Advances in Finite Mixture Models with Applications to Unsupervised Learning

Samyajoy Pal

München 2025

Advances in Finite Mixture Models with Applications to Unsupervised Learning

Samyajoy Pal

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Samyajoy Pal
aus Kolkata, India

München, den 11.02.2025

Erster Berichterstatter: Prof. Dr. Christian Heumann
Zweiter Berichterstatter: Prof. Dr. Jochen Einbeck
Dritter Berichterstatter: Prof. Shalabh Shalabh, Ph.D.

Tag der Disputation: 26.05.2025

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12. Juli 2011, §8 Abs. 2 Nr. 5)

Hiermit versichere ich an Eides statt, dass die Dissertation von mir selbstständig und ohne unerlaubte Beihilfe angefertigt ist.

München, den 11.02.2025

Samyajoy Pal

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Dr. Christian Heumann, for his unwavering support and guidance throughout my PhD journey. His expertise in Statistics has been an invaluable source of knowledge and inspiration. I am particularly grateful for the trust he placed in me, allowing me complete independence to explore research problems and develop solutions. This level of freedom and encouragement has been instrumental in shaping my early research career. Through the many ups and downs of this journey, his steadfast support has been a source of immense motivation, and for that, I am truly grateful.

I would also like to extend my sincere thanks to Prof. Dr. Jochen Einbeck for graciously agreeing to review my thesis. I had the privilege of discussing my research with him in great detail, and his expert insights and constructive feedback have significantly strengthened my work, providing me with new perspectives and ideas for future research. My heartfelt thanks also go to Prof. Shalabh Shalabh, Ph.D., for serving as a reviewer of my thesis. His expert opinion and comments mean a great deal to me. Furthermore, I am grateful to Prof. Dr. Michael Schomaker and Prof. Dr. Volker Schmid for being part of my doctoral committee and for their valuable time and expertise.

I would like to express my deepest respect and gratitude to all my teachers, from my school days to my university years, who have shaped my academic journey and enabled me to reach where I am today. In particular, I extend my heartfelt thanks to Mr. Ashok Panda and Dr. Partho Sarathi Chakrabarti for introducing me to the field of Statistics and imparting invaluable knowledge. Though I have absorbed only a fraction of their profound wisdom, their influence has been pivotal in my academic growth. I am also immensely grateful to my Master's supervisors, Dr. M. Subbiah and Prof. Dr. M.R. Srinivasan, for giving me my first opportunities in research and guiding me towards a career in Statistics.

I am forever indebted to my family—especially my parents—for their unconditional love, support, and encouragement. My parents have made immense sacrifices to ensure my success, and I am profoundly grateful for their endless dedication. A heartfelt thanks to Dayasri, who has been so much more than a companion throughout this journey. As the first reviewer, toughest critic, and a truly invaluable collaborator, her constant support has meant the world. This research would not have been possible without her encouragement every step of the way.

Finally, as researchers, we are devoted to the pursuit of knowledge, seeking to uncover the fundamental truths of the universe. I humbly dedicate this work and my life to the relentless quest for real, limitless knowledge.

Summary

This research work focuses on advancing the theory and application of finite mixture models, specifically through the development and modification of algorithms for clustering and parameter estimation in various complex data scenarios. The overarching theme is to enhance computational efficiency, accuracy, flexibility and interpretability within mixture models, particularly where traditional methods fall short.

Our first contribution explores a novel mixture model-based approach for compositional data, often challenging to analyze due to constraints like unit-sum requirements. Traditional methods rely on data transformation, but we leverage a Dirichlet Mixture Model (DMM) with a modified Hard EM algorithm to address issues related to rapid convergence and empty clusters. Through rigorous simulation studies, we compare this model against popular clustering methods (such as KMeans, PAM, and DBSCAN) and show that it provides robust clustering performance across diverse scenarios, including two real-world datasets from business and physical sciences. This work highlights the model's adaptability in handling the unique distributional characteristics of compositional data.

In the second study, we tackle the challenge of estimating the Kullback-Leibler (KL) divergence in Dirichlet Mixture Models, essential for compositional data analysis. Traditional Monte Carlo-based methods for KL divergence estimation are computationally intensive, prompting our development of a variational approach with a closed-form solution. This new method enhances computational efficiency and accuracy, allowing for faster model comparisons. Validation with real and simulated data shows that this approach outperforms existing methods, paving the way for more efficient exploration of DMMs in practical applications.

The third study delved into parameter estimation within Dirichlet Mixture Models, where we introduce an alternative parametrization using mean and precision parameters. This approach offers greater interpretability, where the mean indicates location, and precision reflects the peakedness of the distribution. We derive maximum likelihood estimates (MLEs) for various scenarios using the EM algorithm and introduce novel solutions to address high-dimensional data challenges. This includes employing Stirling's approximation and moment approximation to provide closed-form solutions, ultimately enhancing both the computational speed and robustness of parameter estimation. This work demonstrates the identifiability of the DMM and proposes a closed-form KL divergence approximation for goodness-of-fit evaluation, validated on simulated and real data.

In the fourth study, We revisit the Hard EM algorithm, commonly employed for unsupervised learning due to its computational simplicity. Despite its perceived limitations, such as biased estimates and lack of consistency, we propose modifications to Hard EM tailored for Gaussian Mixture Models to address convergence issues. Using extensive sim-

ulations across multiple benchmark datasets, we demonstrate that Hard EM can perform comparably, and sometimes favorably, to standard EM in terms of accuracy and efficiency. Additionally, the modified Hard EM algorithm is tested on real biological datasets, where it shows practical utility in cluster analysis.

Lastly, we extend the flexibility of mixture models by constructing mixtures that combine both identical and non-identical multivariate distributions, such as the Multivariate Skew Normal and Multivariate Generalized Hyperbolic distributions. This innovative framework broadens the applicability of mixture models by enabling combinations across diverse distributional types, which conventional models typically do not allow. The proposed framework encompasses traditional mixture models as special cases, showcasing its utility through applications on simulated and real datasets. The flexibility of this model allows for more accurate pattern recognition and parameter estimation, proving its versatility across varied data structures.

In summary, our research contributes significantly to the field of statistical modeling by expanding the versatility and applicability of finite mixture models, enhancing computational techniques, and improving interpretability in complex data environments. This work has practical implications across fields such as biological sciences, business, and marketing, enabling more accurate and efficient data analysis in real-world applications.

Zusammenfassung

Diese Forschungsarbeit konzentriert sich auf die Weiterentwicklung der Theorie und Anwendung endlicher Mischmodelle, insbesondere durch die Entwicklung und Modifikation von Algorithmen für die Clusterbildung und Parameterschätzung in verschiedenen komplexen Datenszenarien. Das übergeordnete Ziel ist es, die Effizienz, Genauigkeit und Interpretierbarkeit von Mischmodellen zu verbessern, insbesondere in Fällen, in denen herkömmliche Methoden an ihre Grenzen stoßen.

Unser erster Beitrag untersucht einen neuartigen Ansatz für die Analyse von Kompositionsdaten basierend auf Mischmodellen, die aufgrund der Restriktion, dass sich die Anteile zu Eins aufsummieren müssen, oft schwierig zu analysieren sind. Traditionelle Methoden stützen sich auf Datenumwandlungen, aber wir nutzen ein Dirichlet-Mischmodell (DMM) mit einem modifizierten Hard EM-Algorithmus, um Probleme im Zusammenhang mit schneller Konvergenz und leeren Clustern zu bewältigen. Durch umfangreiche Simulationsstudien vergleichen wir dieses Modell mit populären Clustermethoden (wie KMeans, PAM und DBSCAN) und zeigen, dass es in verschiedenen Szenarien, einschließlich zweier realer Datensätze aus der Unternehmens- und Naturwissenschaft, eine robuste Leistung bei der Clusterbildung bietet. Diese Arbeit unterstreicht die Anpassungsfähigkeit des Modells an die spezifischen Verteilungseigenschaften von Kompositionsdaten.

In der zweiten Studie stellen wir uns der Herausforderung, die Kullback-Leibler (KL)-Divergenz in Dirichlet-Mischmodellen zu schätzen, was für die Analyse von Kompositionsdaten entscheidend ist. Da traditionelle Monte-Carlo-Methoden zur Schätzung der KL-Divergenz rechnerisch aufwendig sind, haben wir einen neuen variationalen Ansatz mit einer geschlossenen Lösung entwickelt. Diese Methode verbessert die rechnerische Effizienz und Genauigkeit, was schnellere Modellvergleiche ermöglicht. Die Validierung an realen und simulierten Daten zeigt, dass dieser Ansatz bestehende Methoden übertrifft und somit den Weg für eine effizientere Untersuchung von DMMs in der Praxis ebnet.

Die dritte Studie widmet sich der Parameterschätzung innerhalb von Dirichlet-Mischmodellen, wobei wir eine alternative Parametrisierung mit Mittelwert- und Präzisionsparametern einführen. Dieser Ansatz bietet eine größere Interpretierbarkeit, da der Mittelwert die Lage und die Präzision die Konzentration der Verteilung um den Mittelwert widerspiegelt. Wir leiten Maximum-Likelihood-Schätzungen (MLE) für verschiedene Szenarien mithilfe des EM-Algorithmus ab und stellen neuartige Lösungen für Herausforderungen in hochdimensionalen Daten vor. Dazu gehört die Verwendung der Stirlingschen Näherung und der Momentenapproximation zur Bereitstellung geschlossener Lösungen, die sowohl die Rechengeschwindigkeit als auch die Robustheit der Parameterschätzung erhöhen. Diese Arbeit zeigt die Identifizierbarkeit des DMM und schlägt eine geschlossene Approximation für die KL-Divergenz zur Gütebewertung vor, die an simulierten und realen Daten validiert wurde.

In der vierten Studie wird der Hard EM-Algorithmus erneut untersucht, der aufgrund seiner rechnerischen Einfachheit häufig für unüberwachtes Lernen eingesetzt wird. Trotz seiner bekannten Einschränkungen, wie verzerrten Schätzungen und mangelnder Konsistenz, schlagen wir Modifikationen des Hard EM-Algorithmus vor, die speziell für Gaußsche Mischmodelle entwickelt wurden, um Konvergenzprobleme zu adressieren. Durch umfangreiche Simulationen auf verschiedenen Benchmark-Datensätzen zeigen wir, dass der Hard EM-Algorithmus in Bezug auf Genauigkeit und Effizienz vergleichbare, teils sogar bessere Ergebnisse als der Standard-EM-Algorithmus liefern kann. Darüber hinaus wird der modifizierte Hard EM-Algorithmus an realen biologischen Datensätzen getestet und zeigt praktische Nützlichkeit bei der Clusteranalyse.

Schließlich erweitern wir die Flexibilität von Mischmodellen, indem wir Mischungen konstruieren, die sowohl identische als auch nicht identische multivariate Verteilungen, wie die multivariate schiefe Normalverteilung und die multivariate verallgemeinerte hyperbolische Verteilung, kombinieren. Dieses innovative Rahmenwerk vergrößert die Anwendbarkeit von Mischmodellen, indem es Kombinationen unterschiedlicher Verteilungstypen ermöglicht, die herkömmliche Modelle typischerweise nicht erlauben. Das vorgeschlagene Rahmenwerk umfasst traditionelle Mischmodelle als Spezialfälle und zeigt seine Nützlichkeit durch Anwendungen auf simulierte und reale Datensätze. Die Flexibilität dieses Modells ermöglicht eine genauere Mustererkennung und Parameterschätzung und beweist seine Vielseitigkeit in verschiedenen Datenstrukturen.

Zusammenfassend trägt unsere Forschung maßgeblich zum Bereich der statistischen Modellierung bei, indem sie die Vielseitigkeit und Anwendbarkeit endlicher Mischmodelle erweitert, rechnerische Techniken verbessert und die Interpretierbarkeit in komplexen Datenumgebungen erhöht. Diese Arbeit hat praktische Implikationen in Bereichen wie Biowissenschaften, Wirtschaft und Marketing und ermöglicht eine genauere und effizientere Datenanalyse in realen Anwendungen.

Contents

Chapter 1

Introduction

Analyzing complex data structures with inherent clustering remains a significant challenge across various disciplines, including biology (Balaban et al., 2019; Petegrosso et al., 2020), finance (Li et al., 2021), image processing (Kim et al., 2020), and social sciences (Grimmer et al., 2021). Two main approaches dominate the field of clustering: model-based methods and algorithms based on similarity or distance measures. In model-based methods, such as the Gaussian Mixture Model (GMM) (McLachlan et al., 2019), clusters are identified by fitting a mixture of distributions to the data. In contrast, similarity-based algorithms like hierarchical clustering (Ward Jr, 1963) and K-Means (MacQueen et al., 1967) create clusters by assessing the relationships or distances between data points.

Among these methods, mixture models have become a highly versatile and widely used framework for clustering complex datasets. They allow for the detection of latent subpopulations within diverse datasets by accommodating variations in the observed data while capturing its underlying distribution. The Gaussian Mixture Model, a prime example of mixture models, assumes that each cluster's data follows a multivariate Gaussian distribution. However, the applicability of mixture models is not limited to Gaussian distributions. Increasingly, the literature emphasizes the potential of other multivariate distributions, such as the multivariate t-distribution, Multivariate Skew Normal (MSN), and Multivariate Generalized Hyperbolic (MGH) distributions, for modeling intricate data structures. Browne and McNicholas (2015) demonstrated how to model multivariate data using mixtures of Multivariate Generalized Hyperbolic distributions, while Lin (2009) and Abe et al. (2021) provided detailed methods for fitting mixtures of Multivariate Skew Normal distributions. Other special forms of Multivariate Generalized Hyperbolic distributions, including mixtures with the Multivariate Normal Inverse Gaussian (MNIG) distribution (O'Hagan et al., 2016), the Skew t distribution (Vrbik and McNicholas, 2012; Lee and McLachlan, 2014), and the Variance-Gamma (McNicholas et al., 2013) distribution, have also been explored. Cabral et al. (2012) examined a flexible class of models that include finite mixtures of multivariate skew normal independent distributions, specifically focusing on finite mixtures involving skew normal, skew t, skew slash, and skew contaminated normal distributions. Meanwhile, Zehra Doğru et al. (2021) introduced finite mixtures of multivariate skew Laplace distributions to effectively capture both skewness and heavy-tailed characteristics in heterogeneous datasets.

1.1 Research Objectives

The primary goal of this research is to advance the theoretical and practical applications of finite mixture models, with a particular emphasis on unsupervised learning and clustering. Given the complexity of real-world datasets in fields such as biology, finance, and social sciences, this work seeks to develop more flexible and interpretable models that improve computational efficiency and accuracy. Specifically, this research aims to:

1. **Develop a Novel Model-Based Clustering Approach for Compositional Data Using Dirichlet Mixture Models (DMM):**
    - Design a clustering algorithm specifically suited for compositional data, which is constrained by a unit-sum requirement, where traditional clustering methods like KMeans or GMM are less effective and often require data transformations.
    - Construct a Dirichlet Mixture Model (DMM) to naturally accommodate the unit-sum constraint inherent to compositional data, and apply a modified Hard EM algorithm for efficient parameter estimation.
    - Conduct simulation studies and real data applications to validate the proposed method's robustness and adaptability in handling compositional datasets, improving both interpretability and clustering accuracy.
2. **Propose a Variational Approach for Efficient KL Divergence Estimation in DMMs:**
    - Address the computational challenges in estimating Kullback-Leibler (KL) divergence in Dirichlet Mixture Models, a critical metric for model selection and evaluation.
    - Replace the traditionally intensive Monte Carlo-based estimation methods with a novel variational approach that yields a closed-form solution for KL divergence, thereby significantly enhancing computational efficiency.
    - Validate the variational method's accuracy and efficiency through comparisons with Monte Carlo methods, enabling faster and more reliable model evaluations for clustering compositional data.
3. **Enhance Interpretability and Estimation Efficiency in DMMs through Mean-Precision Parametrization:**
    - Develop an alternative parametrization of the Dirichlet distribution using mean and precision parameters, providing a more interpretable framework where the mean reflects the distribution's location and precision indicates its concentration.
    - Derive parameter estimates for DMMs under various scenarios, distinguishing cases where one or both parameters are known and adapting to specific data constraints.
    - Address the computational challenges of high-dimensional DMMs by proposing special estimates based on Stirling's approximation and moment approximation, yielding closed-form solutions that increase computational efficiency.
    - Prove the identifiability of Dirichlet Mixture Models, which is crucial for theoretical robustness but has not been previously discussed in depth.

4. **Evaluate and Enhance Classification EM (Hard EM) for Gaussian Mixture Models (GMM):**

    - Assess the clustering performance of the Hard EM algorithm on various benchmark datasets to investigate if its performance is genuinely inferior to the standard EM algorithm, as is commonly assumed.
    - Identify specific scenarios where Hard EM could perform comparably or even favorably due to its computational simplicity.
    - Develop tailored modifications to Hard EM to mitigate issues of greedy convergence, thereby improving stability and accuracy when applied to Gaussian Mixture Models, particularly in complex, high-dimensional data.

5. **Extend Flexibility in Mixture Models by Combining Identical and Non-Identical Distributions:**

    - Introduce a flexible framework for finite mixture models that allows combinations of identical and non-identical multivariate distributions, such as Multivariate Skew Normal, Multivariate Generalized Hyperbolic, and other complex distributions.
    - Explore model selection criteria tailored to this versatile framework to determine the optimal mixture configuration and provide diagnostics to evaluate model performance across diverse datasets.
    - Demonstrate the practical benefits of this framework in identifying latent clusters and improving parameter estimation accuracy, broadening the applicability of finite mixture models in unsupervised learning tasks across disciplines.

6. **Develop a Python Package for Software Implementation of the Proposed Models:**

    - Provide a comprehensive Python package incorporating all developed mixture models, ensuring accessibility and ease of use for practitioners and researchers.
    - Implement efficient algorithms for parameter estimation and clustering, integrating enhancements from the study to maximize computational efficiency.
    - Include visualization tools, model selection criteria, and performance diagnostics to facilitate practical application in real-world clustering tasks.
    - Encourage broader adoption by ensuring compatibility with popular Python libraries such as NumPy, SciPy, and scikit-learn, and by providing insightful examples with motivating datasets.

1.2 Overview of Contributing Papers

Chapter 2. Pal, Samyajoy, and Christian Heumann. "Clustering compositional data using Dirichlet mixture model." *Plos one 17, no. 5 (2022): e0268438.* https://doi.org/10.1371/journal.pone.0268438.

This chapter aligns with Research Objective 1, where we introduce a Dirichlet Mixture Model (DMM) for compositional data clustering. We demonstrate how the model naturally accommodates the unit-sum constraint and propose an adaptation of the Hard

EM algorithm for efficient parameter estimation. Simulation studies and real-world applications validate the model's performance.

Chapter 3. Pal, Samyajoy, and Christian Heumann. "Gene coexpression analysis with Dirichlet mixture model: accelerating model evaluation through closed-form KL divergence approximation using variational techniques." In *International Workshop on Statistical Modelling*, pp. 134-141. Cham: Springer Nature Switzerland, 2024. https://doi.org/10.1007/978-3-031-65723-8_21.

