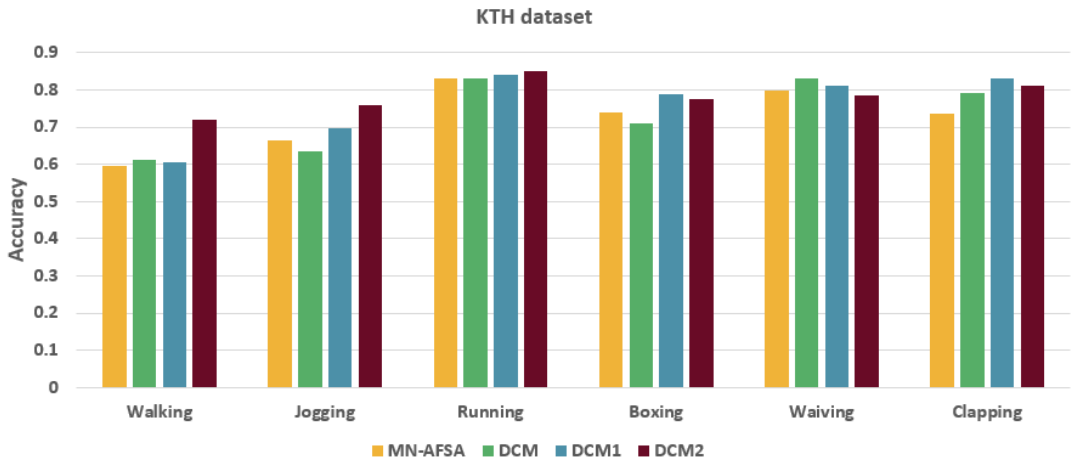

Figure 11: Class recognition accuracy for KTH database.

modeling overdispersed count data.

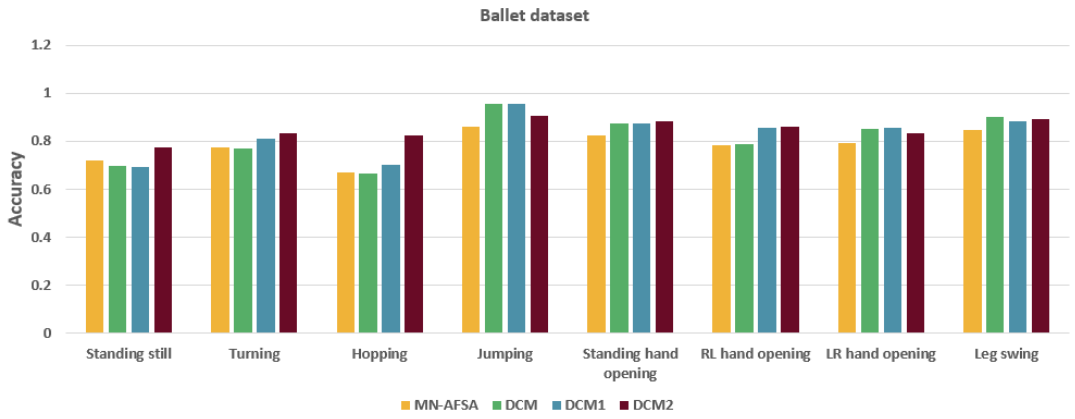

Figure 12: Class recognition accuracy for Ballet database .

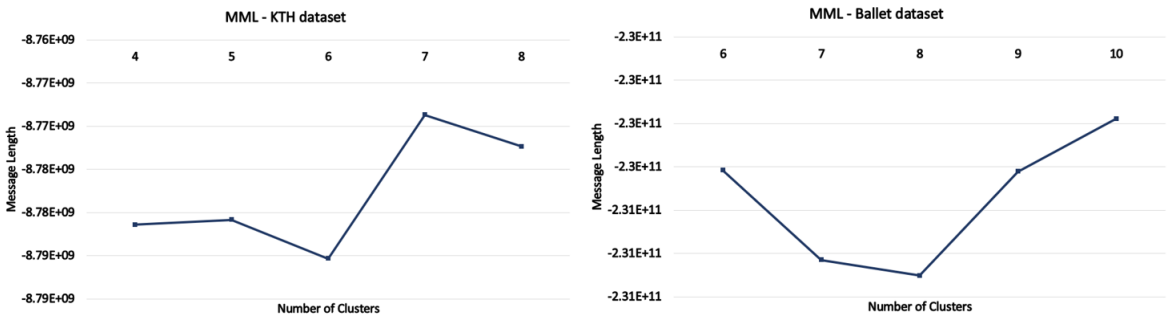

Figure 13: Optimal number of clusters from MML criteria for KTH and Ballet datasets

Chapter 3

Online Mixture-Based Clustering for High Dimensional Count Data Using Neerchal-Morel distribution

In this chapter, we introduce the Neerchal-Morel mixture model where the parameters are to be learnt by the minorization-maximization algorithm. Then, we intergrate the MML criterion to select the optimal number of components in the mixture. In addition, we propose the concept of online learning and adapt the stochastic gradient ascent method to estimate the model parameters. Finally, we demonstrate the experimental results.

3.1 Neerchal-Morel Mixture Model

Finite mixture models offer great flexibility regarding the choice of the statistical distribution and optimal number of clusters that best represents the data [22, 23] as well as the learning algorithm for the mixture′s parameter estimation [14]. Yet, it often remains unclear which of them is suitable for a specific task and how they perform in comparison to each other. Let $\mathcal{X} = \{X_1, \ldots, X_N\}$ be a set of N independently and identically distributed documents or images, where each can be represented as a sparse D dimensional vector of cell counts $X_i = (x_{i1}, \ldots, x_{iD})$, assumed to follow a

Neerchal-Morel distribution, whose probability density function is given by:

$$NMD(X_i|\pi,\rho) = \sum_{d=1}^{D} \pi_d \begin{pmatrix} m_i \\ \mathbf{x_i} \end{pmatrix} [(1-\rho)\pi_1]^{x_{i1}} \cdots [(1-\rho)\pi_d + \rho]^{x_{id}} \cdots [(1-\rho)\pi_D]^{x_{iD}} \tag{48}$$

where D is the vocabulary size; $m_i = \sum_{d=1}^{D} x_{id}$ represents the length of the document; π_d is the probability of emitting a word d which is subject to the constraints $\pi_d > 0$ and $\sum_{d=1}^{D} \pi_d = 1$; and $\rho \in [0,1]$ is the overdispersion parameter. Note that NMD represents a mixture of D multinomial distributions where π_d distributes different weights to each word of the vocabulary, tackling in this way the rare words challenge of high dimensional count data. Moreover, the extra parameter ρ counts for the extra variation found in data which do not adhere to the independency assumption made by the multinomial distribution. Indeed, when $\rho = 0$, the Neerchal-Morel distribution collapses to the latter mentioned.

Then, a finite mixture model of K Neerchal-Morel distributions is denoted as follows:

$$P(X_i|\Theta) = \sum_{k=1}^{K} \mu_k NMD(X_i|\pi_k,\rho_k) \tag{49}$$

where $K \geqslant 1$ is number of components in the mixture, $NMD(X_i|\pi_k,\rho_k)$ is the $k-th$ component of the mixture defined by its own set of parameters $\Theta = \{\mu_1,\ldots,\mu_K,\pi_1,\ldots,\pi_K,\rho_1,\ldots,\rho_K\}$ where μ_k are the mixing weights, which must satisfy the condition $\sum_{k=1}^{K} \mu_k = 1$.

Finally, we can write the data log-likelihood for the whole dataset $\mathcal{X} = \{X_1,\ldots,X_N\}$ as following:

$$\mathcal{L}(\mathcal{X}|\Theta) = \prod_{i=1}^{N} \sum_{k=1}^{K} \log\Big(\mu_k NMD(X_i|\pi_k,\rho_k)\Big) \tag{50}$$

In order to learn the finite mixture model, we seek to maximize the log-likelihood function $\mathcal{L}(\mathcal{X}|\Theta)$ with respect to the parameters Θ. However, the inner summation of the mixture models prevents maximum likelihood (ML) estimates to be obtained analytically. Hence, different methods, such as Newton-Raphson, Expectation-Maximization or Maximization-Minorization can be used to obtain the ML estimates numerically.