This chapter corresponds to Research Objective 2, where we focus on efficient estimation of the Kullback-Leibler (KL) divergence for Dirichlet Mixture Models. We develop a novel variational approach that yields a closed-form KL divergence approximation, significantly reducing computational costs and enhancing model evaluation for compositional data.

Chapter 4. Pal, Samyajoy, and Christian Heumann. "Revisiting Dirichlet Mixture Model: Unraveling Deeper Insights and Practical Applications." *Statistical Papers* 66, no. 1 (2025): 1-38. https://doi.org/10.1007/s00362-024-01627-0.

This chapter contributes to Research Objective 3 by introducing a mean-precision parametrization of the Dirichlet distribution for improved interpretability. We derive efficient parameter estimation techniques, discuss identifiability issues, and propose approximations that enhance the computational efficiency of DMMs, particularly in high-dimensional settings.

Chapter 5. Pal, Samyajoy, and Christian Heumann. "Gaussian mixture model with modified hard EM algorithm in clustering problems." In *Statistical Modeling and Applications on Real-Time Problems*, pp. 153-179. CRC Press, 2024. https://doi.org/10.1201/9781003356653-7.

This chapter addresses Research Objective 4, where we analyze the performance of the Hard EM algorithm for Gaussian Mixture Models (GMM) and propose modifications to improve convergence and clustering accuracy. Extensive simulations and real data applications illustrate the effectiveness of the modified Hard EM algorithm.

Chapter 6. Pal, Samyajoy, and Christian Heumann. "Flexible Multivariate Mixture Models: A Comprehensive Approach for Modeling Mixtures of Non-Identical Distributions." *International Statistical Review* (2024). https://doi.org/10.1111/insr.12593.

This chapter aligns with Research Objective 5, where we extend traditional mixture models by allowing combinations of identical and non-identical distributions, including the Multivariate Skew Normal and Multivariate Generalized Hyperbolic distributions. We propose a unified framework that broadens the applicability of finite mixture models and demonstrate its effectiveness on diverse datasets.

In addition to the above chapters, this dissertation includes a software implementation of the developed models in a Python package, addressing Research Objective 6. The package provides efficient algorithms for parameter estimation and clustering, along with visualization tools and model selection criteria to facilitate practical applications in unsuper-

vised learning. The package is available at https://github.com/samyajoypal/fmvmm.

1.3 Brief History of Finite Mixture Model

Finite mixture models have a rich history spanning nearly 140 years. The historical evolution and significant advancements in finite mixture models have been comprehensively reviewed by McLachlan et al. (2019) and McLachlan (2000). One of the first significant analyses using mixture models was conducted by Karl Pearson, a renowned biometrician. In his seminal 1894 paper (Pearson, 1894), Pearson—who was also a statistician and eugenicist—applied a mixture of two normal probability density functions, each with distinct means (μ_1 and μ_2) and variances (σ_1^2 and σ_2^2), to a set of crab data provided by evolutionary biologist Weldon (1892, 1894). The concept of breaking down a normal mixture into its individual components had been suggested in the earlier works of Quetelet (1846, 1852) and explicitly mentioned by Galton (1891). For a detailed overview of these early contributions to mixture models, see Stigler (1990). Another early contribution came from Holmes (1892), who introduced the idea of population mixtures by arguing that an average was insufficient when considering wealth inequality. Prior to Pearson's work, Newcomb (1886) proposed an iterative reweighting method, which can be viewed as an early application of the EM algorithm later formalized by Dempster et al. (1977), to calculate the common mean of a mixture of univariate normal distributions with known variances. For more detailed discussions on the history of mixture models, refer to McLachlan and Basford (1988) and McLachlan (2000).

However, aside from contributions by Jeffreys (1932) and Rao (1948), the use of maximum likelihood (ML) for fitting mixture models did not gain much traction until the 1960s. Important work on an iterative Maximum Likelihood (ML) fitting scheme for mixture distributions was published by Day (1969) and Wolfe (1970), who also authored several technical reports. The formalization of this iterative approach in a broader context by Dempster et al. (1977) through their Expectation Maximization (EM) algorithm marked a significant theoretical advancement in understanding the convergence properties of the ML solution for mixture problems. The EM algorithm spurred renewed research interest in finite mixture models, leading to a surge of subsequent papers in the field, beginning with works like Ganesalingam and McLachlan (1978) and O'neill (1978).

1.4 EM Algorithm

The Expectation-Maximization (EM) algorithm Dempster et al. (1977) is a widely used iterative method for finding maximum likelihood estimates (MLE) in models with latent variables. It alternates between two steps: the Expectation (E)-step, where the expected complete data log-likelihood is obtained, and the Maximization (M)-step, where the parameters are updated to maximize this log-likelihood. The EM algorithm is particularly useful when dealing with incomplete or hidden data.

1.4.1 General Formulation of EM Algorithm

Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N$ denote an independent random sample of size N, where $\boldsymbol{X}_i$ is a p dimensional random vector with probability density function (p.d.f.) $f(\boldsymbol{x}_i; \boldsymbol{\theta})$ on $\mathbb{R}^p$, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)^T$ is a vector of unknown parameters with parameter space $\boldsymbol{\Theta}$. We can

write $\boldsymbol{X} = (\boldsymbol{X}_1^T, \ldots, \boldsymbol{X}_N^T)^T$, where the superscript T denotes vector transpose. An observed random sample is denoted by $\boldsymbol{x} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_N^T)^T$, where $\boldsymbol{x}_i$ is the observed value of the random vector $\boldsymbol{X}_i$.

In general, the maximum likelihood estimate of $\boldsymbol{\theta}$ is obtained by maximizing the log-likelihood function. Let us denote the likelihood function by $L(\boldsymbol{\theta}; \boldsymbol{x})$. Then the MLE is given by, $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}}\, l(\boldsymbol{\theta})$ (Casella and Berger, 2024), where

$$l(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta}; \boldsymbol{x}) = \sum_{i=1}^{N} \log f(\boldsymbol{x}_i; \boldsymbol{\theta}) \ . \tag{1.1}$$

The Expectation-Maximization (EM) algorithm is a versatile method that provides an iterative approach for computing maximum likelihood estimates (MLEs) in scenarios where MLE computation would be simple if additional data were available (McLachlan and Krishnan, 2008). In this context, the observed data vector $\boldsymbol{x}$ is considered incomplete and viewed as a function of the so-called complete data. The concept of "incomplete data" encompasses both cases of missing data in the traditional sense and situations where complete data represent outcomes from a hypothetical experiment. In the latter case, the complete data may include variables that are never directly observable. Under this framework, we use $\boldsymbol{y} = (\boldsymbol{x}, \boldsymbol{z})$ to represent the complete or augmented data, where $\boldsymbol{z} = (\boldsymbol{z}_1, \ldots \boldsymbol{z}_N)$ denotes the unobservable or missing data.

Let, $f_c(\boldsymbol{y}; \boldsymbol{\theta})$ be the p.d.f. of the random vector $\boldsymbol{Y}$ corresponding to the complete data vector $\boldsymbol{y}$. Then the complete data log-likelihood is given by,

$$l_c(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta}; \boldsymbol{y}) \ .$$

The EM algorithm approaches the problem of solving the incomplete data log-likelihood equation (1.1) indirectly by proceeding iteratively in terms of the complete data log-likelihood function, $l_c(\boldsymbol{\theta})$. As it is unobservable, it is replaced by its conditional expectation given $\boldsymbol{x}$ using the current parameter values for $\boldsymbol{\theta}$. The algorithm alternates between the following steps:

E-step: In the E-step, we compute the expectation of the complete data log-likelihood with respect to the conditional distribution of the latent variables given the observed data and current parameter estimates $\boldsymbol{\theta}^{(t)}$:

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\boldsymbol{\theta}^{(t)}}[l_c(\boldsymbol{\theta}) \mid \boldsymbol{x}] \ , \tag{1.2}$$

where the Q function is the expected complete data log-likelihood function.

M-step: In the M-step, we maximize the expected log-likelihood from the E-step with respect to the model parameters:

$$\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}}\, Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) \tag{1.3}$$

The algorithm repeats these two steps until convergence, typically when the change in the log-likelihood between iterations falls below a certain threshold.

1.4.2 Monotonicity of EM Algorithm

One of the most important properties of the EM algorithm is that it ensures that the likelihood is non-decreasing at each iteration.

Theorem 1.4.1 *Given a sequence of parameters $\boldsymbol{\theta}^t$, $\boldsymbol{\theta}^{t+1}$ generated by the EM steps t and $t+1$, the incomplete data log-likelihood at each step is non-decreasing, that is,*

$$l(\boldsymbol{\theta}^{t+1}) \geq l(\boldsymbol{\theta}^t) \ . \tag{1.4}$$

Proof The proof follows the approach established by Dempster et al. (1977). We denote the conditional p.d.f. of $\boldsymbol{Y}$ given $\boldsymbol{x}$ by $g(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta})$.

It implies that,

$$g(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta}) = g(\boldsymbol{z} \mid \boldsymbol{x}; \boldsymbol{\theta}) = \frac{f_c(\boldsymbol{y}; \boldsymbol{\theta})}{f(\boldsymbol{x}; \boldsymbol{\theta})} \tag{1.5}$$

Then the log-likelihood is given by,

$$\begin{aligned} \log L(\boldsymbol{\theta} \mid \boldsymbol{x}) &= \log f(\boldsymbol{x}; \boldsymbol{\theta}) \\ &= \log f_c(\boldsymbol{y}; \boldsymbol{\theta}) - \log g(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta}) \\ &= \log L_c(\boldsymbol{\theta}) - \log g(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta}) \ . \end{aligned} \tag{1.6}$$

On taking the expectations of both sides of the above equation with respect to the conditional distribution of $\boldsymbol{Z}$ given $\boldsymbol{X} = x$ using the fit $\boldsymbol{\theta}^{(t)}$ for $\boldsymbol{\theta}$, we have that,

$$\begin{aligned} \log L(\boldsymbol{\theta} \mid \boldsymbol{x}) &= \mathbb{E}_{\boldsymbol{\theta}^{(t)}} \left[\log L_c(\boldsymbol{\theta}) \mid \boldsymbol{x} \right] - \mathbb{E}_{\boldsymbol{\theta}^{(t)}} \left[\log g(\boldsymbol{Y} \mid \boldsymbol{x}; \boldsymbol{\theta}) \mid \boldsymbol{x} \right] \\ &= Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) - H(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) \end{aligned} \tag{1.7}$$

where,

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\boldsymbol{\theta}^{(t)})} \left[\log L_c(\boldsymbol{\theta}) \mid \boldsymbol{x} \right] \text{ and} \tag{1.8}$$

$$H(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) = \mathbb{E}_{g(\boldsymbol{\theta}^{(t)})} \left[\log g(\boldsymbol{Y} \mid \boldsymbol{x}; \boldsymbol{\theta}) \mid \boldsymbol{x} \right] \tag{1.9}$$

From eq. (1.7), we have that,

$$\begin{aligned} &\log L(\boldsymbol{\theta}^{(t+1)}) - \log L(\boldsymbol{\theta}^{(t)}) \\ &= \left[Q(\boldsymbol{\theta}^{(t+1)}; \boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)}; \boldsymbol{\theta}^{(t)}) \right] \\ &- \left[H(\boldsymbol{\theta}^{(t+1)}; \boldsymbol{\theta}^{(t)}) - H(\boldsymbol{\theta}^{(t)}; \boldsymbol{\theta}^{(t)}) \right] \end{aligned} \tag{1.10}$$

The first difference on the right-hand side is non-negative since $\boldsymbol{\theta}^{(t+1)}$ is chosen so that

$$Q(\boldsymbol{\theta}^{(t+1)}; \boldsymbol{\theta}^{(t)}) \geq Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}), \quad \forall \boldsymbol{\theta} \in \boldsymbol{\Theta}.$$

Hence eq. (1.4) holds if the second difference on the right-hand side is nonpositive; that is, if

$$H(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)}) - H(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) \leq 0 \ . \tag{1.11}$$

Now for any $\boldsymbol{\theta}$,

$$\begin{aligned} &H(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) - H(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) \\ &= \mathbb{E}_{\boldsymbol{\theta}^{(t)})} \left[\log \frac{g(\boldsymbol{Y} \mid \boldsymbol{x}; \boldsymbol{\theta})}{g(\boldsymbol{Y} \mid \boldsymbol{x}; \boldsymbol{\theta}^{(t)})} \mid \boldsymbol{x} \right] \ . \end{aligned} \tag{1.12}$$

Now by Jensen's inequality (Durrett, 2019),

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\theta}^{(t)})}\left[\log \frac{g(\boldsymbol{Y} \mid \boldsymbol{x}; \boldsymbol{\theta})}{g(\boldsymbol{Y} \mid \boldsymbol{x}; \boldsymbol{\theta}^{(t)})} \mid \boldsymbol{x}\right] \\
&\leq \log \mathbb{E}_{\boldsymbol{\theta}^{(t)})}\left[\frac{g(\boldsymbol{Y} \mid \boldsymbol{x}; \boldsymbol{\theta})}{g(\boldsymbol{Y} \mid \boldsymbol{x}; \boldsymbol{\theta}^{(t)})} \mid \boldsymbol{x}\right] && (1.13) \\
&= \log \int_{\mathcal{Y}(\boldsymbol{x})} g(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta}^{(t)}) d\boldsymbol{y} && (1.14) \\
&= \log 1 && (1.15) \\
&= 0 && (1.16)
\end{aligned}$$

Hence, it is proved that,

$$l(\boldsymbol{\theta}^{(t+1)}) \geq l(\boldsymbol{\theta}^{(t)}) \ . \tag{1.17}$$

∎

1.4.3 Advantages and Criticisms of the EM Algorithm

The EM algorithm offers several advantages over other iterative methods (McLachlan and Krishnan, 2008), such as the Newton-Raphson and Fisher's scoring algorithms, which are traditionally used to find maximum likelihood estimates (MLEs). Some of the key benefits of the EM algorithm include:

1. The EM algorithm exhibits numerical stability. Each iteration guarantees an increase in the likelihood function.

2. The algorithm demonstrates robust global convergence under broad conditions. It generally converges to a local maximum even when starting from arbitrary points in the parameter space, assuming that the initial parameter values are not severely ill-chosen or affected by some pathological feature of the log-likelihood function.

3. The EM algorithm is relatively simple to implement. This simplicity arises from the fact that each iteration's E-step involves taking expectations over complete-data conditional distributions, and the M-step requires maximum likelihood estimation (MLE) for the complete data, which often has closed-form solutions.

4. It is computationally efficient since the algorithm avoids directly computing the likelihood function or its derivatives. As a result, the EM algorithm is easy to program.

5. The algorithm has low memory requirements, making it feasible to run on machines with limited computational resources. For example, the algorithm does not need to store the information matrix or its inverse at any stage of the iterations.

6. In cases where the complete-data problem can be handled using standard statistical software, the M-step can often be completed using these tools. When the MLE of the complete data does not exist in closed form, extensions like the Generalized EM (GEM) or the Expectation-Conditional Maximization (ECM) algorithms can be used. These methods allow the M-step to be solved iteratively and still maintain the stable, monotone convergence of the EM algorithm.

7. The EM algorithm reduces the complexity of analytical work. It only requires maximizing the conditional expectation of the log-likelihood function for the complete-data problem. While the E-step may involve some analytical effort, it tends to be straightforward for many practical problems.

8. Although the EM algorithm may require more iterations compared to other methods, each iteration is computationally inexpensive, which often compensates for the higher iteration count.

9. Monitoring the progress of the algorithm is straightforward. By observing the monotonic increase in the likelihood over iterations (if its evaluation is manageable), one can easily detect convergence and debug potential errors in the implementation.

10. The EM algorithm can also provide estimates of the missing data, which is particularly useful in certain applications.

Despite its numerous advantages, the EM algorithm has certain limitations:

1. Unlike Fisher's scoring method, the EM algorithm does not intrinsically provide an estimate of the covariance matrix for the parameter estimates. However, this drawback can be mitigated by using appropriate extensions or modifications associated with the EM algorithm.

2. In some cases, the EM algorithm can exhibit slow convergence, particularly in problems that involve excessive amounts of incomplete information or have relatively simple structures.

3. Like Newton-type methods, the EM algorithm does not guarantee convergence to the global maximum when the likelihood function has multiple maxima. Consequently, the final estimate depends on the initial parameter values. It is worth noting, however, that this issue is not unique to the EM algorithm—most optimization procedures share this limitation. More advanced techniques, such as simulated annealing, can address these challenges, though they are often complex to implement.

4. In certain scenarios, the E-step of the EM algorithm may be analytically intractable. In such cases, Monte Carlo methods can be used to approximate the required expectations, though these techniques introduce additional complexity and increase the run time.

1.4.4 ECM Algorithm

The Expectation Conditional Maximization (ECM) algorithm (Meng and Rubin, 1993) is an extension of the EM algorithm that replaces the single M-step with multiple conditional maximization (CM) steps. This allows for greater flexibility in optimizing complex models, where the M-step may not be easy to compute directly. In ECM, the parameter updates are performed conditionally on subsets of the parameters, which can lead to faster convergence.

The ECM algorithm, like the EM algorithm, begins with an E-step, where we compute the expected value of the complete data log-likelihood. However, instead of a single M-step, ECM divides the M-step into multiple conditional maximization steps.

Let $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_d)$ represent the partition of the model parameters into d subsets. The algorithm alternates between the E-step and the following CM steps:

E-step: Compute the expected complete data log-likelihood:

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\boldsymbol{\theta}^{(t)}}[l_c(\boldsymbol{\theta}) \mid \boldsymbol{x}] \tag{1.18}$$

CM-step 1: Maximize the expected log-likelihood with respect to the first subset of parameters $\boldsymbol{\theta}_1$, keeping the others fixed:

$$\boldsymbol{\theta}_1^{(t+1)} = \arg\max_{\boldsymbol{\theta}_1} Q(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2^{(t)}, \ldots, \boldsymbol{\theta}_d^{(t)} \mid \boldsymbol{\theta}^{(t)}) \tag{1.19}$$

CM-step 2: Maximize with respect to the second subset of parameters $\boldsymbol{\theta}_2$, keeping the rest fixed:

$$\boldsymbol{\theta}_2^{(t+1)} = \arg\max_{\boldsymbol{\theta}_2} Q(\boldsymbol{\theta}_1^{(t+1)}, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_d^{(t)} \mid \boldsymbol{\theta}^{(t)}) \tag{1.20}$$

This process is repeated for all d subsets of parameters until all have been updated.

1.5 General Formulation of Finite Mixture Model

Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N$ denote a random sample of size N, where $\boldsymbol{X}_i$ is a p dimensional random vector with probability density function $f(\boldsymbol{x}_i \mid \boldsymbol{\alpha})$ on $\mathbb{R}^p$, where $\boldsymbol{\alpha} \in \mathbb{R}^d$ is the distribution parameter. An observed random sample is denoted by $\boldsymbol{x} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_N^T)^T$, where $\boldsymbol{x}_i$ is the observed value of the random vector $\boldsymbol{X}_i$.