The EM algorithm is one of the most familiar approaches used to find maximum likelihood solutions for probabilistic models with missing data [46]. A membership vector $Z_i = (z_{i1},\ldots,z_{iK})$ is assigned to each observation X_i such that $z_{ik} = 1$ if the

object i belongs to the cluster k and all other elements equal to 0. Therefore, the membership vector is a K-dimensional binary random variable whose values must satisfy the conditions $z_{ik} \in \{0, 1\}$ and $\sum_{k=1}^{K} z_{ik} = 1$. The conditional distribution of latent variable $\mathcal{Z}$, given the mixing coefficients π, can be written as:

$$P(\mathcal{Z}|\pi) \sim Multi(\mu) = \prod_{i=1}^{N} \prod_{k=1}^{K} \mu_k^{z_{ik}} \tag{51}$$

Similarly, from Eq.(50) we can write the conditional distribution of data vectors $\mathcal{X}$, given the latent variables $\mathcal{Z}$ and the component parameters Θ. Thus, we can rewrite the complete data likelihood as following:

$$P(\mathcal{X}, \mathcal{Z}|\Theta) = \prod_{i=1}^{N} \prod_{k=1}^{K} \Big(\mu_k NMD(X_i|\pi_k, \rho_k)\Big)^{z_{ik}} \tag{52}$$

The optimization of the complete-data log likelihood function $\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)$ is significantly easier than the direct optimization of the complete-data likelihood function $P(\mathcal{X}, \mathcal{Z}|\Theta)$, therefore we apply the log function and expand the complete data log-likelihood as follows:

$$\begin{aligned}
\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta) &= \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} \ln \Bigg\{ \mu_k \sum_{d=1}^{D} \pi_{kd} \begin{pmatrix} m_i \\ \mathbf{x_i} \end{pmatrix} [(1-\rho_k)\pi_{k1}]^{x_{i1}} \cdots \\
&\qquad [(1-\rho_k)\pi_{kd} + \rho]^{x_{id}} \cdots [(1-\rho_k)\pi_{kD}]^{x_{iD}} \Bigg\} \\
&= \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} \ln \mu_k + \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} \ln \Bigg\{ \sum_{d=1}^{D} \pi_{kd} \begin{pmatrix} m_i \\ \mathbf{x_i} \end{pmatrix} \\
&\qquad [(1-\rho_k)\pi_{k1}]^{x_{i1}} \cdots [(1-\rho_k)\pi_{kd} + \rho]^{x_{id}} \cdots [(1-\rho_k)\pi_{kD}]^{x_{iD}} \Bigg\}
\end{aligned} \tag{53}$$

The EM algorithm begins with an initial estimate of the parameters and then alternates between two steps: an "E-step", in which the conditional expectation of the complete data log likelihood given the observed data and the current parameter estimates is computed, as:

$$\hat{z}_{ik}^{(t)} = P(\mathcal{Z}|\mathcal{X}, \Theta^{(t)}) = \frac{\mu_k^{(t)} NMD(X_i|\pi_k^{(t)}, \rho_k^{(t)})}{\sum_{j=1}^{K} \mu_j^{(t)} NMD(X_i|\pi_k^{(t)}, \rho_k^{(t)})} \tag{54}$$

and an "M-step", in which parameters that maximize the expected complete-data log likelihood from the E-step are determined. Mathematically, the M step is written as:

$$\Theta^{(t+1)} = \arg\max_{\Theta} \mathcal{Q}\left(\Theta|\Theta^{(t)}\right) \tag{55}$$

where

$$\mathcal{Q}\left(\Theta|\Theta^{(t)}\right) = \sum_{\mathcal{Z}} \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta) P(\mathcal{Z}|\mathcal{X}, \Theta^{(t)}) \tag{56}$$

This two-step process always drives the objective function uphill and is iterated until the log likelihood converges. By setting the derivative of the log-likelihood function equal to zero, we obtain the update formula for the mixing weights as:

$$\mu_k^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} \hat{z}_{ik}^{(t)} \tag{57}$$

Intuitively, the mixing weights for each cluster are calculated by summing posterior probabilities (aka responsibilities) of data points in each cluster and dividing by the total number of observations in the dataset. On the other hand, we cannot obtain a closed-form solution for the π_{kd} and ρ_k parameters since they are intertwined in the summation term of the multinomial admixture. Therefore, to solve the parameter's optimization challenge, we apply the Maximization-Minorization (MM) framework which instead of calculating conditional expectations, relies on recognizing and manipulating inequalities.

3.1.1 MM learning Approach

MM framework has attracted significant attention due to its potential in efficiently solving high-dimensional optimization and estimation problems. The key to the construction of an MM algorithm for calculating MLE of the model parameters is to carefully choose an appropriate surrogate function minorizing the log-likelihood function, which must satisfy two properties, mathematically written as:

$$\begin{aligned} \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta^{(t)}) &= \mathcal{G}(\Theta^{(t)}|\Theta^{(t)}), \\ \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta) &\geqslant \mathcal{G}(\Theta|\Theta^{(t)}), \Theta \neq \Theta^{(t)} \end{aligned} \tag{58}$$

In other words, the surface of the surrogate function lies below the surface of the objective function and they are tangent at the point $\Theta = \Theta^{(t)}$, where $\Theta^{(t)}$ represents the current iterate. Given the definition of $\Theta^{(t)}$ and Eq.(58), one can prove that if the surrogate function reaches its maximum value for $\Theta^{(t+1)}$, then MM procedure drives

the likelihood uphill. This is also known as the ascent property and is based on the following inequalities:

$$\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta^{(t+1)}) \geqslant \mathcal{G}(\Theta^{(t+1)}|\Theta^{(t)}) \geqslant \mathcal{G}(\Theta^{(t)}|\Theta^{(t)}) \equiv \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta^{(t)}) \tag{59}$$

The ascent property holds true even if $\mathcal{G}(\Theta|\Theta^{(t)})$ is increased rather than maximized, leading to significant levels of numerical stability and proving to be exceptionally beneficial in case the maximum of the surrogate function can not be found. Therefore, the surrogate function is maximized during the second step of the MM algorithm in order to produce the next iterate Θ^{t+1}.

As we have already emphasized, MM relies on recognizing and manipulating inequalities after close examination of the log-likelihood. Thus, to construct a surrogate function for the log-likelihood in Eq.(53), two minorizations are needed, based on the Jensen and supporting hyperplane inequalities [52]. To simplify the calculations during the first minorization, we define:

$$\Pi_{ikd} = \pi_{kd} \left[(1-\rho_k)\pi_{k1}\right]^{x_{i1}} \cdots \left[(1-\rho_k)\pi_{kd} + \rho_k\right]^{x_{id}} \cdots \left[(1-\rho_k)\pi_{kD}\right]^{x_{iD}} \tag{60}$$

where Π_{ikd}^t would be the quantity evaluated at the t-th iteration. Following this notation, we state the first minorization rooted in the Jensen inequality:

$$\ln\left(\sum_d \Pi_{ikd}\right) \geq \sum_d w_{ikd}^t \ln\left(\frac{\Pi_{ikd}}{w_{ikd}^t}\right) = \sum_d w_{ikd}^t \ln \Pi_{ikd} - \sum_d w_{ikd}^t \ln w_{ikd}^t \tag{61}$$

where

$$w_{ikd}^n = \frac{\Pi_{ikd}^n}{\sum_j \Pi_{ikj}^n} \tag{62}$$

and

$$\begin{aligned} \ln \Pi_{ikd} =& m_i \ln(1-\rho_k) + \ln \pi_{kd} + x_{i1} \ln \pi_{k1} + \cdots + x_{id} \ln\left(\pi_{kd} + \theta_k\right) \\ &+ \cdots + x_{iD} \ln \pi_{kD} \end{aligned} \tag{63}$$

for $\theta_k = \rho_k/(1-\rho_k)$. Then, we apply the supporting hyperplane inequality to separate the parameters π_{kd} and θ_k in the troublesome term $\ln(\pi_{kd} + \theta_k)$. This produces the second minorization, as follows:

$$\ln\left(\pi_{kd} + \theta_k\right) \geq \frac{\pi_{kd}^n}{\pi_{kd}^n + \theta_k^n} \ln\left(\frac{\pi_{kd}^n + \theta_k^n}{\pi_{kd}^n}\pi_{kd}\right) + \frac{\theta_k^n}{\pi_{kd}^n + \theta_k^n} \ln\left(\frac{\pi_{kd}^n + \theta_k^n}{\theta_k^n}\theta_k\right) \tag{64}$$

After straightforward mathematical transformations, the surrogate function takes the form:

$$\begin{aligned}&\sum_i \sum_k z_{ik} \sum_d w_{ikd}^n \left[\sum_j x_{ij} \ln \pi_{kj} + \left(1 - \tfrac{x_{id}\theta_k^n}{\pi_{kd}^n + \theta_k^n}\right) \ln \pi_{kd}\right] \\ &+ \sum_i \sum_k z_{ik} \sum_d w_{ikd}^n \left[\left(m_i - \tfrac{x_{id}\theta_k^n}{\pi_{kd}^n + \theta_k^n}\right) \ln(1-\rho_k) + \tfrac{x_{id}\theta_k^n}{\pi_{kd}^n + \theta_k^n} \ln \rho_k\right]\end{aligned} \tag{65}$$

Finally, standard arguments now yield the final updates:

$$\begin{aligned}\pi_{kd}^{n+1} = &\left(\sum_i z_{ik} \sum_j w_{ikj}^n x_{id} + \sum_i z_{ik} w_{ikd}^n \left(1 - \tfrac{x_{id}\theta_k^n}{\pi_{kd}^n + \theta_k^n}\right)\right) \\ &/ \left(\sum_l \sum_i z_{ik} \sum_j w_{ikj}^n x_{il} + \sum_l \sum_i z_{ik} w_{ikl}^n \left(1 - \tfrac{x_{il}\theta_k^n}{\pi_{kl}^n + \theta_k^n}\right)\right)\end{aligned} \tag{66}$$

$$\rho_k^{n+1} = \left(\sum_i z_{ik} \sum_d \frac{w_{ikd}^n x_{id} \theta^n}{\pi_{kd}^n + \theta_k^n}\right) / \left(\sum_i m_i\right), \quad \theta_k^{n+1} = \frac{\rho_k^{n+1}}{1 - \rho_k^{n+1}} \tag{67}$$

The updates acquired from the MM algorithm are easy, intuitive, offer numerical stability, natural adaption to parameter constraints, and scalability to high-dimensions, turning the MM approach into a powerful weapon in the arsenal of optimization algorithms.

3.1.2 MML Model Selection Criterion

Even though by using MM algorithms we solve many of the weaknesses of EM algorithm for mixture models, the requirement of knowing the number of clusters during the initialization step still remains. To overcome this limitation, we adopt the approach in [24], which is based on the MML criterion [23]. Minimum Message Length (MML) selection criterion consists of evaluating the statistical model's ability to compress a message containing the data. The basic philosophy of the minimum encoding methods, summarized by Wallace and Freeman [68], is such that we first estimate the parameters of the mixture model and under the assumption that these are the true values, we encode the data. The shorter the code achieved, the better is the representation of the data by the estimated parameters. In the case of unsupervised learning, the message includes two parts where the first part encodes the model using only prior information about the model and no information about the data and the second part encodes only the data. According to information theory, the optimal number of clusters K is the candidate value which minimizes the message length [22], given by:

$$\hat{\Theta} = \arg\min_{\Theta} \left\{ -\log(h(\Theta)) - \log(P(\mathcal{X}|\Theta)) + \frac{1}{2} \log|F(\Theta)| + \frac{D}{2}\left(1 + \log\frac{1}{12}\right)\right\} \tag{68}$$

where $h(\Theta)$ is the prior probability, $P(\mathcal{X}|\Theta)$ is the likelihood for the complete data set, and $|F(\Theta)|$ is the determinant of the expected Fisher information matrix. To use Eq. (68), we must first choose a prior distribution $h(\Theta)$, and derive an expression for the determinant of the expected Fisher Information matrix $|F(\Theta)|$. Fisher information matrix is the determinant of the Hessian matrix of the logarithm of minus the likelihood of the mixture [69]. The Hessian matrix of a mixture leads to a complicated analytical form of MML which cannot be easily reproduced. Therefore, we approximate we replace the Fisher Information Matrix $F(\Theta)$ by the complete-data Fisher information matrix $F_c(\Theta) \equiv -E\left[\partial^2_\Theta \log p(\mathcal{X}, \mathcal{Z} \mid \mathbf{\Theta})\right]$ which upperbounds $F(\Theta)$ [14]. $F_c(\Theta)$ has block-diagonal structure as follows:

$$F_c(\Theta) = N\text{block-diag}\left\{\mu_1 F^{(1)}\left(\Theta_1\right), \ldots, \mu_K F^{(1)}\left(\Theta_K\right), \mathbf{M}\right\} \tag{69}$$

where $F^{(1)}\left(\Theta_k\right)$ is the Fisher matrix for a single observation known to have been produced by the k-th component, and $\mathbf{M}$ is the Fisher matrix of a multinomial distribution (where $|\mathbf{M}| = \left(\mu_1\mu_2\cdots\mu_K\right)^{-1}$)[70]. The approximation of $F\left(\Theta\right)$ by $F_c\left(\Theta\right)$ becomes exact in the limit of nonoverlapping components. Following the logic in [14] we adopt a prior expressing lack of knowledge about the mixture parameters. Naturally, we model the parameters of different components as a priori independent and also independent from the mixing probabilities, as follows:

$$p(\Theta) = p\left(\mu_1, \ldots, \mu_k\right) \prod_{k=1}^{K} p\left(\Theta_k\right) \tag{70}$$

For each factor $p\left(\Theta_k\right)$ and $p\left(\mu_1, \ldots, \mu_k\right)$, we adopt the standard noninformative Jeffreys' prior (refer to [71])

$$\begin{array}{c} p\left(\Theta_k\right) \propto \sqrt{\left|F^{(1)}\left(\Theta_k\right)\right|} \\ p\left(\mu_1, \ldots, \mu_K\right) \propto \sqrt{|\mathbf{M}|} = \left(\mu_1\mu_2\cdots\mu_K\right)^{-1/2} \end{array} \tag{71}$$

for $0 \leq \mu_k \leq 1$ and $\sum_{k=1}^{K} \mu_k = 1$. Finally, the updated MML criterion becomes:

$$\hat{\Theta} = \arg\min_{\Theta}\left\{\frac{D}{2}\sum_{k=1}^{K}\log\left(\frac{N\mu_k}{12}\right) + \frac{K}{2}\log\frac{N}{12} + \frac{K(D+1)}{2} - \log p(\mathcal{X} \mid \Theta)\right\} \tag{72}$$

where, as usual, $-\log p(\mathcal{X} \mid \Theta)$ is the code-length of the data; $N\mu_K$ is the expected number of data points generated by the k th component of the mixture (it can also be

seen as an effective sample size from which Θ_k is estimated); $(D/2)\log(N\mu_K)$ represents the "optimal" (in the MDL sense) code length for each Θ_k and $(K/2)\log(N/12)$ term is related to the estimation of μ_ks over all N observations. Please note, if $\mu_k = 0$ the objective function goes to negative infinity. However, we only need to code the parameters of those components whose probability is nonzero. Finally, from a Bayesian point of view, Eq.(72) is equivalent, for non-zero-probability distributions, to an a posteriori density resulting from the adoption of improper Dirichlet-type prior for the μ_k parameter , such that:

$$p(\mu_1, \ldots, \mu_K) \propto \exp\left\{-\frac{D}{2}\sum_{k=1}^{K}\log\mu_K\right\} \tag{73}$$

and a flat prior leading to ML estimates for the mixture parameters Θ_k. Since Dirichlet priors are conjugate to multinomial likelihoods [71], the EM algorithm undergoes a minor modification of the update of the component's weight in the M-step to:

$$\widehat{\mu}_k = \frac{\max\left(\sum_i \hat{z}_{ik} - \frac{D}{2}, 0\right)}{\sum_j \max\left(\sum_i \hat{z}_{ij} - \frac{D}{2}, 0\right)} \tag{74}$$

The approach we are using allows the model to be initialized with a large value of K, thus surpassing the limitation of initialization dependency and the tendency of such algorithms to get convergence in a local minimum. Starting with a large value of K may lead to several empty components and there will be no need to estimate, and transmit, their parameters. The advantage of the new update formula is its pruning behavior, that when some of the μ_k go to zero they will be removed, preventing in such way the algorithm from going to the boundaries of the parameter space.