The density of a mixture model with k mixture components for one observation $\boldsymbol{x}_i$ is given by the mixture density

$$p(\boldsymbol{x}_i \mid \boldsymbol{\alpha}) = \sum_{j=1}^{k} \pi_j f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j) \ , \tag{1.21}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$ contains the corresponding mixture proportions with $\sum_{j=1}^{k} \pi_j = 1$, $0 < \pi_j < 1$. The density component of mixture j is given by $f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j)$ and $\boldsymbol{\alpha}_j$, $j = 1, 2, ..., k$ is the vector of component specific parameters for each density. Then $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)$ denotes the vector of distribution parameters of the model and $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\alpha})$ denotes the the vector of all parameters of the model.

The log-likelihood of the model for a sample of size N is then given by

$$\log p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \mid \boldsymbol{\alpha}, \boldsymbol{\pi}) = \sum_{i=1}^{N} \log \left[\sum_{j=1}^{k} \pi_j f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j) \right] \ . \tag{1.22}$$

The parameters of the mixture model can be estimated using the EM algorithm (Dempster et al., 1977). For the E step, we introduce latent categorical variables Z_i, assuming values $1, \ldots, k$ with probabilities $\pi_1, \ldots, \pi_k$ such that $Pr(\boldsymbol{X}_i \mid Z_i = j) = f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j)$, $j = 1, \ldots, k$. The posterior probability that the data point i belongs to cluster j is

computed using Bayes rule as

$$\gamma_{ij}(\boldsymbol{x}_i) = Pr(Z_i = j \mid \boldsymbol{X}_i = \boldsymbol{x}_i, \boldsymbol{\alpha}, \boldsymbol{\pi}) = \frac{\pi_j f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j)}{\sum_{r=1}^{k} \pi_r f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_r)} \ . \tag{1.23}$$

The expected complete data log likelihood for the current iteration number t can be decomposed as follows (Murphy, 2022),

$$\begin{aligned}
Q(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{t-1}) &= \mathbb{E}\left[\sum_{i=1}^{N} \log(p(\boldsymbol{x}_i, z_i \mid \boldsymbol{\alpha})) \mid \boldsymbol{x}, \boldsymbol{\alpha}^{t-1}\right] && (1.24)\\
&= \sum_{i=1}^{N} \mathbb{E}\left[\log \left[\prod_{j=1}^{k} (\pi_j p(\boldsymbol{x_i} \mid \boldsymbol{\alpha}_j))^{\mathbb{I}(z_i=j)}\right]\right] && (1.25)\\
&= \sum_{i=1}^{N} \sum_{j=1}^{k} \mathbb{E}\left[[\mathbb{I}(z_i = j)] \log[\pi_j p(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j)]\right] && (1.26)\\
&= \sum_{i=1}^{N} \sum_{j=1}^{k} Pr(Z_i = j \mid \boldsymbol{x}_i, \boldsymbol{\alpha}^{t-1}) \log[\pi_j p(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j)] && (1.27)\\
&= \sum_{i=1}^{N} \sum_{j=1}^{k} \gamma_{ij} \log \pi_j + \sum_{i=1}^{N} \sum_{j=1}^{k} \gamma_{ij} \log f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j) && (1.28)
\end{aligned}$$

The two parts of equation 1.28 can be optimized separately at the M step to estimate the parameters of the model. We denote

$$Q(\boldsymbol{\pi}) = \sum_{i=1}^{N} \sum_{j=1}^{k} \gamma_{ij} \log \pi_j \text{ and } Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \sum_{j=1}^{k} \gamma_{ij} \log f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j) \ . \tag{1.29}$$

Thus, the updates of the parameters are obtained as,

$$\hat{\pi_j^t} = \frac{N_j}{N} \tag{1.30}$$

$$\hat{\boldsymbol{\alpha}_j^t} = \underset{\boldsymbol{\alpha}_j}{\operatorname{argmax}}\, Q(\boldsymbol{\alpha}_j) \ , \tag{1.31}$$

where, $N_j = \sum_{i=1}^{N} \gamma_{ij}$.

1.5.1 Classification EM Algorithm

In numerous applications, calculating the Q function during the E-step can be challenging, particularly when the missing data has a high dimensionality or involves incomplete observations like censored data. In such cases, the conditional expectation often involves a high-dimensional integral or an integral over a complex region. To address these difficulties, a variant of the EM algorithm, known as the Classification EM (CEM) or Hard EM algorithm, can be employed to simplify the computational process.

In the case of a Hard EM, the following objective function is optimized.

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \max_{z_1, \ldots, z_N} p_{\boldsymbol{\theta}}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N, z_1, z_2, \ldots, z_N) \ , \tag{1.32}$$

where $\boldsymbol{\theta}$ denotes all parameters. Hard EM maximizes the classification likelihood. It applies a delta function approximation to the posterior probabilities $Pr(Z_i = j \mid \boldsymbol{X} = \boldsymbol{x}, \boldsymbol{\alpha})$, where $Z_i, i = 1, \ldots, N$ are the latent variables representing class labels. For iteration t the approximation changes the E step as follows,

$$Pr(Z_i^t = j \mid \boldsymbol{X} = \boldsymbol{x}, \boldsymbol{\alpha}^t) \approx \boldsymbol{I}(j = z_i^{t*}), \tag{1.33}$$

where, $z_i^{t*} = \underset{j}{\operatorname{argmax}}\, \gamma_{ij}^t = \underset{j}{\operatorname{argmax}} \log Pr(\boldsymbol{x_i} \mid Z_i^t = j, \boldsymbol{\alpha^t}) + \log Pr(Z_i^t = j \mid \boldsymbol{\alpha}^t)$ (Murphy, 2012) .

In other words, Hard EM includes a classification step between E and M step, where data points are assigned to their respective clusters based on the maximum posterior probability z_i^{t*}. At the M step, parameters are updated by obtaining the MLE of the parameters considering only the data points asigned to that respective cluster. Although hard EM uses an approximation to estimate the MLE, it maximizes the classification likelihood by obtaining the MAP estimate, i.e. the mode of the distribution of $Pr(Z_i = j \mid \boldsymbol{X}, \boldsymbol{\alpha})$. Celeux and Govaert (1992) have shown that for a mixture of identical distributions at each iteration the classification likelihood increases and if ML estimates of the mixture densities are well defined, it converges to a stationary point. Below we discuss the proof of convergence of of the classification EM algorithm in case of mixtures of identical distributions as shown by Celeux and Govaert (1992). The proof of convergence in case of mixtures of non-identical distributions is given in chapter 6.

Let, $\boldsymbol{P} = (P_1, P_2, \ldots, P_k)$ be the k partitions or clusters. Then a classification maximum likelihood (CML) criterion can be defined as,

$$C(\boldsymbol{P}, \boldsymbol{\pi}, \boldsymbol{\alpha}) = \sum_{j=1}^{k} \sum_{\boldsymbol{x_i} \in P_j} \log \left[\pi_j f(\boldsymbol{x_i}, \boldsymbol{\alpha_j})\right] \ . \tag{1.34}$$

Theorem 1.5.1 *Any sequence $(\boldsymbol{P}^t, \boldsymbol{\pi}^t, \boldsymbol{\alpha}^t)$ for iteration t increases the CML criterion C and the sequence $C(\boldsymbol{P}^t, \boldsymbol{\pi}^t, \boldsymbol{\alpha}^t)$ converges to a stationary point. Furthermore, if the ML estimates of the parameters are well-defined, the sequence $(\boldsymbol{P}^t, \boldsymbol{\pi}^t, \boldsymbol{\alpha}^t)$ converges to a stationary position.*

Proof At first we show that CML criterion increases at each iteration step. Since, $(\boldsymbol{\pi}^t, \boldsymbol{\alpha_j}^t)$ maximizes $\sum_{\boldsymbol{x_i} \in P_j^t} \log \left[\pi_j f(\boldsymbol{x_i}, \boldsymbol{\alpha_j})\right]$, we can directly write from eq. (1.34),

$$C(\boldsymbol{P}^t, \boldsymbol{\pi}^{t+1}, \boldsymbol{\alpha}^{t+1}) \geq C(\boldsymbol{P}^t, \boldsymbol{\pi}^t, \boldsymbol{\alpha}^t)$$

Now, $\boldsymbol{x_i} \in P_j^{t+1}$ is equivalent to $\gamma_{ij}^{t+1} \geq \gamma_{ij'}^{t+1}\ \forall j' \neq j$, which implies,

$$\pi_j^{t+1} f(\boldsymbol{x_i}, \boldsymbol{\alpha}_j^{t+1}) \geq \pi_{j'}^{t+1} f(\boldsymbol{x_i}, \boldsymbol{\alpha}_{j'}^{t+1})$$

Thus, we can write,

$$C(\boldsymbol{P}^{t+1}, \boldsymbol{\pi}^{t+1}, \boldsymbol{\alpha}^{t+1}) \geq C(\boldsymbol{P}^t, \boldsymbol{\pi}^{t+1}, \boldsymbol{\alpha}^{t+1})$$

Since there is a finite number of partitions of the sample into k clusters, the increasing sequence $C(\boldsymbol{P}^t, \boldsymbol{\pi}^t, \boldsymbol{\alpha}^t)$ takes a finite number of values and as a result it converges to a stationary value. If the ML estimates of $\boldsymbol{\pi}^t$ and $\boldsymbol{\alpha}^t$ are well defined, for a t large enough we can deduce, $\boldsymbol{\pi}^t = \boldsymbol{\pi}^{t+1}$ and $\boldsymbol{\alpha}^t = \boldsymbol{\alpha}^{t+1}$. That directly leads to $\boldsymbol{P}^t = \boldsymbol{P}^{t+1}$. Thus, for a t large enough, we can write,

$$C(\boldsymbol{P}^t, \boldsymbol{\pi}^t, \boldsymbol{\alpha}^t) = C(\boldsymbol{P}^t, \boldsymbol{\pi}^{t+1}, \boldsymbol{\alpha}^{t+1}) = C(\boldsymbol{P}^{t+1}, \boldsymbol{\pi}^{t+1}, \boldsymbol{\alpha}^{t+1})$$

∎

1.6 Identifiability of Finite Mixture Model

The identifiability of a statistical model is crucial because it ensures that the model parameters can be uniquely determined from the observed data, preventing ambiguities in parameter estimation. A well-identified model is essential for reliable and interpretable statistical inferences, providing a solid foundation for drawing meaningful conclusions from empirical observations.
Teicher (1963) has laid down the foundations of identifiability of finite mixture models.

Definition Let, $\mathcal{F} = \{F(\boldsymbol{x}, \boldsymbol{\alpha}); \boldsymbol{\alpha} \in \mathbb{R}_1^m, \boldsymbol{x} \in \mathbb{R}^p\}$ be a family of p dimensional cdf's indexed by a point $\boldsymbol{\alpha}$ in a Borel subset $\mathbb{R}_1^m$ of the Euclidean m space $\mathbb{R}^m$ such that $F(\boldsymbol{x}, \boldsymbol{\alpha})$ is measurable in $\mathbb{R}^p \times \mathbb{R}_1^m$. Then $\mathcal{H}$, the set of all finite mixtures of a class of distributions $\mathcal{F}$ is defined as the convex hull of $\mathcal{F}$:

$$\mathcal{H} = \{H(\boldsymbol{x}) : H(\boldsymbol{x}) = \sum_{j=1}^{k} \pi_j F(\boldsymbol{x}, \boldsymbol{\alpha}_j), \pi_j > 0, \sum_{j=1}^{k} \pi_j = 1, F(\boldsymbol{x}, \boldsymbol{\alpha}_j) \in \mathcal{F}, k = 1, 2, \ldots\} \ . \tag{1.35}$$

$\mathcal{F}$ generates indentifiable finite mixtures if and only if $\mathcal{H}$ has the uniqueness of representation property.

$$\sum_{j=1}^{k} \pi_j F(\boldsymbol{x}, \boldsymbol{\alpha}_j) = \sum_{j=1}^{q} \pi'_j F'(\boldsymbol{x}, \boldsymbol{\alpha}_j) \tag{1.36}$$

implies, $k = q$ and for each $j, 1 \leq j \leq k$ there is some $l, 1 \leq l \leq k$ such that $\pi_j = \pi'_l$ and $F(\boldsymbol{x}, \boldsymbol{\alpha}_j) = F'(\boldsymbol{x}, \boldsymbol{\alpha}_l)$.

Theorem 1.6.1 *The class $\mathcal{H}$, of all finite mixtures of the family $\mathcal{F}$ is identifiable if and only if $\mathcal{F}$ is a linearly independent set over the field of real numbers.*

Corollary 1.6.2 *A necessary and sufficient condition that the class $\mathcal{H}$ of all finite mixtures of the family $\mathcal{F}$ be identifiable is that the image of $\mathcal{F}$ under any vector isomorphism on $\langle \mathcal{F} \rangle$ be linearly independent in the image space.*

The proofs of the above theorem and corollary can be found in Yakowitz and Spragins (1968). In the context of finite mixture model, the theorem and the corollary can be understood in a simple way as follows. The family $\mathcal{F}$ consists of functions $f(\boldsymbol{x} \mid \boldsymbol{\alpha_j})$. If $\mathcal{F}$ is linearly independent over $\mathbb{R}$, this means that the only way a finite linear combination of functions from $\mathcal{F}$ can be zero for all $\boldsymbol{x}$ is if all coefficients are zero.

Mathematically, if

$$\sum_{j=1}^{k} \pi_j f(\boldsymbol{x} \mid \boldsymbol{\alpha_j}) = 0, \quad \text{for all } \boldsymbol{x},$$

then it must be that $\pi_j = 0$ for all j.

If $\mathcal{F}$ is not linearly independent, then there exist non-trivial coefficients π_j such that the sum is identically zero. This means different sets of mixture weights π_j and component densities $f(\boldsymbol{x} \mid \boldsymbol{\alpha_j})$ could lead to the same mixture distribution, causing non-identifiability.

The corollary states that the class $\mathcal{H}$ of all finite mixtures of the family $\mathcal{F}$ is identifiable if and only if the image of $\mathcal{F}$ under any vector isomorphism on the span $\langle \mathcal{F} \rangle$ is linearly

independent in the image space. The span $\langle \mathcal{F} \rangle$ is the vector space generated by all finite linear combinations of the functions in $\mathcal{F}$. A vector isomorphism is a bijective linear map that preserves the vector space structure. The corollary asserts that for the mixture model to be identifiable, the transformed functions under any isomorphism (mapping from the span $\langle \mathcal{F} \rangle$ to another vector space) must be linearly independent in the image space. Mathematically, this means that if we apply an isomorphism ϕ to each component function in $\mathcal{F}$, the set of transformed functions $\{\phi(f(\boldsymbol{x} \mid \boldsymbol{\alpha}_1)), \phi(f(\boldsymbol{x} \mid \boldsymbol{\alpha}_2)), \ldots, \phi(f(\boldsymbol{x} \mid \boldsymbol{\alpha}_k))\}$ must be linearly independent in the new space. Specifically, for any linear combination,

$$\sum_{j=1}^{k} \pi_j \phi(f(\boldsymbol{x} \mid \boldsymbol{\alpha}_j)) = 0, \quad \text{for all } \boldsymbol{x},$$

the only solution must be $\pi_1 = \pi_2 = \ldots = \pi_k = 0$.

If this condition holds, it implies that no two different finite mixtures of the functions $\mathcal{F}$ can result in the same distribution, and the model is identifiable. If the transformed set of functions is not linearly independent, different mixtures might produce the same distribution, leading to non-identifiability. Thus, the linear independence of the image functions guarantees that each distinct mixture corresponds to a unique set of parameters, ensuring identifiability.

Identifiability of popular mixture models such as the Gaussian Mixture Model has been discussed before by many researchers. In chapter 4 we show that the mixtures of Dirichlet distributions are identifiable as well.

1.7 Mixture of Multivariate Gaussian Distributions

For a $p \times 1$ continuous random vector $\boldsymbol{X}$, the density of p variate multivariate normal distribution is given by,

$$f(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})] \ , \tag{1.37}$$

where $\boldsymbol{\mu}$ is a $p \times 1$ vector, $\boldsymbol{\Sigma}$ is a $p \times p$ symmetric, positive definite matrix and the support of $\boldsymbol{X}$ is $\mathbb{R}^p$.

The density of a gaussian mixture model with k mixture components for one observation $\boldsymbol{x}_i$ is given by the mixture density

$$p(\boldsymbol{x}_i) = \sum_{j=1}^{k} \pi_j f(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \ , \tag{1.38}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$ contains the corresponding mixture proportions with $\sum_{j=1}^{k} \pi_j = 1$, $0 < \pi_j < 1$. The density component of mixture j is given by $f(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ and $\boldsymbol{\mu}_j$,$\boldsymbol{\Sigma}_j$, $j = 1, 2, ..., k$ are the location and shape parameters for each density.

The log-likelihood of the model for a sample of size N is then given by

$$\log p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{N} \log \left[\sum_{j=1}^{k} \pi_j f(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \right] \ . \tag{1.39}$$

The posterior probability that the data point i belongs to cluster j is obtained by,

$$\gamma_{ij}(\boldsymbol{x}_i) = Pr(Z_i = j \mid \boldsymbol{X}_i = \boldsymbol{x}_i, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\pi_j f(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{r=1}^{k} \pi_r f(\boldsymbol{x}_i \mid \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)} \ . \tag{1.40}$$

Then the expected complete data log-likelihood can be obtained in a similar way as of eq. (1.28). The first part of the equation remains the same. The second part is given by,

$$\begin{aligned} Q(\boldsymbol{\alpha}) &= \sum_{i=1}^{N}\sum_{j=1}^{k} \gamma_{ij} \left[-\frac{p}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}_j| - \frac{1}{2}\text{tr}\left[(\boldsymbol{x_i} - \boldsymbol{\mu}_j)(\boldsymbol{x_i} - \boldsymbol{\mu}_j)'\boldsymbol{\Sigma}_j^{-1}\right]\right] \\ &= -\frac{Np}{2}\log 2\pi \sum_{j=1}^{k}\frac{N_j}{2}\log|\boldsymbol{\Sigma}_j| - \sum_{j=1}^{k}\frac{N_j}{2}\text{tr}\{\boldsymbol{S_j}\boldsymbol{\Sigma}_j^{-1}\} \ , \end{aligned} \tag{1.41}$$

where, $N_j = \sum_{i=1}^{N}\gamma_{ij}$ and $\boldsymbol{S_j} = \frac{1}{N_j}\sum_{i=1}^{N}\gamma_{ij}(\boldsymbol{x_i} - \boldsymbol{\mu}_j)(\boldsymbol{x_i} - \boldsymbol{\mu}_j)'$.

Finally, at the M step the updates of the parameters are obtained as,

$$\hat{\pi_j} = \frac{N_j}{N} \tag{1.42}$$

$$\hat{\boldsymbol{\mu}_j} = \frac{1}{N_j}\sum_{i=1}^{N}\gamma_{ij}\boldsymbol{x}_i \tag{1.43}$$

$$\hat{\boldsymbol{\Sigma}_j} = \frac{1}{N_j}\sum_{i=1}^{N}\gamma_{ij}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}_j})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}_j})' \tag{1.44}$$

The Classification EM (CEM) algorithm can be utilized to estimate the parameters of a Gaussian Mixture Model, particularly in situations where the standard EM algorithm becomes computationally demanding and slow. As described in the preceding section, CEM approximates the E-step in such a way that, in the M-step, the Gaussian parameters can be updated directly using their maximum likelihood estimates (MLEs).

In the CEM algorithm, the MLEs for Gaussian parameters are calculated individually for each cluster by considering only the data points assigned to that specific cluster. The formulas for these MLEs are provided below and can be directly applied during the M-step in the classification EM framework. This method streamlines the estimation process by leveraging the cluster-specific MLEs, making it especially suitable for high-dimensional or complex datasets where standard EM iterations may be impractical.

The updates of $\boldsymbol{\mu_j}$ and $\boldsymbol{\Sigma_j}$ are given by

$$\hat{\boldsymbol{\mu}_j} = \frac{1}{\#\boldsymbol{x}_i \in P_j}\sum_{\boldsymbol{x}_i \in P_j}\boldsymbol{x}_i \tag{1.45}$$

$$\hat{\boldsymbol{\Sigma}_j} = \frac{1}{\#\boldsymbol{x}_i \in P_j}\sum_{\boldsymbol{x}_i \in P_j}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T \tag{1.46}$$

The updates of π_j is generally obtained by $\frac{\#\boldsymbol{x}_i \in P_j}{N}$. We can also use a modified hard EM algorithm, where the updates of π_j is obtained as of standard EM. The Hard EM

algorithm, while computationally efficient, does have certain theoretical drawbacks that often make it less accurate than the standard EM algorithm (Koloydenko et al., 2007). Unlike standard EM, which iteratively increases the likelihood of the parameters given the observed data, Hard EM instead maximizes the joint likelihood of the latent variables and parameters. This characteristic makes Hard EM inconsistent (Leroux, 1992) and potentially prone to producing biased estimates (Ephraim and Merhav, 2002). Despite these limitations, Hard EM is widely applied in practice due to its simplicity and speed.

However, the question of when Hard EM should be preferred over standard EM remains unresolved, warranting further research into the conditions that favor its use. In Chapter 5, we demonstrate that, for clustering problems, Gaussian Mixture Models estimated using Hard EM can perform comparably to, or even surpass, those estimated with standard EM in certain scenarios.

Gaussian Mixture Models (GMMs) are effective for clustering data that forms spherical or elliptical shapes. However, when the data exhibits asymmetry or heavy-tailed distributions, GMMs are often inadequate. In such cases, mixtures of multivariate t-distributions, skew-normal distributions, or generalized hyperbolic distributions can provide a better fit by accommodating the irregularities in the data.

For more specialized data types, such as compositional data, these common alternatives may still fall short. In these instances, a mixture of Dirichlet distributions is a natural choice, as it inherently respects the constraints and structure of compositional data. This approach ensures a more appropriate modeling framework that aligns with the unique characteristics of such data.

1.8 Mixture of Dirichlet Distributions

The Dirichlet distribution is a widely used probabilistic model for compositional data, where observations lie within the unit simplex, ensuring that values are constrained between 0 and 1 while summing to unity. Over the years, several modifications and generalizations of the Dirichlet distribution have been proposed to address specific limitations. Ongaro and Migliorati (2013) introduced the Flexible Dirichlet distribution, which can be interpreted as a mixture of standard Dirichlet densities with identical precision (Migliorati et al., 2017), allowing for greater modeling flexibility. Tang et al. (2022) developed a variant of the Dirichlet distribution capable of accommodating zero values, overcoming a common limitation in traditional Dirichlet modeling. Makgai et al. (2021) proposed the Kummer–Dirichlet gamma distribution, which enhances robustness against outliers. In the context of mixture models, Di Brisco et al. (2017) employed Flexible Dirichlet distributions; however, this approach significantly increases the number of parameters, making it less practical in high-dimensional settings. Wang et al. (2008) introduced a method for dimensionality reduction of compositional data, yet their approach comes at the cost of higher computational complexity.

Beyond compositional data, the Dirichlet distribution is extensively utilized as a prior for the Multinomial distribution in categorical data analysis, where Holmes et al. (2012) leveraged the Dirichlet-multinomial model within a mixture framework. Dirichlet distribution is also used in Dirichlet Process Mixture Models (DPMM) (Antoniak, 1974), where it serves as a nonparametric Bayesian prior to allow for an infinite mixture of

components, enabling the model to adaptively determine the number of clusters based on the data rather than requiring a predefined number of components. However, their computational complexity often necessitates approximation techniques such as Markov Chain Monte Carlo (MCMC) or variational inference for practical implementation. In this study, we focus on the Dirichlet Mixture Model (DMM) due to its simpler structure and reduced number of parameters, thereby improving computational efficiency in clustering applications. For a detailed discussion on estimation techniques and different variants of Dirichlet mixture models, refer to Chapter 4.

Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N$ denote a random sample of size N, where $\boldsymbol{X}_i$ is a p dimensional random vector with probability density function $f(\boldsymbol{x}_i)$ on $\mathbb{R}^p$.

The Dirichlet density is given by

$$f(\boldsymbol{x}_i) = \frac{\Gamma(\sum_{m=1}^{p} \alpha_m)}{\prod_{m=1}^{p} \Gamma(\alpha_m)} \prod_{m=1}^{p} x_{im}^{\alpha_m - 1} \ , \tag{1.47}$$

where $\sum_{m=1}^{p} x_{im} = 1, x_{im}$'s > 0 , α_m's > 0 and $\Gamma(\cdot)$ denotes a gamma function.

The density of a mixture model with k mixture components for one observation $\boldsymbol{x}_i$ is given by the mixture density

$$p(\boldsymbol{x}_i) = \sum_{j=1}^{k} \pi_j f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j) \ , \tag{1.48}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$ contains the corresponding mixture proportions with $\sum_{j=1}^{k} \pi_j = 1$, $0 < \pi_j < 1$.

The log likelihood of the model for a sample of size N is then given by

$$\log p(\boldsymbol{x_1}, \ldots, \boldsymbol{x_N} | \boldsymbol{\alpha}, \pi) = \sum_{i=1}^{N} \log \left[\sum_{j=1}^{k} \pi_j f(\boldsymbol{x_i} | \boldsymbol{\alpha_j}) \right] \ . \tag{1.49}$$

The parameters of Dirichlet Mixture Model (DMM) can be estimated using an Expectation-Maximization (EM) algorithm (Dempster et al., 1977). We can employ both soft and hard version of EM for estimation purpose. Similar to a GMM, latent variables Z_i's are introduced, which are categorical variables taking on values $1, \ldots, k$ with probabilities $\pi_1, \ldots, \pi_k$ such that $Pr(\boldsymbol{X_i} | Z_i = j) = f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j)$, $j = 1, \ldots, k$.

For the soft EM version the E and M step can be obtained as follows.

E-Step: The cluster membership probabilities of data point i for cluster j can be obtained by γ_{ij} like before, where,

$$\gamma_{ij}(x_i) = Pr(Z_i = j | \boldsymbol{x_i}, \boldsymbol{\alpha_j}) = \frac{\pi_j f(\boldsymbol{x_i} | \boldsymbol{\alpha_j})}{\sum_{j=1}^{k} \pi_j f(\boldsymbol{x_i} | \boldsymbol{\alpha_j})} \ . \tag{1.50}$$

Subsequently the expected complete data log-likelihood is given by,

$$Q(\alpha, \alpha^{t-1}) = \mathbb{E}\left[\sum_{i=1}^{N} \log(p(x_i, z_i|\alpha))|x, \alpha^{t-1}\right] \tag{1.51}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{k} \gamma_{ij} \log \pi_j + \sum_{i=1}^{N}\sum_{j=1}^{k} \gamma_{ij} \log f(x_i|\alpha_j) \ . \tag{1.52}$$

M-Step: Here we maximize the expected complete data log-likelihood obtained from the E step. The update of $\boldsymbol{\pi}$ is obtained as $\hat{\pi}_j = \frac{N_j}{N}$, where $N_j = \sum_{i=1}^{N} \gamma_{ij}$. The update of α_{jm} is the solution of the following equation (Pal and Heumann, 2025).

$$\Psi(\alpha_{jm}) = \Psi\left(\sum_{m=1}^{p} \alpha_{jm}\right) + \frac{1}{N_j}\sum_{i=1}^{N} \gamma_{ij} \log(x_{im}) \ , \tag{1.53}$$

where, $\Psi(\cdot)$ is the di-gamma function. To obtain the updates of all the parameters we need to solve the equation for all m and all j. As there is no analytically tractable solution available, it is generally solved using Fixed-Point iteration or Newton-Rahpson Method.

For the hard version of EM the E-step remains unchanged. However, we need to include an additional classification step.

Classification Step: It applies a delta function approximation that changes the E step as follows,

$$Pr(Z_i = j|\boldsymbol{x_i}, \boldsymbol{\alpha_j}) \approx \mathbb{I}(j = z_i^*) \ , \tag{1.54}$$

where $z_i^* = \underset{j}{\text{argmax}}\, \gamma_{ij}$.

M-Step: Here we maximize the approximated expected complete data log-likelihood. The update of $\boldsymbol{\pi}$ is obtained as $\hat{\pi}_j = \frac{\#\boldsymbol{x_i} \text{in cluster } j}{N}$. The update of α_{jm} is found by solving the following equation.

$$\Psi(\alpha_{jm}) = \Psi\left(\sum_{m=1}^{p} \alpha_{jm}\right) + \frac{1}{\#\boldsymbol{x}_i \in P_j}\sum_{\boldsymbol{x}_i \in P_j} \log(x_{im}) \tag{1.55}$$

Subsequently, the updates of all the parameters are obtained by solving the equation for all m and all j. Similar to soft EM there is no analytically tractable solution available. And that is why it is solved using Fixed-Point iteration or Newton-Rahpson Method.

1.8.1 Mean-Precision Parametrization

The standard parametrization of the Dirichlet distribution is often challenging to interpret, so we propose an alternative parametrization using mean and precision parameters. The mean parameter dictates the distribution's location, while the precision parameter controls its concentration. When precision is high, the Dirichlet random variable clusters around the mean values, but with low precision, the distribution spreads out more. This parametrization offers clearer insights into the model's parameters, enhancing interpretability and providing a better understanding of how they influence the distribution.

By interpreting these parameters, we can also identify scenarios where it may be beneficial to fix one and optimize the other. Notably, the mean and precision parameters show partial decoupling in the maximum-likelihood framework, allowing for simplifications and potential speed improvements through alternative optimization strategies. This alternative parametrization not only improves the interpretability of the Dirichlet Mixture Model (DMM), but also increases its adaptability to the data, thus expanding the range of fitting and optimization strategies available. This approach is versatile, accommodating a wide variety of scenarios with different locations and concentration levels. The parameter estimates vary depending on whether one or both parameters are unknown and can further differ if the precision is uniform across mixture components.

For component j, we denote the mean parameter as $\boldsymbol{M}_j$ and the precision parameter as S_j. Here, $\boldsymbol{M}_J = (M_{j1}, \ldots, M_{jp})$ is a p dimensional vector.

Let us consider the following reparameterization of Dirichlet parameters.

$$S_j = \sum_{m=1}^{p} \alpha_{jm} \text{ and } M_{jm} = \mathbb{E}[X_{jm}] = \frac{\alpha_{jm}}{S_j} \ .$$

Hence, we denote $\alpha_{jm} = S_j M_{jm}$. In chapter 4, we provide an in-depth discussion of four distinct scenarios, along with their respective estimation procedures. The scenarios are namely,

- M_{jm} known, S_j unknown
- M_{jm} unknown, S_j known
- M_{jm}, S_j both unknown
- S_j's are identical

1.8.2 Estimates For High Dimensional Data

Estimating Dirichlet parameters becomes challenging with high-dimensional data, as computation time grows with increasing p. Additionally, since there are no closed-form solutions for parameter updates at the M-step, this can lead to computational errors. In our work, we introduce two approximations that yield closed-form solutions at the M-step, independent of p. In high-dimensional settings, methods like Newton-Raphson or non-quadratic approximations at the M-step can fail, with issues such as non-invertible Hessian matrices or unmet conditions for convergence. Our approximations bypass iterative algorithms at each M-step, thus reducing execution time and avoiding computational errors. While our parametrization offers clear benefits, optimizing mean and precision separately increases computation time. This is particularly important in high-dimensional cases where efficient computation is crucial. Therefore, in chapter 4, we employ the standard parametrization, using Stirling's and moment approximations to provide robust estimates for high-dimensional applications.

Let $\sum_{r=1}^{p}(1 - \delta_{r,m})\alpha_{jr} = \beta_{jm}$, where $\delta_{r,m}$ is the Kronecker delta, defined as:

$$\delta_{r,m} = \begin{cases} 1 & \text{if } r = m \\ 0 & \text{if } r \neq m \end{cases}$$

Then we can show that using stirling's approximation (Artin, 2015), eq. (1.53) becomes,

$$\Psi(\alpha_{jm}) = \log \sum_{r=1}^{p} (1 - \delta_{r,m})\alpha_{jr} + \frac{1}{N_j} \sum_{i=1}^{N} \gamma_{ij} \log(x_{im}) \tag{1.56}$$

$$\alpha_{jm} = \Psi^{-1} \left(\log \sum_{r=1}^{p} (1 - \delta_{r,m})\alpha_{jr} + \frac{1}{N_j} \sum_{i=1}^{N} \gamma_{ij} \log(x_{im}) \right) \tag{1.57}$$

Furthermore, using moment approximation for iteration t it can be deduced to,

$$\alpha_{jm}^{t} = \Psi^{-1} \left(\log(\hat{S}_j - \alpha_{jm}^{t-1}) + \frac{1}{N_j} \sum_{i=1}^{N} \gamma_{ij} \log(x_{im}) \right) \tag{1.58}$$

1.8.3 Kullback-Leibler (KL) Divergence

The Kullback-Leibler (KL) Divergence (Csiszár, 1975) stands as a fundamental measure in statistics, quantifying the statistical distance between probability distributions. The Kullback-Leibler (KL) divergence (also recognized as relative entropy) between two probability density functions $f(x)$ and $g(x)$ is defined by the integral expression:

$$D(f\|g) \stackrel{\text{def}}{=} \int f(x) \log \left(\frac{f(x)}{g(x)} \right) \mathrm{dx} \ . \tag{1.59}$$

It operates as a measure of the dissimilarity between the probability distributions encoded by $f(x)$ and $g(x)$. In statistical inference, generally $f(x)$ is the distribution with true parameter values and $g(x)$ is the distribution with estimated parameter values.

The KL divergence adheres to fundamental properties, denoted as the divergence properties:

- **Self Similarity:** $D(f\|f) = 0$
- **Self Identification:** $D(f\|g) = 0$ if and only if $f = g$
- **Positivity:** $D(f\|g) \geq 0$ for all f, g.

These properties underscore the significance of KL divergence in capturing the nuances of distributional disparities, making it a cornerstone in statistical analyses.

Particularly in the realm of Dirichlet Mixture Models (DMM), the utility of KL Divergence becomes pronounced. While closed-form solutions for KL divergence exist for the Dirichlet distribution, extending this analytical tractability to Dirichlet Mixture Models has posed a significant challenge. Past research (Ma et al., 2014) predominantly turned to Monte Carlo methods to approximate the KL divergence in DMMs. However, these methods are computationally intensive and time consuming, which presents a substantial hurdle.

In our study, we address these challenges by proposing a variational approach to approximate KL divergence in Dirichlet Mixture Models. Unlike previous methods, our approach provides a closed-form solution, substantially enhancing computational efficiency. This advancement not only facilitates rapid model comparisons but also ensures a robust evaluation of estimation quality.

Theorem 1.8.1 *Let $\boldsymbol{X}$ be an $p \times 1$ random vector. Assume two Dirichlet distributions u and v specifying the probability distribution of $\boldsymbol{X}$ as,*

$$u: \boldsymbol{X} \sim \text{Dir}(\alpha_{11}, \ldots, \alpha_{1p})$$
$$v: \boldsymbol{X} \sim \text{Dir}(\alpha_{21}, \ldots, \alpha_{2p}) .$$

Then, the Kullback-Leibler divergence of u from v is given by,

$$\begin{aligned} \mathrm{D}(u \,||\, v) = & \log \frac{\Gamma\left(\sum_{i=1}^{p} \alpha_{1i}\right)}{\Gamma\left(\sum_{i=1}^{p} \alpha_{2i}\right)} + \sum_{i=1}^{p} \log \frac{\Gamma(\alpha_{2i})}{\Gamma(\alpha_{1i})} \\ & + \sum_{i=1}^{p} (\alpha_{1i} - \alpha_{2i}) \left[\psi(\alpha_{1i}) - \psi\left(\sum_{i=1}^{p} \alpha_{1i} \right) \right] . \end{aligned} \tag{1.60}$$

Proposition 1.8.2 *Let f_a and g_b be two DMMs such that,*

$$f_a = f(x) = \sum_a \pi_a Dir(\boldsymbol{x}; \boldsymbol{\alpha}_a)$$
$$g_b = g(x) = \sum\nolimits_b \omega_b Dir(\boldsymbol{x}; \boldsymbol{\alpha}_b)$$

Then using a variational approach, an approximated KL divergence can be expressed as,

$$D_{\text{variational}}(f \| g) = \sum_a \pi_a \log \frac{\sum_{a'} \pi_{a'} e^{-D(f_a \| f_{a'})}}{\sum_b \omega_b e^{-D(f_a \| g_b)}}. \tag{1.61}$$

The proofs of the above theorem and proposition are explained in detail in chapter 3.

1.9 Mixtures of Identical and Non-Identical Distributions

Traditionally, mixture models have adhered to the principle of combining components from the same distribution family. While the range of distributions has expanded over time, this core concept remains central to the theory of multivariate finite mixture models. In their work, Doğru and Arslan (2016) utilized univariate two-component mixtures within mixture regression models, incorporating combinations of normal and t distributions, as well as skew-t and skew-normal distributions. However, the use of multivariate mixtures with both identical and non-identical distributions in unsupervised learning remains unexplored. To address this gap, we propose a novel and flexible framework that allows for the mixing of diverse distributions in any permutation. This comprehensive framework encompasses traditional mixture models as special cases, offering a new perspective on the flexibility and utility of mixture modeling.

Our framework also tackles the complexities of parameter estimation in setups involving intricate multivariate distributions. By employing Classification EM or Hard EM (Celeux and Govaert, 1992), we leverage known Maximum Likelihood Estimates (MLEs) of the component densities to effectively model a wide range of distribution mixtures, circumventing the challenges of parametric inference. This proposed framework is particularly useful for addressing practical challenges related to parameter estimation and pattern recognition, which are common across various real-world applications.

Since we will be using the Classification EM algorithm, we begin by outlining the process for obtaining the Maximum Likelihood Estimates (MLEs) of the parameters for the multivariate skew normal and multivariate generalized hyperbolic distributions.

1.9.1 Multivariate Skew Normal Distribution

The multivariate skew normal distribution (MSN) was first formulated by Azzalini and Capitanio (1999). Let us first consider the following stochastic representation.