3.2 Online Neerchal-Morel Mixture Model

As new data become available everyday, the built model should be able to seize the new information and reflect the changes in the model's parameter estimates. The traditional iterative algorithms fail to efficiently address the situation. Indeed, the traditional batch methods require the model to be retrained from scratch, therefore, demanding a great deal of computational time and memory usage. Moreover, the whole dataset must be available in the memory at every iteration. Considering the volume, velocity and variety in which new data becomes available everyday, the approach lacks feasibility. On the other hand, online learning algorithms show promise

to address the batch learning drawbacks and efficiently update the already built models. The online scheme can be easily adapted to the Neerchal Morel Mixture models with MM learning approach. The online MM algorithm is essentially a stochastic approximation procedure and can be considered as the stochastic analog of the deterministic MM algorithms. Therefore, we use the Stochastic Gradient Ascent Learning method to update the component's parameters of the mixture model with the new input vector. Thus, lets consider a dataset represented by K multivariate Neerchal-Morel distributions with parameters Θ_N. Suppose, now, at time $t+1$, a new data vector X_{t+1} becomes available, thus, the mixture model parameters need to be updated incrementally considering the new data vector. The problem we aim to solve is how to update the different mixture models parameters. For this goal, we use the stochastic ascent gradient parameter updating proposed by [72] , where the model parameters Θ_{N+1} will be updated according to:

$$\Theta_{N+1}^{(t+1)} = \Theta_N^{(t)} + \frac{1}{N+1} \frac{\partial \mathcal{L}\left(X_{N+1}, Z_{N+1} | \Theta_N^{(t)}\right)}{\partial \Theta_N^{(t)}} \tag{75}$$

Naturally, to ensure the unity constraint of the mixing proportions μ_k as well as the unity constraint of feature proportions π_{kd} we have considered new variables $\alpha_1, \ldots, \alpha_{K-1}$ and $\beta_{k1}, \ldots, \beta_{k,D-1}$, respectively, that belong to $\mathbb{R}$ by introducing the Logit transformation:

$$\alpha_k = \log \frac{\mu_k}{\mu_K}, \quad k = 1, \ldots, K-1 \tag{76}$$

and

$$\beta_{kd} = \log \frac{\pi_{kd}}{\pi_{kD}}, \quad d = 1, \ldots, D-1 \tag{77}$$

The mixing proportion, in this case, can be updated as follows:

$$\mu_k^{(t+1)} = \frac{\exp\left(\alpha_k^{(t+1)}\right)}{1 + \sum_{k=1}^{K-1} \exp\left(\alpha_k^{(t+1)}\right)}, k = 1, \ldots, K-1 \tag{78}$$

$$\mu_K^{(t+1)} = \frac{1}{1 + \sum_{k=1}^{K-1} \exp\left(\alpha_k^{(t+1)}\right)} \tag{79}$$

such that,

$$\alpha_k^{(t+1)} = \alpha_k^{(t)} + \frac{1}{N+1}\left(z_{N+1,k}^{(t)} - \mu_k^{(t)}\right), k = 1, \ldots, K-1 \tag{80}$$

Similarly, the feature weights, in this case, can be updated as follows:

$$\pi_{kd}^{(t+1)} = \frac{\exp\left(\beta_{kd}^{(t+1)}\right)}{1 + \sum_{d=1}^{D-1} \exp\left(\beta_{kd}^{(t+1)}\right)}, d = 1, \ldots, D-1 \tag{81}$$

$$\pi_{kD}^{(t+1)} = \frac{1}{1 + \sum_{d=1}^{D-1} \exp\left(\beta_{kd}^{(t+1)}\right)} \tag{82}$$

such that,

$$\beta_{kd}^{(t+1)} = \beta_{kd}^{(t)} + \frac{1}{N+1}\left(z_{N+1,k}^{(t)} - \mu_k^{(t)}\right) \tag{83}$$

where $z_{N+1,k}^{(t)}$ is the posterior probability of the new coming vector given the old set of parameters $\Theta^{(t)}$. Finally, the update formula for the overdispersion parameter is given below:

$$\rho_k^{(t+1)} = \rho_k^{(t)} + \frac{z_{N+1,k}^{(t+1)}}{N+1} \frac{\frac{X_{N+1}\theta_k^{(t)}}{\pi_{kd}^{(t)} + \theta_k^{(t)}}}{m_{N+1}}, \quad \theta_k^{n+1} = \frac{\rho_k^{n+1}}{1 - \rho_k^{n+1}} \tag{84}$$

Note how the learning parameter $\frac{1}{N+1}$ decreases when the number of observations increase. The decreasing property helps the model to overcome catastrophic forgetting, that is, the tendency to forget previously learned information upon learning new information.

3.3 Experimental Results

The importance of online learning in the real-world scenarios, where data are generated with high velocity, can be demonstrated by many applications in a wide spectrum of domains. Consider, for example, occupant activity and behavior modeling as key information for minimizing energy consumption in smart buildings [73]. Moreover, real-time surveillance has gained a significant value nowadays from traffic monitoring to prevention of possible threats, such as terrorist attacks, environmental hazards or disease outbreaks. The most recent threat to global public health has emerged by a severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), causing coronavirus disease 2019 (COVID-19). COVID-19 emerged in late 2019 and caused an ongoing pandemic, with more than 200 countries affected around the world [74]. Researchers, policy makers and other interested parties immediately initiated machine learning projects in order to help solving the big questions related to the outbreak and limit its impact on our society as much as possible.

3.3.1 COVID-19-Related Applications

To test the effectiveness of the proposed model and demonstrate the advantages of online learning in a scenario where a sheer volume of data is generated daily, we use two challenging datasets concerning COVID-19.

First, given the rapid increase of coronavirus related literature, text and data mining approaches are needed to provide insights and find answers to high priority scientific questions. To date, the COVID-19 Open Research Dataset (CORD-19) contains 52481 articles in English, divided as follows: 26434 with general information, 10639 related to business, 2601 articles in technology, 506 in the science domain and 12266 finance-related. The amount of the articles increases daily, therefore, we apply the proposed online algorithm to the dataset in order to test its effectiveness. Here, each article is represented as a vector of word counts, achieved after performing the Bag-Of-Words (BOW) approach.

Second, chest x-rays are widely used as a tool for diagnosing COVID-19 to mitigate the overwhelming demand for tests. Therefore, machine learning and more advanced methods, such as deep learning, have proven to be highly effective in identifying patterns of the disease found in patient's lungs. The dataset used contains 67 images of patients diagnosed with COVID-19, 2 patients diagnosed with pneumonia and 21 patients whose test's results were negative. We aim to use our proposed models to efficiently distinguish between healthy patients, COVID-19 patients or pneumonia patients. Again, each image is represented as a vector of counts (i.e. latent aspects). The methodology used for the extraction of the features is Scale-Invariant feature transformation. Then, the extracted features are clustered into visual words using the K-means algorithm, where each image is represented as a histogram of frequencies. Finally, a Probabilistic Latent Semantic Analysis (pLSA) is applied to transform the number of visual words to a predetermined D dimension. Sample images form the COVID-19 chest x-rays dataset are shown in Fig.(14).