Suppose, $\begin{pmatrix} \boldsymbol{Y} \\ Y_0 \end{pmatrix} \sim N_{p+1}(\mathbf{0}, \boldsymbol{\Omega}^*)$, $\boldsymbol{\Omega}^* = \begin{pmatrix} \boldsymbol{\Omega} & \delta_0 \\ {\delta_0}^T & 1 \end{pmatrix}$, where $\delta_0 \in \mathbb{R}^p$ and $\boldsymbol{\Omega}$ is a $p \times p$ symmetric positive definite matrix. Then, $\boldsymbol{U} = sgn(Y_0)\boldsymbol{Y}$ has a density,

$$f(\boldsymbol{u}) = 2\Phi(\boldsymbol{\alpha}^{\boldsymbol{T}}\boldsymbol{u})\phi_p(\boldsymbol{u}; \mathbf{0}, \boldsymbol{\Omega}), \boldsymbol{u} \in \mathbb{R}^p, \tag{1.62}$$

where $\boldsymbol{\alpha} = \boldsymbol{\Omega}^{-1}\frac{\delta_0}{(1-{\delta_0}^T\boldsymbol{\Omega}^{-1}\delta_0)^{1/2}}$,$\Phi(\cdot)$ is the cumulative distribution function of the univariate standard normal distribution and $\phi_p(\boldsymbol{u}; \mathbf{0}, \boldsymbol{\Omega})$ is the p-variate normal density function with mean vector $\mathbf{0}$ and covariance matrix $\boldsymbol{\Omega}$. Now, $\boldsymbol{X} = \boldsymbol{U} + \boldsymbol{\mu}$, is said to have a p dimensional multivariate skew normal distribution with location $\boldsymbol{\mu}$, which is expressed as $SN_p(\boldsymbol{\mu}, \boldsymbol{\Omega}, \boldsymbol{\alpha})$.

Before we formulate the mixture of multivariate skew-normal distribution, let us first look at the parameter estimation procedure of multivariate skew-normal distribution as it will be required at later stage.

Estimating the parameters of the multivariate skew normal distribution is challenging. We follow a technique used by Abe et al. (2021), which incorporates an overparameter into the conventional stochastic representation and then obtains the EM algorithm in a closed form. The stochastic representation is given below.
$\begin{pmatrix} \boldsymbol{Y} \\ Y_0 \end{pmatrix} \sim N_{p+1}(\mathbf{0}, \boldsymbol{\Sigma})$, $\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Omega} & \tau\boldsymbol{\Omega}^{1/2}\boldsymbol{\delta} \\ \tau\boldsymbol{\delta}^T\boldsymbol{\Omega}^{1/2} & \tau^2 \end{pmatrix}$, where $\tau \in \boldsymbol{R}$ and $\boldsymbol{\delta} \in \mathbb{R}^p$.
Let us now denote, $\boldsymbol{\lambda} = \frac{\boldsymbol{\delta}}{\sqrt{1-\boldsymbol{\delta}^T\boldsymbol{\delta}}}$.
Then, it can be shown that $\boldsymbol{U} = sgn(Y_0)\boldsymbol{Y}$ has a multivariate skew normal density with location $\mathbf{0}$, given by,

$$f(\boldsymbol{u}) = 2\Phi(\boldsymbol{\lambda}^{\boldsymbol{T}}\boldsymbol{\Omega}^{-1/2}\boldsymbol{u})\phi_p(\boldsymbol{u}; \mathbf{0}, \boldsymbol{\Omega}), \boldsymbol{u} \in \mathbb{R}^p \ . \tag{1.63}$$

Then we say that $\boldsymbol{X} = \boldsymbol{U} + \boldsymbol{\mu}$, has a p dimensional multivariate skew normal distribution with location $\boldsymbol{\mu}$, which is expressed as $SN_p(\boldsymbol{\mu}, \boldsymbol{\Omega}, \boldsymbol{\lambda})$.
To estimate the parameters using an EM algorithm, let us introduce a latent variable $\boldsymbol{\xi}$ which consists of parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. For N independent random samples drawn from a multivariate skew normal distribution, the expected complete data log-likelihood function for the E step can be written as,

$$\begin{aligned} Q(\boldsymbol{\xi}; \boldsymbol{\xi'}) &= \sum_{i=1}^{N} E[\log f(\boldsymbol{x}_i, y_{0i}; \boldsymbol{\xi}) \mid \boldsymbol{x}_i, \boldsymbol{\xi'}] \\ &= -N\frac{p+1}{2}\log 2\pi - N/2 \log \mid \boldsymbol{\Sigma} \mid -\frac{1}{2}tr\left(\sum_{i=1}^{N} S(\boldsymbol{x}_i, \boldsymbol{\mu}, \boldsymbol{\xi'})\boldsymbol{\Sigma}^{-1}\right), \end{aligned} \tag{1.64}$$

where,

$$S(\boldsymbol{x}_i, \boldsymbol{\mu}, \xi') = \begin{pmatrix} (\boldsymbol{x_i} - \boldsymbol{\mu})(\boldsymbol{x_i} - \boldsymbol{\mu})^T & (\boldsymbol{x_i} - \boldsymbol{\mu})E\left[|y_{0i}| \mid \boldsymbol{x}_i, \boldsymbol{\xi'}\right] \\ (\boldsymbol{x_i} - \boldsymbol{\mu})^T E\left[|y_{0i}| \mid \boldsymbol{x}_i, \boldsymbol{\xi'}\right] & E\left[y_{0i}^2 \mid \boldsymbol{x_i}; \boldsymbol{\xi}\right]. \end{pmatrix} .$$

Let us denote, $c_\lambda = 1/\sqrt{1 + \boldsymbol{\lambda}^T\boldsymbol{\lambda}}$, $\boldsymbol{\gamma} = \boldsymbol{\Omega}^{-1/2}\boldsymbol{\lambda}$, $v_i = \boldsymbol{\gamma}^T(\boldsymbol{x_i} - \boldsymbol{\mu})$,$\rho_1(v) = \frac{\phi(v)}{\Phi(v)} + v$ and $\rho_2(v) = 1 + v\rho_1(v)$.

It can be shown that $E\left[|Y_0| \mid \boldsymbol{X}\right] = \tau c_\lambda \rho_1(\boldsymbol{\gamma}^T \boldsymbol{x})$ and $E\left[Y_0^2 \mid \boldsymbol{X}\right] = \tau^2 c_\lambda^2 \rho_2(\boldsymbol{\gamma}^T \boldsymbol{x})$
Then for the t-th iteration in the M step, the updates of the parameters are given below.

$$\hat{\boldsymbol{\mu}}^{t+1} = \bar{\boldsymbol{x}} - c_{\lambda^t} \left(\hat{\boldsymbol{\Omega}}^t\right)^{1/2} \hat{\boldsymbol{\delta}}^t \frac{1}{N} \sum_{i=1}^{N} \rho_1(\hat{v_i}^t) \ , \tag{1.65}$$

$$\hat{\boldsymbol{\Omega}}^{t+1} = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{x_i} - \hat{\boldsymbol{\mu}}^{t+1})(\boldsymbol{x_i} - \hat{\boldsymbol{\mu}}^{t+1})^T \ , \tag{1.66}$$

$$\hat{\boldsymbol{\delta}}^{t+1} = \left[\frac{1}{N} \sum_{i=1}^{N} \rho_2(\hat{v_i}^t)\right]^{-1/2} \times \left(\hat{\boldsymbol{\Omega}}^{t+1}\right)^{1/2} \times \left[\frac{1}{N} \sum_{i=1}^{N} \rho_1(\hat{v_i}^t)(\boldsymbol{x_i} - \hat{\boldsymbol{\mu}}^{t+1})\right]. \tag{1.67}$$

1.9.2 Multivariate Generalized Hyperbolic Distribution

Before delving into the Generalized Hyperbolic Distribution, we first discuss the Generalized Inverse Gaussian (GIG) Distribution. It was first introduced by Good (1953). Later many other authors (see Barndorff-Nielsen and Halgreen (1977); Blæsild (1978); Halgreen (1979); Jorgensen (2012)) discussed its statistical properties which laid down the foundation for the application of the GIG distribution. If $W \sim GIG(\psi, \chi, \lambda)$, the probability density function can be written in the form,

$$f(w \mid \psi, \chi, \lambda) = \frac{(\psi/\chi)^{\lambda/2} w^{\lambda-1}}{2K_\lambda(\sqrt{\psi\chi})} \exp\left[-\frac{\psi w + \chi/w}{2}\right], \tag{1.68}$$

for $w > 0$, where $\psi, \chi \in \boldsymbol{R}^+$ and K_λ is the modified Bessel function of the third kind with index λ. Gamma distribution and inverse Gaussian distribution are special forms of the GIG distribution. When $\chi = 0$ and $\lambda > 0$, the GIG density reduces to a gamma density. On the other hand, when $\lambda = -1/2$, the GIG density can be seen as a density of an inverse Gaussian distribution.

The Generalized Hyperbolic Distribution has been discussed vividly by McNeil et al. (2015). If $\boldsymbol{X}$ follows a Generalized Hyperbolic Distribution, then its probability density function is given by

$$\begin{aligned} f(\boldsymbol{x} \mid \boldsymbol{\vartheta}) = & \left[\frac{\chi + \delta(\boldsymbol{x}, \boldsymbol{\mu} \mid \boldsymbol{\Sigma})}{\psi + \boldsymbol{\gamma}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\gamma}}\right]^{\frac{\lambda - p/2}{2}} \\ & \times \frac{[\psi/\chi]^{\lambda/2} K_{\lambda - p/2}\left(\sqrt{[\psi + \boldsymbol{\gamma}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\gamma}][\chi + \delta(\boldsymbol{x}, \boldsymbol{\mu} \mid \boldsymbol{\Sigma})]}\right)}{(2\pi)^{p/2} \mid \boldsymbol{\Sigma} \mid^{1/2} K_\lambda(\sqrt{\chi\psi}) \exp[(\boldsymbol{\mu} - \boldsymbol{x})'\boldsymbol{\Sigma}^{-1}\boldsymbol{\gamma}]}, \end{aligned} \tag{1.69}$$

where $\delta(\boldsymbol{x}, \boldsymbol{\mu} \mid \boldsymbol{\Sigma}) = (\boldsymbol{x} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})$ is the squared Mahalanobis distance between $\boldsymbol{x}$ and $\boldsymbol{\mu}$ and $\boldsymbol{\vartheta} = (\lambda, \chi, \psi, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma})$ denotes the parameter space.

A $p \times 1$ Generalized Hyperbolic random vector $\boldsymbol{X}$ can be represented as a variance-mean mixture, consisting of a Generalized Inverse Gaussian (GIG) random variable W and a multivariate Gaussian random vector $\mathbf{Z}$. A random vector $\boldsymbol{X}$ follows a multivariate Generalized Hyperbolic (MGH) distribution, if

$$\boldsymbol{X} = \boldsymbol{\mu} + W\boldsymbol{\gamma} + \sqrt{W}\boldsymbol{Z}, \tag{1.70}$$

where

1. $\boldsymbol{Z} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_{p\times p})$,
2. $\boldsymbol{\mu}, \boldsymbol{\gamma} \in \mathbb{R}^p$,
3. $W \geq 0$ is a scalar-valued random variable which is independent of $\boldsymbol{Z}$ and follows $GIG(\lambda, \chi, \psi)$.

It is important to note that there are other different definitions available that lead to different parameterizations. We now discuss several limiting cases of the MGH distribution.

- **Multivariate Hyperbolic Distribution:** A random vector $\boldsymbol{X}$ is said to have a Multivariate Hyperbolic density (MVH) if it follows an MGH distribution with $\lambda = \frac{p+1}{2}$. It is to be noted that if $\lambda = 1$, its univariate margins follow one-dimensional hyperbolic distributions.
- **Normal Inverse Gaussian Distribution:** When a random vector $\boldsymbol{X}$ follows an MGH distribution with $\lambda = -1/2$, it is said to have a Multivariate Normal Inverse Gaussian (MNIG) density .
- **Variance-Gamma Distribution:** The Variance-Gamma distribution (Barndorff-Nielsen, 1978) is also known as generalized Laplace distribution or the Bessel function distribution. A random vector $\boldsymbol{X}$ is said to have a Multivariate Variance-Gamma density if it follows an MGH distribution with $\lambda > 0$ and $\chi \to 0$.
- **Multivariate Student-t Distribution:** The multivariate t-distribution (MVT) is also a special case of a MGH distribution. When $\psi = 0$, $\lambda < 0$ and $\boldsymbol{\gamma} = 0$, by setting the degree of freedom $\nu = -2\lambda^2$, a MGH distribution can be seen as a multivariate student t distribution.

1.9.3 Mixtures of Non-Identical Distributions

Now we introduce the mixture model with mixture densities from different distributions.

Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N$ denote a random sample of size N, where $\boldsymbol{X}_i$ is a p dimensional random vector with probability density function $f(\boldsymbol{x}_i)$ on $\mathbb{R}^p$. We can write $\boldsymbol{X} = (\boldsymbol{X}_1^T, \ldots, \boldsymbol{X}_N^T)^T$, where the superscript T denotes vector transpose and N denotes the total number of observations. An observed random sample is denoted by $\boldsymbol{x} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_N^T)^T$, where $\boldsymbol{x}_i$ is the observed value of the random vector $\boldsymbol{X}_i$.

The density of a mixture model with k components for one observation $\boldsymbol{x}_i$ is given by the mixture density

$$p(\boldsymbol{x}_i) = \sum_{j=1}^{k} \pi_j f_j(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j) \ , \tag{1.71}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$ contains the corresponding mixture proportions with $\sum_{i=1}^{k} \pi_i = 1$ and $0 \leq \pi_i \leq 1$. $f_j(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j)$ is the density component of mixture j and $\boldsymbol{\alpha}_j$, $j = 1, 2, ..., k$, are vectors of component specific parameters for each density. Then $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)$ denotes the vector of all parameters (except $\boldsymbol{\pi}$) of the model. The log-likelihood of the model for a sample of size N is then given by

$$\log p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \mid \boldsymbol{\alpha}, \boldsymbol{\pi}) = \sum_{i=1}^{N} \log \left[\sum_{j=1}^{k} \pi_j f_j(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j) \right] \ . \tag{1.72}$$

The parameters can be estimated using a Hard EM algorithm with some modifications. For that purpose, let us introduce latent variables Z_i, which are categorical variables taking on values $1, \ldots, k$ with probabilities $\pi_1, \ldots, \pi_k$ such that $Pr(\boldsymbol{X}_i \mid Z_i = j) = f_j(\boldsymbol{x}_i)$, $j = 1, \ldots, k$.

Further probabilities γ_{ij} are introduced (conditional on the observed data $\boldsymbol{X} = \boldsymbol{x}$ and the parameters $\boldsymbol{\alpha}$):

$$\gamma_{ij}(\boldsymbol{x}_i) = Pr(Z_i = j \mid \boldsymbol{X} = \boldsymbol{x}, \boldsymbol{\alpha}) = \frac{\pi_j f_j(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j)}{\sum_{j=1}^{k} \pi_j f_j(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j)} \ . \tag{1.73}$$

Equation 1.73 is the probability of cluster membership j for a data point $\boldsymbol{x}_i$.

Given that we are employing Hard EM, the E-step is approximated as previously discussed. At the M step, we obtain the estimates of π and $\boldsymbol{\alpha}$. For a Hard EM π_j is estimated by $\frac{N_j}{N}$, where, N_j = number of data points in cluster j. $\boldsymbol{\alpha}_j$'s are estimated by ML estimation method considering only the assigned observations of cluster j.

Chapter 6 provides an in-depth exploration of various mixtures involving both identical and non-identical distributions along with appropriate model selection criteria and model diagnostics.

1.10 Discussion of Contributions

This thesis has made substantial contributions to the field of statistical modeling, particularly in advancing the theory and application of finite mixture models. Each component of this work has addressed specific challenges in clustering and parameter estimation, resulting in a suite of novel methodologies that enhance computational efficiency, model flexibility, and interpretability in diverse data scenarios. In this section, we reflect on the key contributions and their implications for research and application.

A significant contribution, shown in Pal and Heumann (2022), lies in the development of a Dirichlet Mixture Model (DMM) tailored for compositional data, which are inherently constrained by unit-sum requirements. Unlike traditional clustering methods, which often rely on data transformations that may distort the underlying structure, the DMM leverages the natural properties of the Dirichlet distribution. By combining this model with a modified Hard EM algorithm, we address challenges such as empty clusters and rapid convergence, providing a robust solution for clustering compositional data. Rigorous comparisons with existing clustering methods (e.g., KMeans and DBSCAN) highlight the effectiveness of the DMM in preserving data characteristics and delivering superior clustering performance across real-world and simulated datasets.

The research laid down in Pal and Heumann (2024b) addresses computational challenges associated with estimating the Kullback-Leibler (KL) divergence in DMMs, a critical measure for model comparison and validation. Traditional Monte Carlo-based approaches, though accurate, are computationally expensive and impractical for large-scale applications. To overcome this limitation, we developed a variational approach that provides a closed-form solution for the KL divergence, significantly improving computational efficiency without compromising accuracy. This contribution has practical implications for extending the applicability of DMMs to complex datasets requiring iterative model

comparisons.

In Pal and Heumann (2025) further advancements were made in parameter estimation for Dirichlet Mixture Models through the introduction of a mean-precision parametrization. This innovative approach enhances model interpretability by linking the mean parameter to the distribution's location and the precision parameter to its concentration. By deriving maximum likelihood estimates for a variety of scenarios and proposing novel estimation techniques for high-dimensional data, this work addresses computational bottlenecks often encountered in high-dimensional settings. The identifiability of the DMM, a previously unexplored area, was rigorously established, providing theoretical foundations for its reliable application.

A foundational aspect of the research done in Pal and Heumann (2024c) is the revisit and refinement of the Hard EM algorithm for Gaussian Mixture Models (GMMs). Traditionally overshadowed by the standard EM algorithm, the Hard EM algorithm is often dismissed due to its perceived limitations, including convergence issues and biased estimates. This work challenges these preconceptions by demonstrating that, with carefully designed modifications, Hard EM can achieve competitive performance in clustering accuracy and computational efficiency. The proposed modifications address convergence challenges, yielding a robust algorithm that outperforms standard methods in specific scenarios. This contribution is particularly valuable for large-scale datasets and applications requiring rapid analysis, as demonstrated through simulations and applications to biological data.

Finally, the study done in Pal and Heumann (2024a) broadens the flexibility of finite mixture models by introducing a framework for mixtures of both identical and non-identical multivariate distributions. This extension allows for combinations of distributions, such as Multivariate Skew Normal and Multivariate Generalized Hyperbolic distributions, providing unparalleled flexibility in modeling diverse data structures. By including traditional mixture models as special cases, this framework bridges gaps in the existing literature, enabling more accurate clustering and parameter estimation in complex, real-world datasets.

Collectively, these contributions push the boundaries of finite mixture modeling, offering theoretical insights and practical tools for analyzing complex datasets. By addressing long-standing limitations in clustering algorithms, parameter estimation, and model flexibility, this research provides a robust foundation for future advancements in statistical modeling and data science. The methodologies developed in this work have broad applicability, extending to fields such as biological sciences, finance, marketing, and beyond, where accurate and interpretable clustering remains a fundamental challenge.