The recognition accuracy for both datasets, obtained by applying the different approaches is shown in Table (5). Here, the accuracy has been considerably improved using the new proposed mixture model. Specifically, the Neerchal Morel Mixture Model outperforms the Multinomial and DCM models, with an overall average clustering accuracy of 78.30%, and 83.33% for CORD-19 and COVID-19 datasets, respectively. The justification of the excellent performance of our model is two-fold: the

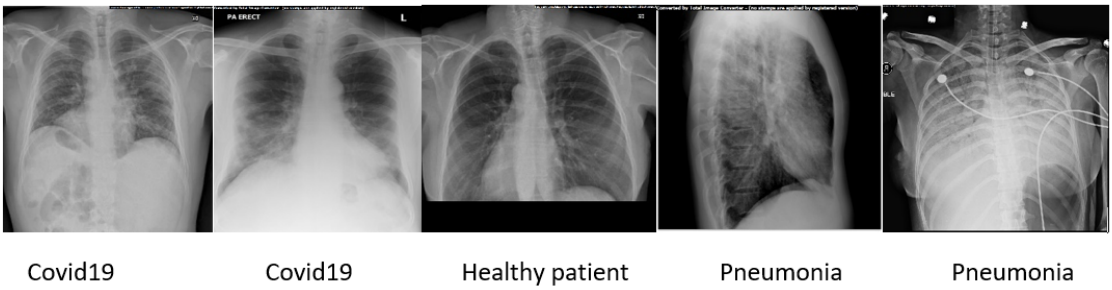

Figure 14: Different sample frames of chest x-rays in COVID-19 database.

capability of its parameters to capture the overdispersion and burstiness phenomena by fitting the high dimensional data into mixtures of multinomial mixtures, and the efficiency of the MM principle as an optimization algorithm. Indeed, as shown by the class accuracies in Fig.(15), the NMD model allows a better representation of the variance of the data. We can observe from the plot that the model can distinguish better the classes which have more articles, as represented by a higher clustering accuracy. Note, for example, the MN and DCM models perform poorly for the science-related articles, whereas the NMD model perform significantly better, demonstrating once again that assigning the extra weight parameter for the features captures better the overdispersion of the data and enhances the clustering accuracy. Moreover, the proposed approach increases the accuracy of the clustering accuracy by almost 20% as compared to the baseline approach (MN) in the COVID-19 dataset, even though the dataset is extremely challenging and highly imbalanced. Other than better clustering accuracy, the proposed model is also more efficient in terms of time and memory usage. Additionally, adopting the online learning to the Neerchal-Morel Mixture Model, turns the model into a powerful weapon when dealing with daily and exponentially increasing volume of data. The performance of the online learning algorithm is shown in Fig.(16). After the offline learning of the mixture parameters, the new coming data vectors are immediately used to update the estimates. In Fig.(16), the accuracy of the model after batches of 100 new data points is plotted. Here, the accuracy of the model is not significantly increased after the new data become available, therefore, the trade-off between performance and time or memory resources depends on the application and preference of the user. Moreover, as shown in Table (5) the model selection approach proposed with NMD finds that the average number of classes are 5.4 and 3.1 for CORD-19 and COVID-19 datasets, respectively, in accordance with

Table 5: Average accuracy (in %), and estimated number of components of NMD algorithms over different runs for COVID-19 datasets.

	MN	**DCM**	**NMD**	$\hat{K}$	K
CORD-19	65.62	70.00	78.30	5.4	5
COVID-19	65.21	67.39	83.33	3.1	3

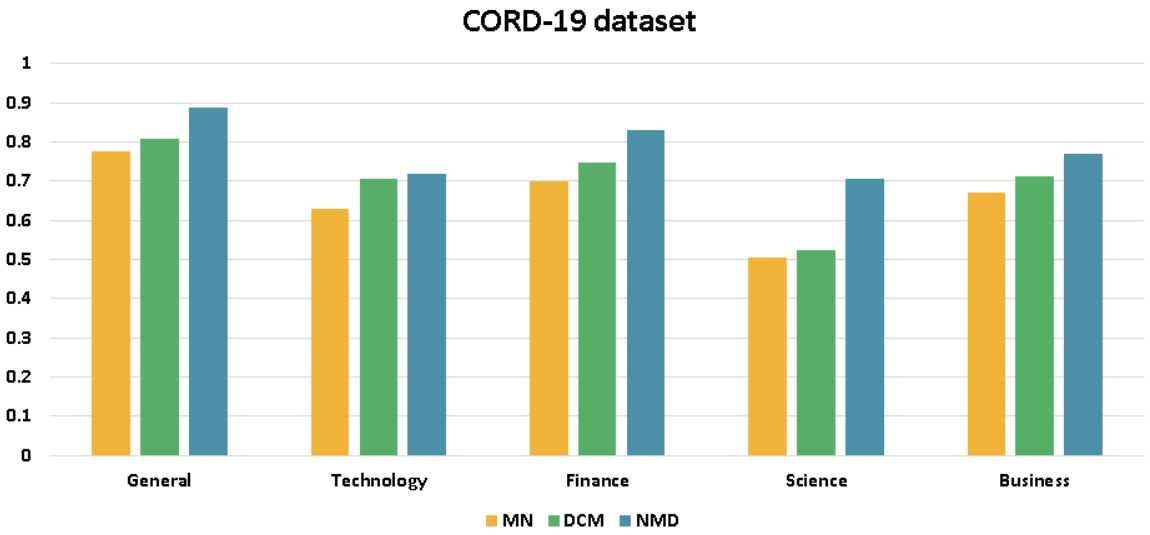

Figure 15: Class accuracies for CORD-19 dataset.

their true number of clusters.

3.3.2 Human Action Recognition

Recognizing human activities from video sequences, or images, has been one of the most challenging problems in computer vision in the recent years. The online learning shows great potential and brings tremendous advantages in applications such as video surveillance and security systems, action-based human-computer interaction or intelligent robots for human behavior characterization, where immediate decision-making carries a crucial role. However, alongside to the enormous volume, high dimensionality and heterogeneity nature of datasets in the mentioned domain, several other issues arise, such as occlusion, background clutter, changes in scale, viewpoint, lighting conditions, shadows, appearance, frame resolution, etc. Therefore, given the increased difficulty to efficiently represent and model the data, we have carefully chosen two

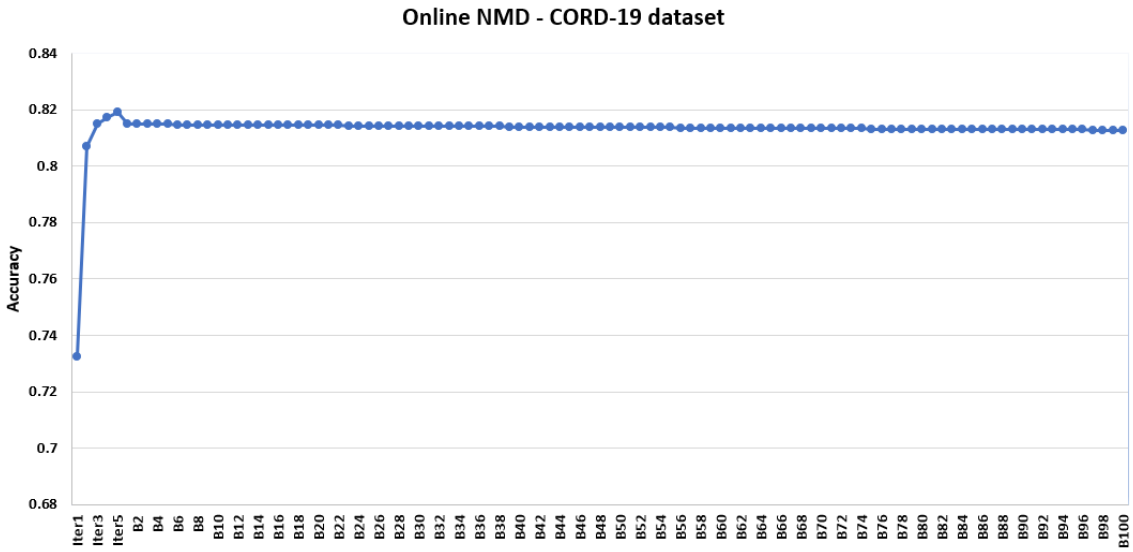

Figure 16: Online NDM algorithm accuracies for CORD-19 dataset.

datasets, namely KTH [66] and Ballet dataset [67], to test the performance of our models on recognizing multiple high level activities from video sequences composing several actors performing different movements.

KTH Human Action Dataset contains single-action video sequences of 25 actors who perform six different types of human actions; walking, jogging, running, boxing, hand waving, and hand-clapping in 4 different scenarios: outdoors s1, outdoors with scale variation s2, outdoors with different clothes s3 and indoors s4. Representative frames are shown in Fig.(17). It contains a total of 2391 sequences, all taken over homogeneous backgrounds with a static camera with a 25 fps frame rate. The spatial resolution of the sequences is 160x120 pixels, and their average length is around 4 seconds. This dataset has been used at video level, i.e., each video sequence has been represented as a histogram using the bag of features approach. [75]

The Ballet Dataset contains multiple actions in a video sequence, so in that case we perform per-frame classification instead of per-video classification. It consists of 44 labelled video sequences with 8 different ballet dancing activities; standing hand opening, standing still, turning, left-to-right hand opening, leg swinging, jumping, hopping, and right-to-left hand opening. The activities are performed by one woman and two men and only one actor is performing in each video at a particular time. The example sequences from video of Ballet dataset are shown in Fig.(18). This dataset has been used at the frame level, i.e., we extracted the frames and treated each frame

Figure 17: Different sample frames on human actions in KTH database.

Table 6: Average accuracy (in %), and estimated number of components of NMD algorithms over different runs for human action recognition.

	MN	**DCM**	**NMD**	$\hat{K}$	K
KTH	72.75	73.56	78.85	5.8	6
Ballet	76.93	81.23	84.81	8.2	8

as an image.

From the experimental results shown in Table (6), we observe that the model using the Neerchal-Morel distribution and MM learning approach performs better as compared to the Multinomial and DCM distributions. The proposed model achieves a clustering accuracy of 78.85% and 84.81% for the KTH and Ballet datasets, respectively. The gain in performance is higher for the KTH dataset, with almost 12% as compared to the multinomial model and 5% as compared with the DCM model. To perform the online learning for the Ballet dataset, we splitted the dataset into batches of 50 datapoints. The accuracy of the model is increasing as new data become available, as shown in Fig.(21). Thus, the incremental learning serves as a powerful tool to tackle the many challenges brought by huge volume and high dimensionality nature of count data. In addition, the performance of our model selection approach based on

Figure 18: Different sample frames on human actions in Ballet database .

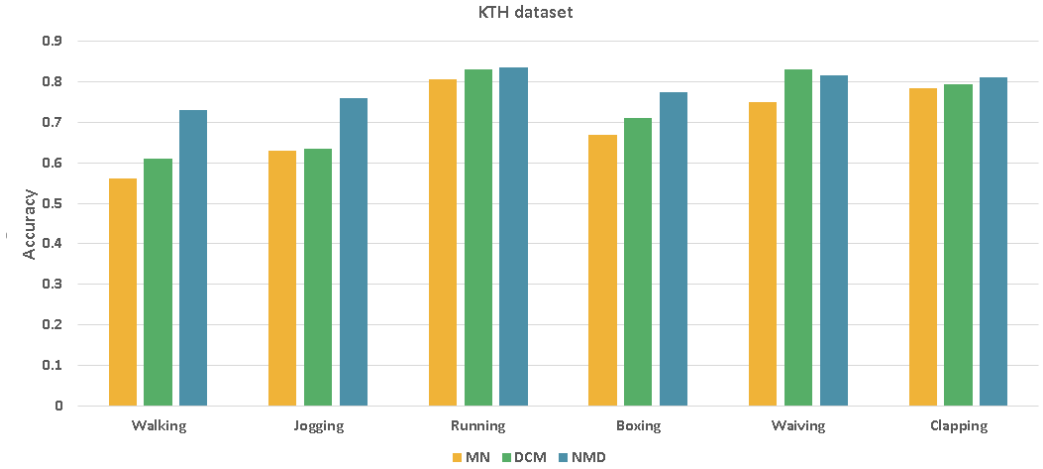

Figure 19: Class recognition accuracy for KTH database.

NMD was evaluated on both datasets, and the average numbers of clusters found over the 10 runs were 5.8 and 8.2 for KTH and Ballet datasets, respectively as shown in Table (6). Both clustering accuracy and the selected optimal number of components confirmed that our proposed frameworks are capable of providing promising results in modeling overdispersed count data. Finally, the random initialization along with the model selection criteria, where the number of clusters decreases gradually, mitigates the tendency of the algorithm to overfit the data or get stuck in a local maxima.

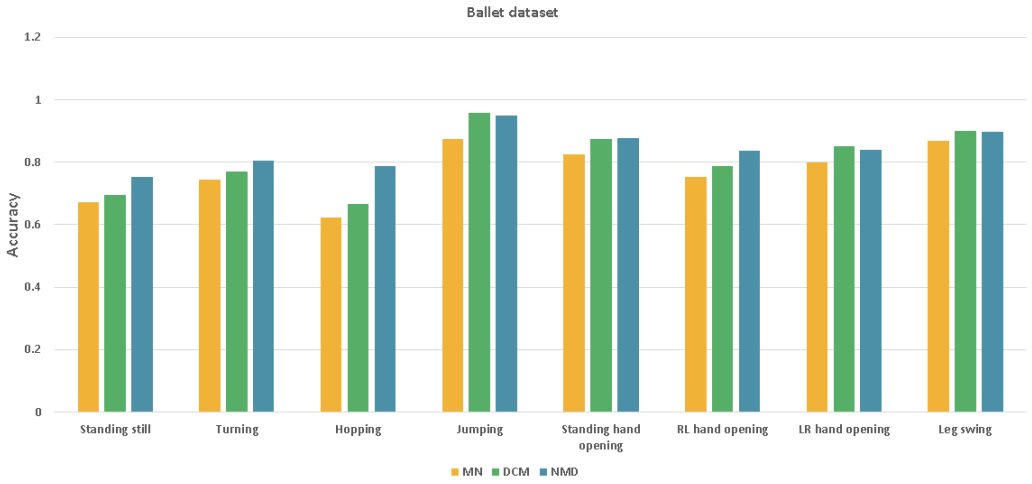

Figure 20: Class recognition accuracy for Ballet database .

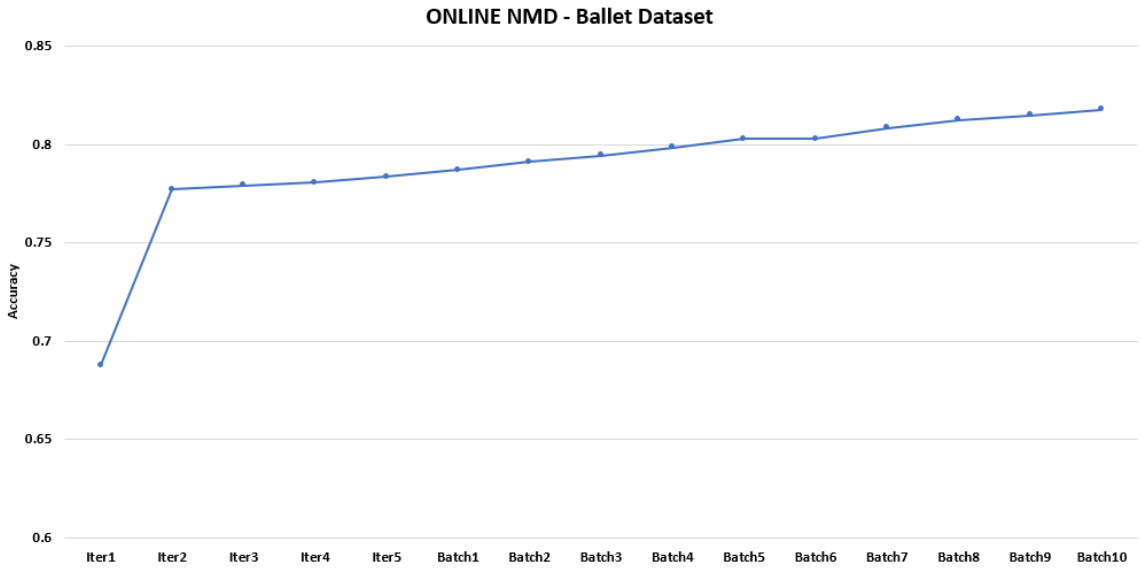

Figure 21: Online NDM algorithm accuracies for Ballet dataset.