1.11 Concluding Remarks and Outlook

This research has advanced the field of finite mixture models by addressing key challenges in unsupervised learning and clustering, particularly for complex and heterogeneous data structures. Through the development of novel algorithms, improved parameter estimation techniques, and extended model flexibility, this work has expanded the applicability of finite mixture models to new domains and data scenarios. By revisiting and refining the Hard EM algorithm for Gaussian Mixture Models, we demonstrated that computation-

ally simple approaches, when carefully modified, can rival more complex methods in terms of clustering accuracy and efficiency. These findings open pathways for broader applications of model-based clustering in real-world problems requiring fast and robust solutions.

Significant progress was also made in clustering compositional data, which often pose difficulties due to their inherent unit-sum constraint. The introduction of a Dirichlet Mixture Model (DMM) combined with a modified Hard EM algorithm provides a natural, interpretable, and efficient solution for clustering compositional data without requiring transformations. Furthermore, the development of a variational approach for estimating the Kullback-Leibler divergence in DMMs enhanced computational efficiency and accuracy, making it feasible to compare and validate models on large and complex datasets. These innovations contribute to the theoretical understanding and practical utility of mixture models in diverse fields such as biology, business, and the physical sciences.

A key theoretical contribution of this research is the mean-precision parametrization of the Dirichlet distribution, which improves model interpretability by linking the mean to the distribution's location and precision to its concentration. Special estimation techniques tailored for high-dimensional data further enhance the computational feasibility of these models, enabling their use in modern data-intensive applications. Additionally, the introduction of a framework for mixtures of non-identical multivariate distributions—such as Multivariate Skew Normal and Multivariate Generalized Hyperbolic distributions—broadens the flexibility of mixture models, allowing for more accurate modeling of complex, real-world datasets with varied structures.

Looking forward, an exciting direction for future research lies in integrating deep learning techniques with finite mixture models. Advanced methods like autoencoders (Rumelhart et al., 1986) and variational autoencoders (VAEs) (Kingma and Welling, 2014) can be employed to extract latent features from high-dimensional or non-linear data, creating a meaningful low-dimensional representation that preserves the underlying structure. By combining these latent features with the developed mixture models, it is possible to achieve more accurate and interpretable clustering results. Additionally, this hybrid framework holds the potential to enhance clustering performance in datasets with intricate dependencies or noise, making it particularly useful in domains such as image analysis, genomics, and social network analysis.

Further extensions could involve using deep generative models (Goodfellow et al., 2014) for data generation and imputation, addressing challenges such as missing data or imbalanced datasets. Variational inference algorithms that integrate deep learning architectures with the proposed mixture models could provide scalable solutions for large-scale datasets while maintaining the interpretability of model-based clustering. These approaches, coupled with advancements in model diagnostics and validation methods, will pave the way for further innovations in unsupervised learning and data analysis, ensuring the robustness and reliability of these techniques in diverse real-world applications.

In conclusion, this thesis lays the groundwork for future exploration of finite mixture models, combining methodological rigor with practical innovations. By bridging traditional statistical techniques with modern computational approaches, the findings presented here have the potential to inspire a wide range of research endeavors across multiple scientific disciplines, driving progress in both theory and application.

Bibliography

T. Abe, H. Fujisawa, T. Kawashima, and C. Ley. EM algorithm using overparameterization for the multivariate skew-normal distribution. *Econometrics and Statistics*, 19: 151–168, 2021.

C. E. Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The annals of statistics*, pages 1152–1174, 1974.

E. Artin. The gamma function, 2015.

A. Azzalini and A. Capitanio. Statistical applications of the multivariate skew normal distribution. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):579–602, 1999.

M. Balaban, N. Moshiri, U. Mai, X. Jia, and S. Mirarab. Treecluster: Clustering biological sequences using phylogenetic trees. *PloS one*, 14(8):e0221068, 2019.

O. Barndorff-Nielsen. Hyperbolic distributions and distributions on hyperbolae. *Scandinavian Journal of statistics*, pages 151–157, 1978.

O. Barndorff-Nielsen and C. Halgreen. Infinite divisibility of the hyperbolic and generalized inverse Gaussian distributions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 38(4):309–311, 1977.

P. Blæsild. *The shape of the generalized inverse Gaussian and hyperbolic distributions.* Department of Theoretical Statistics, Inst., Univ., 1978.

R. P. Browne and P. D. McNicholas. A mixture of generalized hyperbolic distributions. *Canadian Journal of Statistics*, 43(2):176–198, 2015.

C. R. B. Cabral, V. H. Lachos, and M. O. Prates. Multivariate mixture modeling using skew-normal independent distributions. *Computational Statistics & Data Analysis*, 56 (1):126–142, 2012.

G. Casella and R. Berger. *Statistical inference.* CRC Press, 2024.

G. Celeux and G. Govaert. A classification em algorithm for clustering and two stochastic versions. *Computational statistics & Data analysis*, 14(3):315–332, 1992.

I. Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pages 146–158, 1975.

N. E. Day. Estimating the components of a mixture of normal distributions. *Biometrika*, 56(3):463–474, 1969.

A. P. Dempster, N. M. Laird, and D. 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.

A. Di Brisco, S. Migliorati, et al. A special dirichlet mixture model for multivariate bounded responses. In *Cladag 2017 Book of Short Papers*, pages 1–6. Universitas Studiorum srl, 2017.

F. Z. Doğru and O. Arslan. Robust mixture regression using mixture of different distributions. In *Recent advances in robust statistics: theory and applications*, pages 57–79. Springer, 2016.

R. Durrett. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.

Y. Ephraim and N. Merhav. Hidden markov processes. *IEEE Transactions on information theory*, 48(6):1518–1569, 2002.

F. Galton. *Hereditary genius*. D. Appleton, 1891.

S. Ganesalingam and G. McLachlan. The efficiency of a linear discriminant function based on unclassified initial samples. *Biometrika*, 65(3):658–665, 1978.

I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3-4):237–264, 1953.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

J. Grimmer, M. E. Roberts, and B. M. Stewart. Machine learning for social science: An agnostic approach. *Annual Review of Political Science*, 24(1):395–419, 2021.

C. Halgreen. Self-decomposability of the generalized inverse Gaussian and hyperbolic distributions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 47(1): 13–17, 1979.

G. K. Holmes. Measures of distribution. *Publications of the American Statistical Association*, 3(18-19):141–157, 1892.

I. Holmes, K. Harris, and C. Quince. Dirichlet multinomial mixtures: generative models for microbial metagenomics. *PloS one*, 7(2):e30126, 2012.

H. Jeffreys. An alternative to the rejection of observations. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 137(831):78–87, 1932.

B. Jorgensen. *Statistical properties of the generalized inverse Gaussian distribution*, volume 9. Springer Science & Business Media, 2012.

W. Kim, A. Kanezaki, and M. Tanaka. Unsupervised learning of image segmentation based on differentiable feature clustering. *IEEE Transactions on Image Processing*, 29: 8055–8068, 2020.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

A. Koloydenko, M. Käärik, and J. Lember. On adjusted viterbi training. *Acta Applicandae Mathematicae*, 96(1):309–326, 2007.

S. Lee and G. J. McLachlan. Finite mixtures of multivariate skew t-distributions: some recent and new results. *Statistics and Computing*, 24:181–202, 2014.

B. G. Leroux. Maximum-likelihood estimation for hidden markov models. *Stochastic processes and their applications*, 40(1):127–143, 1992.

T. Li, G. Kou, Y. Peng, and S. Y. Philip. An integrated cluster detection, optimization, and interpretation approach for financial data. *IEEE transactions on cybernetics*, 52 (12):13848–13861, 2021.

T. I. Lin. Maximum likelihood estimation for multivariate skew normal mixture models. *Journal of Multivariate Analysis*, 100(2):257–265, 2009.

Z. Ma, P. K. Rana, J. Taghia, M. Flierl, and A. Leijon. Bayesian estimation of dirichlet mixture model with variational inference. *Pattern Recognition*, 47(9):3143–3157, 2014.

J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

S. Makgai, A. Bekker, and M. Arashi. Compositional data modeling through dirichlet innovations. *Mathematics*, 9(19):2477, 2021.

G. J. McLachlan. Finite mixture models. *A wiley-interscience publication*, 2000.

G. J. McLachlan and K. E. Basford. *Mixture models: Inference and applications to clustering*, volume 38. M. Dekker New York, 1988.

G. J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. John Wiley & Sons, 2008.

G. J. McLachlan, S. X. Lee, and S. I. Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6(1):355–378, 2019.

A. J. McNeil, R. Frey, and P. Embrechts. *Quantitative risk management: concepts, techniques and tools-revised edition*. Princeton university press, 2015.

S. M. McNicholas, P. D. McNicholas, and R. P. Browne. Mixtures of variance-gamma distributions. *arXiv preprint arXiv:1309.2695*, 2013.

X.-L. Meng and D. B. Rubin. Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

S. Migliorati, A. Ongaro, and G. S. Monti. A structured dirichlet mixture model for compositional data: inferential and applicative issues. *Statistics and Computing*, 27(4): 963–983, 2017.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

K. P. Murphy. Probabilistic machine learning: an introduction, 2022.

S. Newcomb. A generalized theory of the combination of observations so as to obtain the best result. *American journal of Mathematics*, pages 343–366, 1886.

T. J. O'neill. Normal discrimination with unclassified observations. *Journal of the American Statistical Association*, 73(364):821–826, 1978.

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

A. O'Hagan, T. B. Murphy, I. C. Gormley, P. D. McNicholas, and D. Karlis. Clustering with the multivariate normal inverse Gaussian distribution. *Computational Statistics & Data Analysis*, 93:18–30, 2016.

S. Pal and C. Heumann. Clustering compositional data using dirichlet mixture model. *Plos one*, 17(5):e0268438, 2022.

S. Pal and C. Heumann. Flexible multivariate mixture models: A comprehensive approach for modeling mixtures of non-identical distributions. *International Statistical Review*, 2024a.

S. Pal and C. Heumann. Gene coexpression analysis with dirichlet mixture model: accelerating model evaluation through closed-form kl divergence approximation using variational techniques. In *International Workshop on Statistical Modelling*, pages 134–141. Springer, 2024b.

S. Pal and C. Heumann. *Gaussian mixture model with modified hard EM algorithm in clustering problems*, pages 153–179. 04 2024c. ISBN 9781003356653. doi: 10.1201/9781003356653-7.

S. Pal and C. Heumann. Revisiting dirichlet mixture model: unraveling deeper insights and practical applications. *Statistical Papers*, 66(1):1–38, 2025.

K. Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

R. Petegrosso, Z. Li, and R. Kuang. Machine learning and statistical methods for clustering single-cell rna-sequencing data. *Briefings in bioinformatics*, 21(4):1209–1223, 2020.

A. Quetelet. *Lettres à SAR le duc régnant de Saxe-Coburg et Gotha: sur la théorie des probabilités, appliquée aux sciences morales et politiques*. M. Hayez, 1846.

A. Quetelet. Sur quelques proprietes curieuses qui presentent les resultats d'une serie d'observations. *Bulletin de l'Academie des sciences, des lettres et des beaux arts de Belgique*, 19:303–317, 1852.

C. R. Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203, 1948.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

S. M. Stigler. *The history of statistics: The measurement of uncertainty before 1900*. Harvard University Press, 1990.

M.-L. Tang, Q. Wu, S. Yang, and G.-L. Tian. Dirichlet composition distribution for compositional data with zero components: An application to fluorescence in situ hybridization (fish) detection of chromosome. *Biometrical Journal*, 64(4):714–732, 2022.

H. Teicher. Identifiability of finite mixtures. *The annals of Mathematical statistics*, pages 1265–1269, 1963.

I. Vrbik and P. McNicholas. Analytic calculations for the em algorithm for multivariate skew-t mixture models. *Statistics & Probability Letters*, 82(6):1169–1174, 2012.

H.-Y. Wang, Q. Yang, H. Qin, and H. Zha. Dirichlet component analysis: feature extraction for compositional data. In *Proceedings of the 25th international conference on Machine learning*, pages 1128–1135, 2008.

J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

W. F. R. Weldon. I. certain correlated variations in crangon vulgaris. *Proceedings of the Royal Society of London*, 51(308-314):1–21, 1892.

W. F. R. Weldon. Ii. on certain correlated variations in carcinus mænas. *Proceedings of the Royal Society of London*, 54(326-330):318–329, 1894.

J. H. Wolfe. Pattern clustering by multivariate mixture analysis. *Multivariate behavioral research*, 5(3):329–350, 1970.

S. J. Yakowitz and J. D. Spragins. On the identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 39(1):209–214, 1968.

F. Zehra Doğru, Y. Murat Bulut, and O. Arslan. Finite mixtures of multivariate skew laplace distributions. *REVSTAT-Statistical Journal*, 19(1):35–46, Mar. 2021. doi: 10.57805/revstat.v19i1.330. URL https://revstat.ine.pt/index.php/REVSTAT/article/view/330.

Chapter 2

Clustering compositional data using Dirichlet mixture model

Summary

This article explores a model-based clustering method tailored for compositional data, which typically requires data transformation. The proposed method utilizes a Dirichlet mixture model that naturally adheres to the unit sum constraint of compositional data. To address issues of rapid convergence leading to empty clusters, the model employs a modified hard Expectation-Maximization (EM) algorithm. The effectiveness of this approach is rigorously evaluated through simulations across varying dimensions, cluster numbers, and overlaps. The method's performance is compared against other popular clustering algorithms, such as KMeans, Gaussian Mixture Models (GMM), and DBSCAN, using both simulated data and real-world datasets from business, marketing, and physical sciences. The results indicate that the proposed method effectively captures the unique distributional characteristics of compositional data, showing promise for diverse applications.

Contributing Article

Pal, Samyajoy, and Christian Heumann. "Clustering compositional data using Dirichlet mixture model." *Plos one 17, no. 5 (2022): e0268438.* https://doi.org/10.1371/journal.pone.0268438.

Author Contributions

- Samyajoy Pal: Conceptualization, Methodology, Software, Writing – original draft
- Christian Heumann: Supervision, Writing – review & editing

PLOS ONE

RESEARCH ARTICLE

Clustering compositional data using Dirichlet mixture model

Samyajoy Pal*, Christian Heumann

Department of Statistics, LMU Munich, Munich, Bayern, Germany

* Samyajoy.Pal@stat.uni-muenchen.de

Abstract

A model-based clustering method for compositional data is explored in this article. Most methods for compositional data analysis require some kind of transformation. The proposed method builds a mixture model using Dirichlet distribution which works with the unit sum constraint. The mixture model uses a hard EM algorithm with some modification to overcome the problem of fast convergence with empty clusters. This work includes a rigorous simulation study to evaluate the performance of the proposed method over varied dimensions, number of clusters, and overlap. The performance of the model is also compared with other popular clustering algorithms often used for compositional data analysis (e.g. KMeans, Gaussian mixture model (GMM) Gaussian Mixture Model with Hard EM (Hard GMM), partition around medoids (PAM), Clustering Large Applications based on Randomized Search (CLARANS), Density-Based Spatial Clustering of Applications with Noise (DBSCAN) etc.) for simulated data as well as two real data problems coming from the business and marketing domain and physical science domain, respectively. The study has shown promising results exploiting different distributional patterns of compositional data.

Citation: Pal S, Heumann C (2022) Clustering compositional data using Dirichlet mixture model. PLoS ONE 17(5): e0268438. https://doi.org/10.1371/journal.pone.0268438

Editor: Sheetal Kalyani, IIT Madras, INDIA

Received: August 9, 2021

Accepted: April 30, 2022

Published: May 18, 2022

Data Availability Statement: Data are available from https://archive.ics.uci.edu/ml/datasets/wholesale+customers and https://archive.ics.uci.edu/ml/datasets/wine.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Introduction

In statistics, compositional data are quantitative descriptions of the parts of some whole, which means that it consists of relative information [1]. Mathematically, compositional data follows the Aitchison geometry on the simplex [2]. Measurements including probabilities, proportions, percentages, and ppm can all be thought of as compositional data. In general, compositional data is written as,

$$S^D = \left\{ x = [x_1, x_2, ..., x_D] \in \mathbb{R}^D | x_i > 0, i = 1, 2, ..., D; \sum_{i=1}^{D} x_i = c \right\} \quad (1)$$

In other words, compositional data is a D dimensional real vector, $x = [x_1, x_2, \ldots, x_D]$ of positive components on $\mathbb{R}^D$ such that the sum of all components is c. Often, we observe the sum of all components to be 1; if not, all the components are divided by the sum of all components, such that $\sum_{i=1}^{D} x_i = 1$. Analysis of such data is widely used in the fields of geochemistry [3, 4], biology [5–7], ecology [8, 9], finance and business studies [10–12], etc. But it has

emerged in the literature long before. [13] identified the problem of 'spurious correlation' between ratios of variables and [14] later extended the work and showed that some of the correlations between components of the composition must be negative because of the unit sum constraint. Many transformations have been proposed over the years (e.g. log transformation [15], log ratio transformation [16]) to overcome the unit sum constraint, but still it is argued when it comes to choosing the best transformation [17].

Another issue with compositional data refers to the dealing with zero values as both ratios as well as logarithms are operations that require non-zero elements in the data matrix. Many researchers have tried different approaches to deal with zero values (see [18–21]), but it remains as an open problem even today; mostly because, zero values occur in compositional data for different reasons. Often, the "zero problem" is linked with the missing data problem. Missing data are generally classified into three categories [22], namely: missing completely at random (MCAR), missing at random (MAR) and not missing at random (NMAR). In compositional data analysis, the rounded zeros are considered a NMAR case, where data cannot be observed because their values are below a known value ϵ. Zero values can also occur when the count of an element is zero (known as count zero) and when zero signifies some property or relevant information (known as essential zeros). For our study of compositional data in cluster analysis, we have encountered round zeros and we have used a method proposed by [20], where we replace the zeros with a small quantity and adjust others in a multiplicative way which does not affect the covariance structure of the data. The adjusted values xr_{ij} can be written as

$$xr_{ij} = \begin{cases} \delta_{ij} & \text{if } x_{ij} = 0 \\ x_{ij}\left(1 - \frac{\sum_{k|x_{ik}=0}\delta_{ik}}{c_i}\right) & \text{if } x_{ij} > 0 \end{cases} \tag{2}$$

where c_i is usually the sum constraint. For row i and component j, the above adjustment in Eq 2 replaces the component x_{ij} by a very small quantity δ_{ij} if $x_{ij} = 0$, else it multiplies a term $\left(1 - \frac{\sum_{k|x_{ik}=0}\delta_{ik}}{c_i}\right)$ with x_{ij} to maintain the unit sum constraint. Here, x_{ik}'s are the zero components in row i. The multiplicative term is a fraction by which the non-zero terms to be reduced in order to accommodate the added values of δ_{ik}'s and keep the sum of rows fixed at c_i.