Probability Distributions	Description		Drawbacks
	Characteristic	Learning Approach	
Multinomial Distribution	The most widely used distribution to model multivariate count data	AFSA	Independency assumption
		EM	The calculation of the exact FIM becomes computationally expensive when dealing with huge vocabulary datasets, Independency assumption
Dirichlet Compound Multinomial Distribution	Generalization of the multinomial model obtained by introducing the Dirichlet as prior		Complications with Gamma function , Independency assumption
	Alternative parametrization in terms of rising polynomials	MM	Independency assumption
	Alternative parametrization in terms of rising polynomials with proportion vector + extra overdispersion parameter		Independency assumption
Neerchal-Morel Distribution	Finite mixtures of multinomial distributions, weight parameters for each feature in each cluster		NA

Figure 22: Summary of Probability Distribution and Learning Approaches

Chapter 4

Conclusion

4.1 Contribution

Our major contributions in this thesis can be summarized as follows:
First, we have compared three new parametric models for clustering based on finite mixture model of multinomial and Dirichlet Compound Multinomial (DCM) distributions. By using a complete-data information matrix, approximation of the Fisher Information matrix, we were able to simplify the computations and address the complexity of high-dimensional count data. Moreover, we utilize two alternative representations of DCM distribution, which have several properties that make them more convenient than the original DCM, such as: (1) replacing ratios of Gamma functions by rising polynomials considerably simplifies the calculations and derivations, (2) a second parameter is added which can model overdispersion of the data.
Second, we have used a powerful minorization-maximization (MM) framework to address the mixture's parameter estimation. MM algorithms avoid many complications that arise during the optimization of DCM mixture models due to the non-existence of a closed-form solution and have proven to be easy to implement and provide remarkable numerical stability.
Third, to tackle the problem of the unknown number of clusters in unsupervised learning, we have implemented two different approaches of minimum message length model selection criterion. In the second approach, the weights of irrelevant mixture components are driven towards zero, which resolves the problem of knowing the number of clusters beforehand.

Fourth, a Neerchal-Morel Mixture Model is developed, which due to its representations as mixture of multinomial distributions captures overdispersion of high dimensional count data by assigning different weights to features in each cluster.
Finally, we adapt the latter model into an online scheme, able to address the high velocity of data in real-time applications.
The effectiveness and comparison of the newly proposed mixture models was shown through extensive experiments on challenging clustering problems in a wide range of applications, such as: sentiment analysis, topic detection, facial expression recognition, human action recognition and medical diagnosis. The complete-data information matrix, along with the gained simplicity of AFSA, make the Multinomial mixture model comparable to the DCM mixture model, which is designed to capture the burstiness phenomena of count data. However, the model based on the first parametrization of DCM distribution, supported by the MM framework, achieves higher accuracy on similar levels of simplicity. The results show that the proposed mixture model based on the second parametrization of DCM distribution outperforms the other models, owing to the ability of the extra parameter to capture overdispersion. Moreover, online NDM mixture model has proven to be a robust algorithm which has achieved better or similar performance with the offline model and has, therefore, been able to retain a satisfactory trade off between classification accuracy and time performance. Finally, our unsupervised algorithms provide promising results in selecting the optimal number of clusters by optimizing the message length of the data efficiently. Please refer to Fig.22 or a summary of the contribution in this thesis.

4.2 Future Work

We plan to further improve the online learning algorithm by adding the model selection criteria so that the number of clusters and therefore, the model that best represents the data can be selected automatically after each new data instance or batch. When the range of candidate number of clusters is large, the time-complexity increases significantly. We aim to find an efficient solution for the model selection of the online algorithm.

Bibliography

[1] Seshadri Tirunillai and Gerard J Tellis. Mining marketing meaning from online chatter: Strategic brand analysis of big data using latent dirichlet allocation. *Journal of Marketing Research*, 51(4):463–479, 2014.

[2] Wullianallur Raghupathi and Viju Raghupathi. Big data analytics in healthcare: promise and potential. *Health information science and systems*, 2(1):3, 2014.

[3] Nizar Bouguila. Clustering of count data using generalized dirichlet multinomial distributions. *IEEE Trans. Knowl. Data Eng.*, 20(4):462–474, 2008.

[4] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134, 2000.

[5] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[6] Nizar Bouguila. Count data modeling and classification using finite mixtures of distributions. *IEEE Transactions on Neural Networks*, 22(2):186–198, 2010.

[7] Nizar Bouguila and Ola Amayri. A discrete mixture-based kernel for svms: Application to spam and image categorization. *Inf. Process. Manag.*, 45(6):631–642, 2009.

[8] Nizar Bouguila and Walid ElGuebaly. On discrete data clustering. In Takashi Washio, Einoshin Suzuki, Kai Ming Ting, and Akihiro Inokuchi, editors, *Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference, PAKDD 2008, Osaka, Japan, May 20-23, 2008 Proceedings*, volume 5012 of *Lecture Notes in Computer Science*, pages 503–510. Springer, 2008.

[9] Inderjit S Dhillon and Dharmendra S Modha. Concept decompositions for large sparse text data using clustering. *Machine learning*, 42(1-2):143–175, 2001.

[10] Slava Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing*, 35(3):400–401, 1987.

[11] Kenneth W. Church and William A. Gale. Poisson mixtures. *Natural Language Engineering*, 1(2):163–190, 1995.

[12] Slava M Katz. Distribution of content words and phrases in text and language modelling. *Natural language engineering*, 2(1):15–59, 1996.

[13] John Hinde and Clarice GB Demétrio. Overdispersion: models and estimation. *Computational statistics & data analysis*, 27(2):151–170, 1998.

[14] Mario A. T. Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (3):381–396, 2002.

[15] Nizar Bouguila and Walid ElGuebaly. Discrete data clustering using finite mixture models. *Pattern Recognit.*, 42(1):33–42, 2009.

[16] Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

[17] Makoto Iwayama and Takenobu Tokunaga. Cluster-based text categorization: a comparison of category search strategies. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 273–280. Citeseer, 1995.

[18] Nizar Bouguila and Walid ElGuebaly. A generative model for spatial color image databases categorization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2008, March 30 - April 4, 2008, Caesars Palace, Las Vegas, Nevada, USA*, pages 821–824. IEEE, 2008.

[19] Ali Shojaee Bakhtiari and Nizar Bouguila. An expandable hierarchical statistical framework for count data modeling and its application to object classification.

In *IEEE 23rd International Conference on Tools with Artificial Intelligence, ICTAI 2011, Boca Raton, FL, USA, November 7-9, 2011*, pages 817–824. IEEE Computer Society, 2011.

[20] Mehran Sahami and Daphne Koller. *Using machine learning to improve information access*. PhD thesis, Stanford University, Department of Computer Science, 1998.

[21] Sanjiv K Bhatia and Jitender S Deogun. Conceptual clustering in information retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(3):427–436, 1998.

[22] Rohan A Baxter and Jonathan J Oliver. Finding overlapping components with mml. *Statistics and Computing*, 10(1):5–16, 2000.

[23] Chris S Wallace and David L Dowe. Mml clustering of multi-state, poisson, von mises circular and gaussian distributions. *Statistics and Computing*, 10(1):73–83, 2000.

[24] Martin HC Law, Mario AT Figueiredo, and Anil K Jain. Simultaneous feature selection and clustering using mixture models. *IEEE transactions on pattern analysis and machine intelligence*, 26(9):1154–1166, 2004.