For clustering compositional data there exists many methods in the literature [23, 24]. We generally see two kinds of approaches, namely; model based methods, e.g. mixture models [25] and methods based on dissimilarity distances (e.g. hierarchical clustering [26], KMeans [27]. But most of the time researchers go for Gaussian mixture model or KMeans for clustering purposes [28].

For estimating the parameters of mixture models, the EM algorithm [29–31] is widely used. In many applications of mixture models, e.g. in image matching [32], and audio and video scene analysis [33], the EM algorithm is being used regularly. But the EM algorithm is often not very convenient to apply for other than normal distributions, because it needs to be modified and adapted for each case. Sometimes, updating the parameters in the M step becomes impossible for some distributions [34].

The main objectives of our study are to

- develop a clustering method without the need of transformation of compositional data,
- build a mixture model with distribution other than normal,

- evaluate the performance of the method in different situations (different dimensions, different number of clusters and varied overlap).

We are going to propose a model based clustering method without transformation of compositional data. We have used a Hard EM [35] with some modifications, to build mixture models using Dirichlet distribution. For that purpose we need some point estimates of the parent distributions. It is very convenient as it works with both, likelihood based and Bayesian estimates. But the problem with hard assignment of cluster is that it ignores cluster membership probabilities of less probable clusters. As a result, often the algorithm converges too quickly with one or more clusters being empty. In our study we have also proposed a way to deal with that problem. We have done rigorous simulation study to evaluate the performance of the proposed method over varying dimension, number of clusters and overlap. We have also used two real dataset from business and physical science domain to illustrate the method.

Methodology

Let $X_1, X_2, \ldots, X_N$ denote a random sample of size N, where X_i is a p dimensional random vector with probability density function $f(x_i)$ on $\mathbb{R}^p$. We can write $X = (X_1^T, \ldots, X_N^T)^T$, where the superscript T denotes vector transpose. Note that the entire sample is represented by X, i.e. X is a N—tuple of points in $\mathbb{R}^p$ or an $N \times p$-matrix. $x = (x_1^T, \ldots, x_N^T)^T$ denotes an observed random sample where x_i is the observed value of the random vector X_i.

The density of a mixture model with k components for one observation x_i is given by the mixture density

$$p(x_i) = \sum_{j=1}^{k} \pi_j f_j(x_i|\alpha_j) \ , \tag{3}$$

where $\pi = (\pi_1, \ldots, \pi_k)$ contains the corresponding mixture proportions with $\sum_{i=1}^{k} \pi_i = 1, 0 \leq \pi_i \leq 1$. $f_j(x_i|\alpha_j)$ is the density component of mixture j and $\alpha_j, j = 1, 2, \ldots, k$, are vectors of component specific parameters for each density. Then $\alpha = (\alpha_1, \ldots, \alpha_k)$ denotes the vector of all parameters of the model. The log likelihood of the model for a sample of size N is then given by

$$\log p(x_1, \ldots, x_N|\alpha, \pi) = \sum_{i=1}^{N} \log \left[\sum_{j=1}^{k} \pi_j f_j(x_i|\alpha_j) \right] \ . \tag{4}$$

The parameters can be estimated using the EM algorithm with some modifications. For that purpose, let us introduce latent variables Z_i, which are categorical variables taking on values $1, \ldots, k$ with probabilities $\pi_1, \ldots, \pi_k$ such that $Pr(X_i|Z_i = j) = f_j(x_i), j = 1, \ldots, k$. Further, probabilities γ_{ij} are introduced (conditional on the observed data $X = x$ and the parameters α):

$$\gamma_{ij}(x_i) = Pr(Z_i = j|X = x, \alpha) = \frac{\pi_j f_j(x_i|\alpha_j)}{\sum_{j=1}^{k} \pi_j f_j(x_i|\alpha_j)} \ . \tag{5}$$

Eq 5 can be seen as a cluster membership probability of data point i for cluster j. For an EM algorithm, we try to optimize the function

$$Q(\alpha, \alpha^{t-1}) = E \left[\sum_{i=1}^{N} \log(p(x_i, z_i|\alpha))|x, \alpha^{t-1} \right] \ , \tag{6}$$

where t is the current iteration number. It is nothing but the expected complete data log

likelihood. It can also be shown that (see [36]),

$$Q(\alpha, \alpha^{t-1}) = \sum_{i=1}^{N}\sum_{j=1}^{k}\gamma_{ij}\log\pi_j + \sum_{i=1}^{N}\sum_{j=1}^{k}\gamma_{ij}\log f_j(x_i|\alpha_j)\ , \tag{7}$$

where the expected complete data log likelihood is expressed as sum of two parts. At the M step, we optimize Q with respect to π and α. π_j is estimated in the usual way by $\frac{N_j}{N}$, where, $N_j = \sum_{i=1}^{N}\gamma_{ij}$ and for estimating α, we look at the part in Q (Eq 7) which depends on α, which is given by,

$$l(\alpha) = \sum_{i=1}^{N}\sum_{j=1}^{k}\gamma_{ij}\log f_j(x_i|\alpha_j) \tag{8}$$

Now, we choose α_j such that $\alpha_j^t = \underset{\alpha_j}{\text{argmax}}\, l(\alpha_j)$, which is obtained by the process of assigning data points to respective clusters, given by $\underset{j}{\text{argmax}}\,\gamma_{ij}$, and estimate α_j by some estimation method based on the assigned observations to that cluster. It can be seen as a Bayesian concept (although not strictly Bayesian) for learning where Eq 5 provides the cluster membership probability. The idea of choosing the cluster based on maximum probability is the same as choosing the MAP estimate, the mode of the distribution of $Pr(Z_i = j|X, \alpha)$.

To run the algorithm, at first, some trial values of the distribution parameters α and mixture proportions π are initialized. Then the initial value of the log likelihood is evaluated. For different distributions, different techniques can be used to choose suitable initial values. For example, in the case of a GMM, the centroids of KMeans can be used as initial values of μ and the empirical covariance matrix of each cluster can be taken as an initial value of Σ_j. On the other hand, for a Dirichlet Mixture Model, centroids of KMeans can be multiplied with a scalar c (for our study we have used $c = 60$) to get the initial values of the α parameters. Please recall that the mean vector of a Dirichlet distribution consists of the ratios of α parameters and the sum of all α parameters. Here, the scalar c acts as the sum of α parameter values. The initial values of π can be obtained by generating a random number from a Dirichlet (1,1,1,. . .,1) distribution. The empirical ratios of the number of cluster members in the KMeans algorithm and total observations can also be used as the initial values of π. For our study, we have used the KMeans initialization technique mentioned above for all our experiments.

At the E step, the values of the probabilities γ_{ij} are evaluated using the current parameter values. For an usual EM algorithm (e.g. in a GMM), at the M step, a weighted mean and a weighted covariance matrix are calculated using the γ_{ij} values. But for other distributions, where the model parameters are not mean and (co)variance, this technique can not be used. So, for different distributions, different techniques needs to be used. And also, for such Hard EM, sometimes the algorithm converges with one or more clusters being empty. Hence, one might have to force the algorithm to re-iterate if one or more clusters are found to be empty at each M step. To introduce a flexible, yet convenient solution, we propose a different technique in our algorithm, where at the M step each data point is assigned to a cluster depending on the probability of that data point belonging to each cluster. That cluster is assigned for which the probability is maximum. Now, if one or more clusters are found empty then the initial value of the parameter α_j for empty cluster j is used. And for the non empty clusters, point estimates of the parameters of each parent distributions are obtained using only the data points available in each cluster. For faster convergence and convenience, maximum likelihood estimates can usually be recommended. The mixture component probabilities π_j are estimated as mentioned

above by $\frac{N_j}{N}$. The newly set of estimated values of the parameters is then used as an update over the previous one. After this step, the log likelihood is evaluated again using the updated parameter values. The process is then continued until convergence. The convergence properties of this algorithm follow the properties of the usual EM algorithm, which has been explained in detail by [37, 38].

Algorithm 1: Clustering algorithm for mixture of Dirichlet distributions with provision for empty clusters (Hard DMM 1)

```
Replace zero values in the data, if any, using Eq 2;
Initialize the model parameters, α and π. Evaluate the initial value
of the log likelihood from Eq 4;
while log likelihood difference ≥ ϵ do
  Evaluate γ_ij from Eq 5, using the parameter values and data;
  π_j^new = N_j/N, where, N_j = Σ_{i=1}^{N} γ_ij;
  for i in 1 to N do
    cluster = argmax_j γ_ij;
    Assign data point x_i to cluster z_i;
  end
  for j in 1 to k do
    if cluster j is empty then
      Use initial values of α_j as an update;
    else
      α_j^new = α_j^MLE;
    end
  end
  Re-evaluate log likelihood using the new estimates of the
  parameters.
end
```

For our experiments, we have used 0.0001 as the value of ϵ in Algorithm 1.

For clustering compositional data, a Dirichlet Mixture Model can be used. The Dirichlet density component j is given by

$$\begin{aligned} f_j(x_i) &= \frac{\Gamma(\sum_{m=1}^{p}\alpha_{jm})}{\prod_{m=1}^{p}\Gamma(\alpha_{jm})}\prod_{m=1}^{p}x_{im}^{\alpha_{jm}-1}, \\ & where \sum_{m=1}^{p}x_{im} = 1, x_{im}\text{'s} > 0\ , \alpha_{jm}\text{'s} > 0\ . \end{aligned} \tag{9}$$

If we make a finite mixture with k components, the model is given by Eq 3 and subsequently, the log likelihood is given by Eq 4.

The model parameters, can be easily estimated using our generalized approach. For that we need a good point estimate of the parameters of a Dirichlet distribution to be used in the M step of our algorithm. [39] has discussed a way to find out the maximum likelihood estimates of a Dirichlet distribution, where he proposed to perform a fixed point iteration, given an initial value of the α parameters. The equation is given by

$$\Psi(\alpha_{jm}^{new}) = \Psi\left(\sum_{m=1}^{p}\alpha_{jm}^{old}\right) + \frac{1}{N_j}\sum_{i=1}^{N_j}\log(x_{im}) \tag{10}$$

At each iteration, for an old value of the parameter α_{jm}^{old}, a new value α_{jm}^{new} is obtained. This iteration in the algorithm requires inverting Ψ, which is a digamma function. A suitable initial value and inversion algorithm is also discussed by [39].

A special provision of Bayesian estimates for clusters with fewer data points

It is possible to add a further step in algorithm 1 to consider the case when there are very few data points in a cluster due to hard assignment. In this situation, Bayesian estimates can be very useful as they can use some prior information about the model parameters. Also, for fewer data points, maximum likelihood estimates are known to be less accurate. However, Bayesian estimation of Dirichlet parameters is tricky due to several reasons. Even though the Dirichlet distribution has a conjugate prior for being a member of the exponential family, the posterior distribution is difficult to use in practical problems and not analytically tractable. Few authors have proposed some approximation to the posterior distribution of Dirichlet parameters (e.g. [40] have used multivariate Gaussian distribution), but no method seems to yield satisfactory results. Also, using some Markov Chain Monte Carlo (MCMC) algorithm at each iteration step of the clustering algorithm makes it too time-consuming, which is not practically feasible. Considering all these challenges, we are going to propose a suitable solution that can be adopted in our clustering algorithm.

Let us recall that, if $(X_1, X_2, \ldots, X_p)$ follows a Dirichlet distribution, with parameters $(\alpha_1, \alpha_2, \ldots, \alpha_p)$ then the marginal distribution of X_i follows a Beta distribution with parameters $(\alpha_i, \sum_{j=1}^{p}(\alpha_j - \alpha_i))$. Now, if we choose the prior distribution of α_i as *Gamma* (a, b), then under certain assumptions, the posterior distribution of α_i can be obtained in closed form. It can be shown that (see [41]), posterior distribution of α_i follows a Gamma distribution with parameters $(a + n)$ and $\frac{1}{b-\sum_{i=1}^{n} log\ x_i}$, where n is the sample size.

Thus, for our clustering problem, the Bayesian estimates of α_{jm}, $m = 1, 2, \ldots p$ for cluster j can be obtained by the posterior mean, which is given by,

$$\alpha_{jm}^{Bayes} = E(\alpha_{jm}) = \frac{a + N_j}{b - \sum_{i=1}^{N_j} log\ x_{im}} \tag{11}$$

For our experiment, we have chosen the values of a and b to be 1. The extended algorithm with Bayesian estimates for clusters with fewer data points (Hard DMM 2) is explained below.

Algorithm 2: Clustering algorithm for mixture of Dirichlet distributions with special provision for clusters with fewer data points (Hard DMM 2)

```
Replace zero values in the data, if any, using Eq 2;
Initialize the model parameters, α and π. Evaluate the initial value
of the log likelihood from Eq 4;
while log likelihood difference ≥ ϵ do
  Evaluate γ_ij from Eq 5, using the parameter values and data;
  π_j^new = N_j/N, where, N_j = Σ_{i=1}^{N} γ_ij;
  for i in 1 to N do
    cluster = argmax_j γ_ij;
    Assign data point x_i to cluster z_i;
  end
  for j in 1 to k do
    if cluster j is empty then
      Use initial values of α_j as an update;
    else
      if number of data points in cluster j ≥ 30 then
        α_j^new = α_j^Bayes;
      else
        α_j^new = α_j^MLE;
      end
```

```
    end
  end
  Re-evaluate log likelihood using the new estimates of the
  parameters.
end
```

Simulation study

Comparison with other clustering algorithms

We have done simulation study to check the efficiency of the proposed technique. For a Hard DMM, algorithm 1 and algorithm 2 can be used without any alteration. The objective of our simulation study is to compare the performance of Hard DMM 1 and Hard DMM 2 with other popular clustering algorithms which researchers often use for clustering compositional data. For our study we have considered hierarchical agglomerative clustering with linkage criteria ward, single, average and complete respectively [42]. We have also used partition around medoids (PAM) [43], Clustering Large Applications based on Randomized Search (CLARANS) [44], Fuzzy CMean [45], Kmeans, Gaussian Mixture Model (GMM), Gaussian Mixture Model with hard EM (Hard GMM), spectral clustering [46] and DBSCAN [47] for comparison. We have checked three measures to evaluate the performance.

- Accuracy: The total accurate classifications divided by number of observations.
- Precision: True positives divided by sum of true positives and false positives.
- Recall: True positives divided by the sum of true positives and false negatives.

A detail description of all the measures can be found in [48].
In this section, we have generated data under two schemes.

- **Scheme 1**: 500 random samples from Dirichlet(30,20,10), 100 random samples from Dirichlet (10,20,30) and 300 random samples from Dirichlet (15,15,15).
- **Scheme 2**: 500 random samples from Dirichlet(10,10,3), 100 random samples from Dirichlet (10,20,50), 300 random samples from Dirichlet (15,15,15) and 400 random samples from Dirichlet(0.2,0.5,3)

The data has been generated in python programming language using numpy library [49]. The algorithms of Hard DMM 1, Hard DMM 2 and Hard GMM are also written in python programming language. All the hierarchical clustering algorithms, PAM, KMeans, GMM spectral clustering and DBSCAN algorithm are available in python from scikit-learn, a machine learning library in python [50]. The algorithm for Fuzzy CMean is available in scikit-fuzzy python library [51] and CLARAN is available in PyClustering library [52].

We have generated data under two schemes mentioned above and used different clustering algorithms to find patterns. We have measured the performance of algorithms in terms of accuracy, precision and recall. Fig 1 shows the data generated under scheme 1 with true clusters. And Fig 2 shows how different algorithms finds pattern on the data. We see that Hard DMM 1, Hard DMM 2, KMeans, GMM and Hard GMM recognize pattern in the data somewhat similar to the original pattern. Other algorithms fail to recognize the true patterns. The detailed result can be seen in Table 1. The data generated under scheme 2 is shown in Fig 3. We can see that the data has more complex patterns than data under scheme 1. From Fig 4, we see that only Hard DMM 1 and Hard DMM 2 are able to find pattern similar to the true patterns. All other algorithms fail to understand the true patterns of the generated data. The corresponding results in detail can be seen in Table 2.

Fig 1. Simulated data generated under scheme 1 with true clusters.

https://doi.org/10.1371/journal.pone.0268438.g001

From Fig 5 we see that Hard DMM 1 and Hard DMM 2 have the highest accuracy, precision and recall on data generated under scheme 1. On the other hand from Fig 6 also, we see that Hard DMM 1 and 2 work better than all other algorithms we considered when it comes to the data generated under scheme 2. Hard DMM 1 and Hard DMM 2 have shown best performance in terms of accuracy, precision and recall on our generated dataset. On this note, it is important to understand the interpretation of precision and recall in classification. Precision gives the ratio of number of elements correctly classified under a certain class (or cluster) and all number of elements which actually belong to that class (or cluster). On the other hand, recall gives the ratio of number of elements correctly classified under a certain class (or cluster) and total number of elements classified under that class both correctly and incorrectly. Even though both of these measures are very important, unfortunately we can not maximize both at the same time. When we have different algorithms with same accuracy, we must choose the model with better precision and recall. In our simulation study,both versions of Hard DMM have the best accuracy on both the dataset along with very good precision and recall. Scheme 2 produces a dataset with a complex, non spherical patterns which makes it very difficult to cluster with generic algorithms or mixture model with Gaussian distributions. When compositional data shows asymmetric and non spherical patterns, mixture model using Dirichlet distribution is expected to give better results as Dirichlet distribution can adopt both symmetric and asymmetric shapes. Our simulation study confirms that Hard DMM can be a suitable choice for both spherical and non spherical data when it comes to clustering.

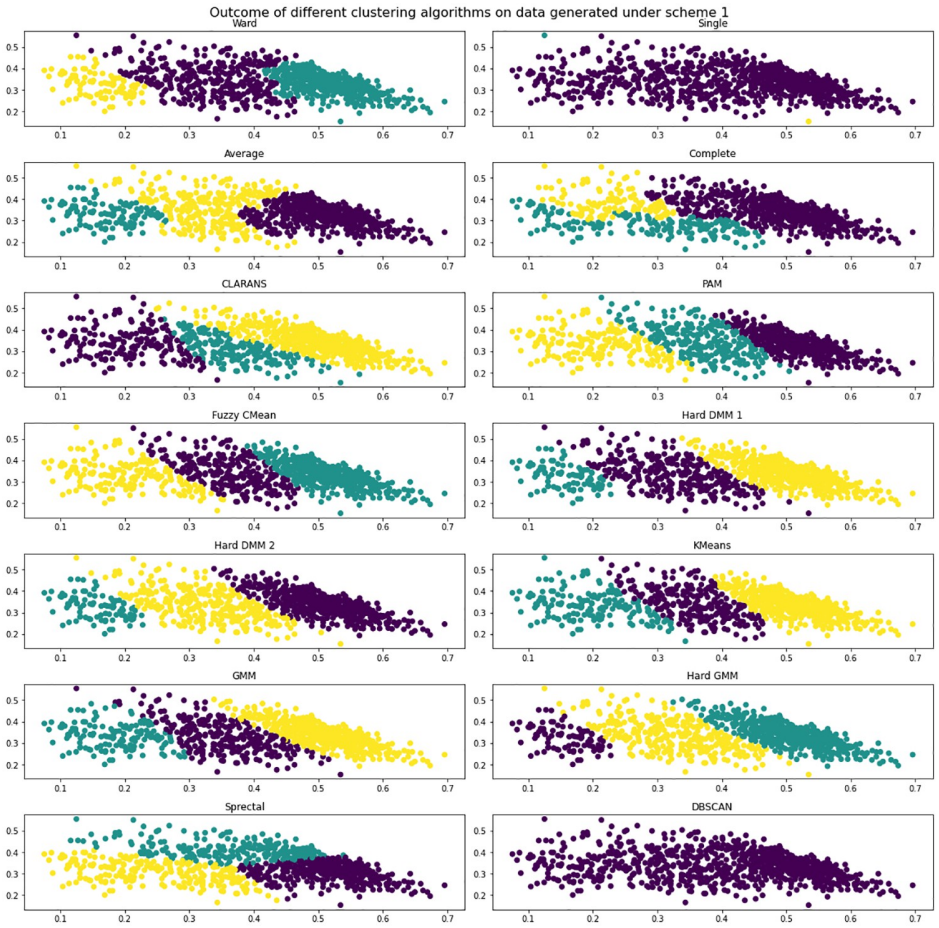

Fig 2. Outcome of different clustering algorithms on simulated data generated under scheme 1.

https://doi.org/10.1371/journal.pone.0268438.g002

Table 1. Accuracy, precision and recall of different clustering algorithms on data generated under scheme 1.