[25] Nizar Bouguila. Count data clustering using unsupervised localized feature selection and outliers rejection. In *IEEE 23rd International Conference on Tools with Artificial Intelligence, ICTAI 2011, Boca Raton, FL, USA, November 7-9, 2011*, pages 1020–1027. IEEE Computer Society, 2011.

[26] N. Bouguila and R. I. Hammoud. Color texture classification by a discrete statistical model and feature selection. In *Proc. of the IEEE International Conference on Image Processing (ICIP)*, pages 1381–1384, 2009.

[27] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1169–1176. IEEE, 2009.

[28] James E Mosimann. On the compound multinomial distribution, the multivariate β-distribution, and correlations among proportions. *Biometrika*, 49(1/2):65–82, 1962.

[29] Rasmus E Madsen, David Kauchak, and Charles Elkan. Modeling word burstiness using the dirichlet distribution. In *Proceedings of the 22nd international conference on Machine learning*, pages 545–552. ACM, 2005.

[30] Yulong Wang, Yuan Yan Tang, Luoqing Li, and Xianwei Zheng. Block sparse representation for pattern classification: Theory, extensions and applications. *Pattern Recognition*, 88:198–209, 2019.

[31] Nizar Bouguila and Walid ElGuebaly. A statistical model for histogram refinement. In Vera Kurková, Roman Neruda, and Jan Koutník, editors, *Artificial Neural Networks - ICANN 2008 , 18th International Conference, Prague, Czech Republic, September 3-6, 2008, Proceedings, Part I*, volume 5163 of *Lecture Notes in Computer Science*, pages 837–846. Springer, 2008.

[32] Jason D Rennie, Lawrence Shih, Jaime Teevan, and David R Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 616–623, 2003.

[33] Matthew A Etterson, Gerald J Niemi, and Nicholas P Danz. Estimating the effects of detection heterogeneity and overdispersion on trends estimated from avian point counts. *Ecological Applications*, 19(8):2049–2066, 2009.

[34] Andreas Lindén and Samu Mäntyniemi. Using the negative binomial distribution to model overdispersion in ecological count data. *Ecology*, 92(7):1414–1421, 2011.

[35] Celestin C Kokonendji. Over-and underdispersion models. *Methods and Applications of Statistics in Clinical Trials*, 2:506–526, 2014.

[36] Nizar Bouguila. A liouville-based approach for discrete data categorization. In Sergei O. Kuznetsov, Dominik Slezak, Daryl H. Hepting, and Boris G. Mirkin, editors, *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing - 13th International Conference, RSFDGrC 2011, Moscow, Russia, June 25-27, 2011. Proceedings*, volume 6743 of *Lecture Notes in Computer Science*, pages 330–337. Springer, 2011.

[37] Nizar Bouguila and Mukti Nath Ghimire. Discrete visual features modeling via leave-one-out likelihood estimation and applications. *J. Vis. Commun. Image Represent.*, 21(7):613–626, 2010.

[38] Dimitris Margaritis and Sebastian Thrun. A bayesian multiresolution independence test for continuous variables. *arXiv preprint arXiv:1301.2292*, 2013.

[39] John BS Haldane. The fitting of binomial distributions. *Annals of Eugenics*, 11(1):179–181, 1941.

[40] Norman TJ Bailey. The mathematical theory of epidemics. Technical report, 1957.

[41] DA Griffiths. Maximum likelihood estimation for the beta-binomial distribution and an application to the household distribution of the total number of cases of a disease. *Biometrics*, pages 637–648, 1973.

[42] Jorge G Morel and Neerchal K Nagaraj. A finite mixture distribution for modelling multinomial extra variation. *Biometrika*, 80(2):363–371, 1993.

[43] Nagaraj K Neerchal and Jorge G Morel. Large cluster results for two parametric multinomial extra variation models. *Journal of the American Statistical Association*, 93(443):1078–1087, 1998.

[44] Nagaraj K Neerchal and Jorge G Morel. An improved method for the computation of maximum likeliood estimates for multinomial overdispersion models. *Computational Statistics & Data Analysis*, 49(1):33–43, 2005.

[45] Kenneth Lange. *MM optimization algorithms*, volume 147. SIAM, 2016.

[46] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[47] Tong Tong Wu, Kenneth Lange, et al. The mm alternative to em. *Statistical Science*, 25(4):492–505, 2010.

[48] Hua Zhou and Yiwen Zhang. Em vs mm: A case study. *Computational statistics & data analysis*, 56(12):3909–3920, 2012.

[49] Andrew M Raim, Minglei Liu, Nagaraj K Neerchal, and Jorge G Morel. On the method of approximate fisher scoring for finite mixtures of multinomials. *Statistical Methodology*, 18:115–130, 2014.

[50] David R Hunter and Kenneth Lange. A tutorial on mm algorithms. *The American Statistician*, 58(1):30–37, 2004.

[51] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[52] Hua Zhou and Kenneth Lange. Mm algorithms for some discrete multivariate distributions. *Journal of Computational and Graphical Statistics*, 19(3):645–665, 2010.

[53] C Wallace and D Boulton. An information measure for classification comput. *J*, 11(2):185–194, 1968.

[54] Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[55] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[56] Christophe Andrieu, PM Djurić, and Arnaud Doucet. Model selection by mcmc computation. *Signal Processing*, 81(1):19–37, 2001.

[57] Sarunas J Raudys and Anil K. Jain. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (3):252–264, 1991.

[58] Nizar Bouguila and Djemel 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.

[59] Nizar Bouguila and Djemel Ziou. High-dimensional unsupervised selection and estimation of a finite generalized dirichlet mixture model based on minimum message length. *IEEE transactions on pattern analysis and machine intelligence*, 29(10):1716–1731, 2007.

[60] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[61] Chris S Wallace. Classification by minimum-message-length inference. In *International Conference on Computing and Information*, pages 72–81. Springer, 1990.

[62] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

[63] Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. *http://www. cs. cmu. edu/mccallum/bow/*, 1996.

[64] Michel Valstar and Maja Pantic. Induced disgust, happiness and surprise: an addition to the mmi facial expression database. In *Proc. 3rd Intern. Workshop on EMOTION (satellite of LREC): Corpora for Research on Emotion and Affect*, page 65. Paris, France, 2010.

[65] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 ieee computer society conference on computer vision and pattern recognition-workshops*, pages 94–101. IEEE, 2010.

[66] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36. IEEE, 2004.

[67] Yang Wang and Greg Mori. Human action recognition by semilatent topic models. *IEEE transactions on pattern analysis and machine intelligence*, 31(10):1762–1774, 2009.

[68] Chris S Wallace and Peter R Freeman. Estimation and inference by compact coding. *Journal of the Royal Statistical Society: Series B (Methodological)*, 49(3):240–252, 1987.

[69] Nizar Bouguila and Djemel Ziou. Online clustering via finite mixtures of dirichlet and minimum message length. *Engineering Applications of Artificial Intelligence*, 19(4):371–379, 2006.

[70] D Michael Titterington, Adrian FM Smith, and Udi E Makov. *Statistical analysis of finite mixture distributions*. Wiley,, 1985.

[71] José M Bernardo and Adrian FM Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009.

[72] Jian-Feng Yao. On recursive estimation in incomplete data models. *Statistics: A Journal of Theoretical and Applied Statistics*, 34(1):27–51, 2000.

[73] Nuha Zamzami, Manar Amayri, Nizar Bouguila, and Stephane Ploix. Online clustering for estimating occupancy in an office setting. In *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, pages 2195–2200. IEEE, 2019.

[74] Victor M Corman, Olfert Landt, Marco Kaiser, Richard Molenkamp, Adam Meijer, Daniel KW Chu, Tobias Bleicker, Sebastian Brünink, Julia Schneider, Marie Luisa Schmidt, et al. Detection of 2019 novel coronavirus (2019-ncov) by real-time rt-pcr. *Eurosurveillance*, 25(3):2000045, 2020.

[75] Nizar Bouguila. Spatial color image databases summarization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2007, Honolulu, Hawaii, USA, April 15-20, 2007*, pages 953–956. IEEE, 2007.