Methods	Accuracy	Precision	Recall
Hard DMM 1	**0.928889**	**0.897778**	**0.917724**
Hard DMM 2	**0.928889**	**0.897778**	**0.917724**
Ward	0.882222	0.877333	0.883416
Single	0.557778	0.337778	0.852264
Average	0.844444	0.854667	0.818948
Complete	0.711111	0.627111	0.615798
CLARANS	0.824444	0.833778	0.767597
PAM	0.805556	0.834889	0.741326
Fuzzy CMean	0.832222	0.854889	0.767149
KMeans	0.852222	0.869556	0.787949
GMM	0.882222	0.891111	0.891111
Hard GMM	0.921111	0.890444	0.910820
Sprectal	0.616667	0.664667	0.664667
DBSCAN	0.555556	0.333333	0.333333

https://doi.org/10.1371/journal.pone.0268438.t001

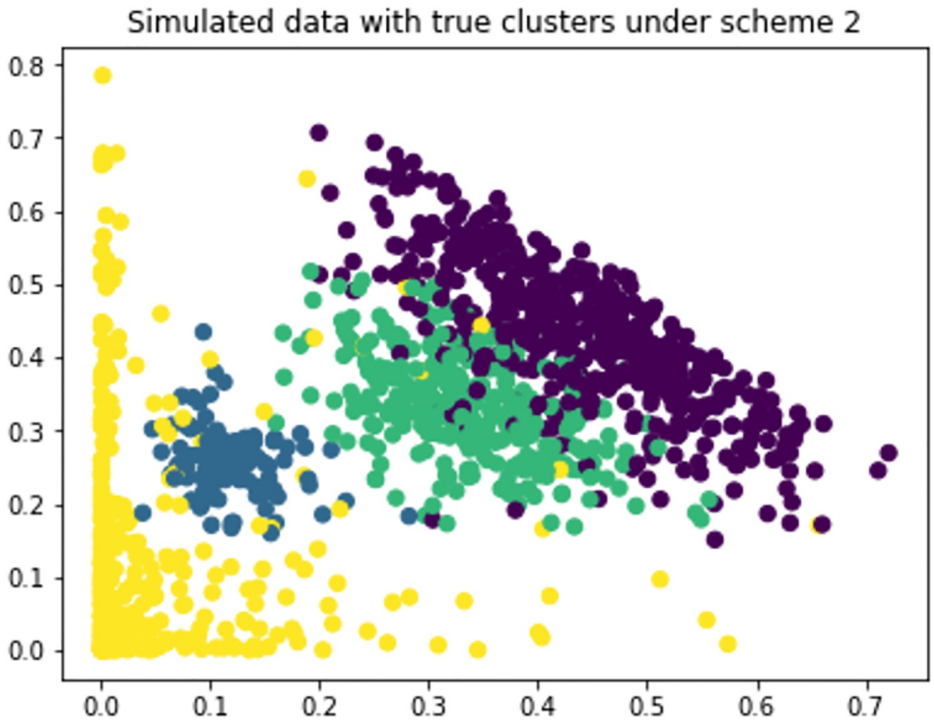

Fig 3. Simulated data generated under scheme 2 with true clusters.

https://doi.org/10.1371/journal.pone.0268438.g003

Performance testing

We wanted to test the performance of our proposed model (Hard DMM 1) for varied dimensions, varied number of clusters and varied overlap. We have simulated data with dimensions 5, 10, 20, 35, 50, 70 and 100 with number of clusters 2 to 6. For each dimension and number of clusters, we have generated data 100 times and used our proposed model to check accuracy. For clusters 2 to 5, we have set very less to none overlap in the data. And for 6 clusters we have introduced overlap in the data with increasing dimension. Let us recall that p denotes the dimension, k denotes the number of cluster and $\alpha_j = (\alpha_{j1}, \ldots, \alpha_{jp})$ denotes the parameters of Dirichlet distribution from mixture component j. The data generating schemes are mentioned below.

- $\boldsymbol{k = 2}$: 800 random samples from a Dirichlet distribution, with p parameters drawn randomly from a range 1 and 110, sorted in ascending order. 500 random samples from a Dirichlet distribution, with p parameters drawn randomly from a range 1 and 110, sorted in descending order.
- $\boldsymbol{k = 3}$: 500 random samples from a Dirichlet distribution, with p parameters drawn randomly from a range 110 and 500, sorted in ascending order. 400 random samples from a Dirichlet distribution, with p parameters drawn randomly from a range 1 and 110, sorted in descending order. 300 random samples from a Dirichlet distribution, with all p parameters equal to 50.
- $\boldsymbol{k = 4}$: 400 random samples from a Dirichlet distribution, with p parameters drawn randomly from a range 1 and 100, sorted in ascending order. 300 random samples from a Dirichlet distribution, with p parameters drawn randomly from a range 1 and 100, sorted in descending

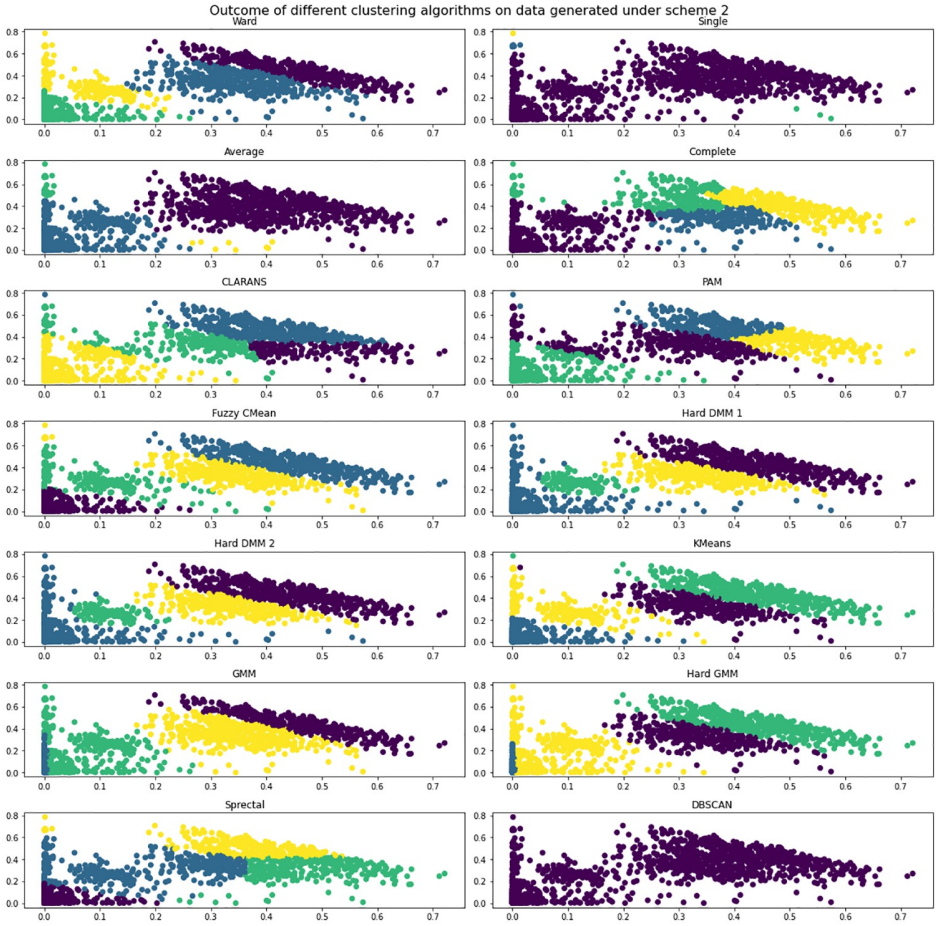

Fig 4. Outcome of different clustering algorithms on simulated data generated under scheme 1.

https://doi.org/10.1371/journal.pone.0268438.g004

Table 2. Accuracy, precision and recall of different clustering algorithms on data generated under scheme 2.

Methods	Accuracy	Precision	Recall
Hard DMM 1	**0.925385**	**0.928167**	**0.909739**
Hard DMM 2	**0.925385**	**0.928167**	**0.909739**
Ward	0.830000	0.858625	0.803385
Single	0.385385	0.250625	0.346824
Average	0.464615	0.495000	0.455818
Complete	0.490000	0.581208	0.511178
CLARANS	0.502308	0.531250	0.454768
PAM	0.510769	0.495250	0.512096
Fuzzy CMean	0.846923	0.877417	0.807752
KMeans	0.858462	0.882708	0.816642
GMM	0.726154	0.786875	0.786875
Hard GMM	0.756154	0.800833	0.764774
Sprectal	0.562308	0.632167	0.632167
DBSCAN	0.384615	0.250000	0.250000

https://doi.org/10.1371/journal.pone.0268438.t002

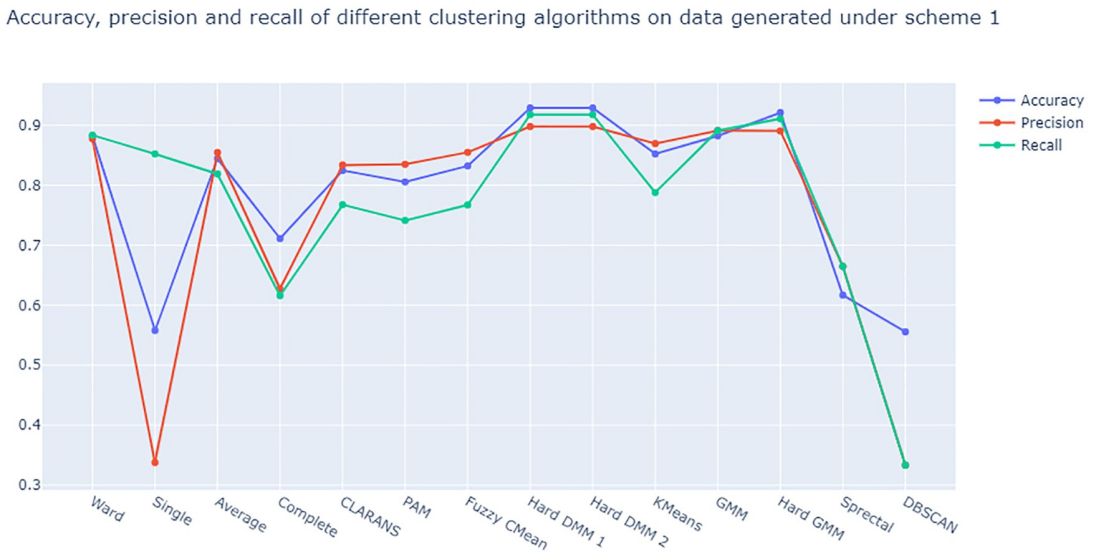

Fig 5. Plot of accuracy, precision and recall of different clustering algorithms on data generated under scheme 1.

https://doi.org/10.1371/journal.pone.0268438.g005

order. 800 random samples from a Dirichlet distribution, with all p parameters equal to 50. 500 random samples from a Dirichlet distribution with l parameters equals to 110 and rest $p - l$ parameters drawn randomly from Uniform (1,5) distribution, sorted in ascending order. l = (2, 3, 5, 8, 12, 15, 18) for p = (5, 10, 20, 35, 50, 70, 100) respectively.

- $\boldsymbol{k = 5}$: 500 random samples from a Dirichlet distribution, with p parameters drawn randomly from a range 110 and 500, sorted in ascending order. 100 random samples from a Dirichlet distribution, with p parameters drawn randomly from a range 110 and 500, sorted in descending order. 300 random samples from a Dirichlet distribution, with p parameters

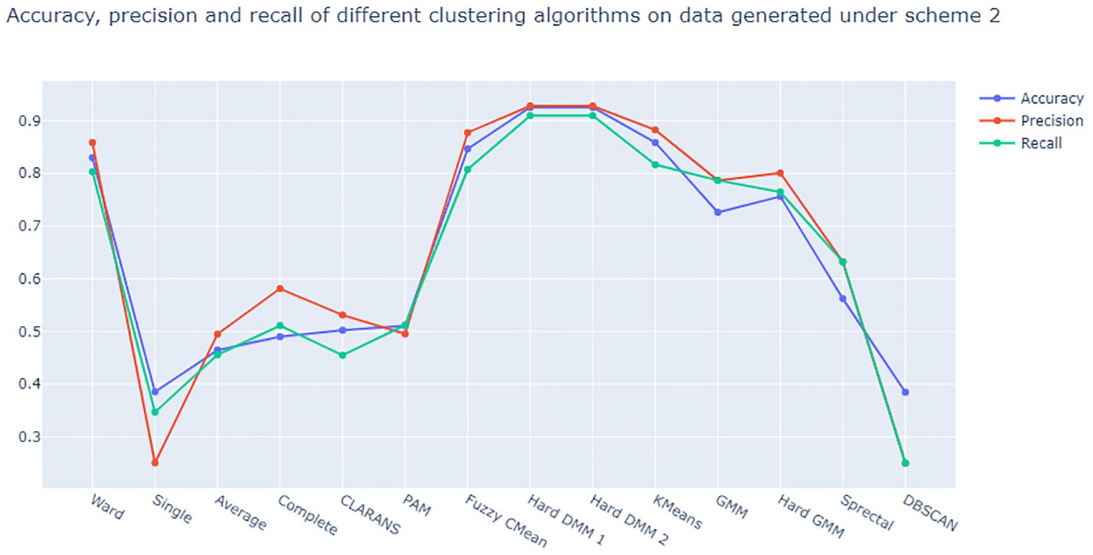

Fig 6. Plot of accuracy, precision and recall of different clustering algorithms on data generated under scheme 2.

https://doi.org/10.1371/journal.pone.0268438.g006

drawn randomly from a range 1 and 110, sorted in ascending order. 400 random samples from a Dirichlet distribution, with p parameters drawn randomly from a range 1 and 110, sorted in descending order. 300 random samples from a Dirichlet distribution, with all p parameters equal to 50.

- $\boldsymbol{k = 6}$: 500 random samples from a Dirichlet distribution, with p parameters drawn randomly from a range 110 and 500, sorted in ascending order. 100 random samples from a Dirichlet distribution, with p parameters drawn randomly from a range 110 and 500, sorted in descending order. 300 random samples from a Dirichlet distribution, with p parameters drawn randomly from a range 1 and 110, sorted in ascending order. 400 random samples from a Dirichlet distribution, with p parameters drawn randomly from a range 1 and 110, sorted in descending order. 300 random samples from a Dirichlet distribution, with all p parameters equal to 50. 500 random samples from a Dirichlet distribution with l parameters equals to 110 and rest $p - l$ parameters drawn randomly from Uniform (1,5) distribution, sorted in ascending order. l = (2, 3, 5, 8, 12, 15, 18) for p = (5, 10, 20, 35, 50, 70, 100) respectively.

The T-SNE [53, 54] plots in Figs 7 and 8 show that with $p = 5$ there is some overlap in one cluster and with $p = 100$ one cluster has completely been overlapped on another. The performances of the model for varied dimension, number of clusters and overlap is shown in Figs 9–13 respectively.

From results in Table 3, we see that increasing dimension and increasing number of clusters do not have much impact on the accuracy of Hard DMM. But increasing overlap has significant impact on the accuracy of Hard DMM. It is to be noted that many algorithms suffer from

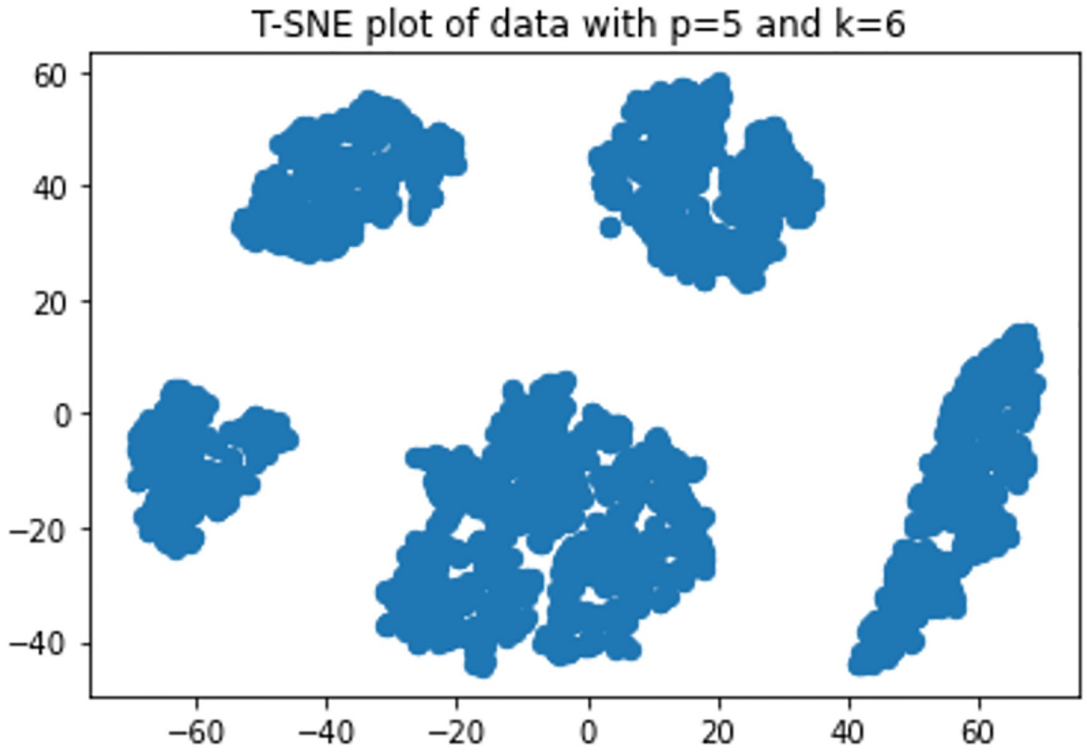

Fig 7. T-SNE plot of data with p = 5 and k = 6.

https://doi.org/10.1371/journal.pone.0268438.g007

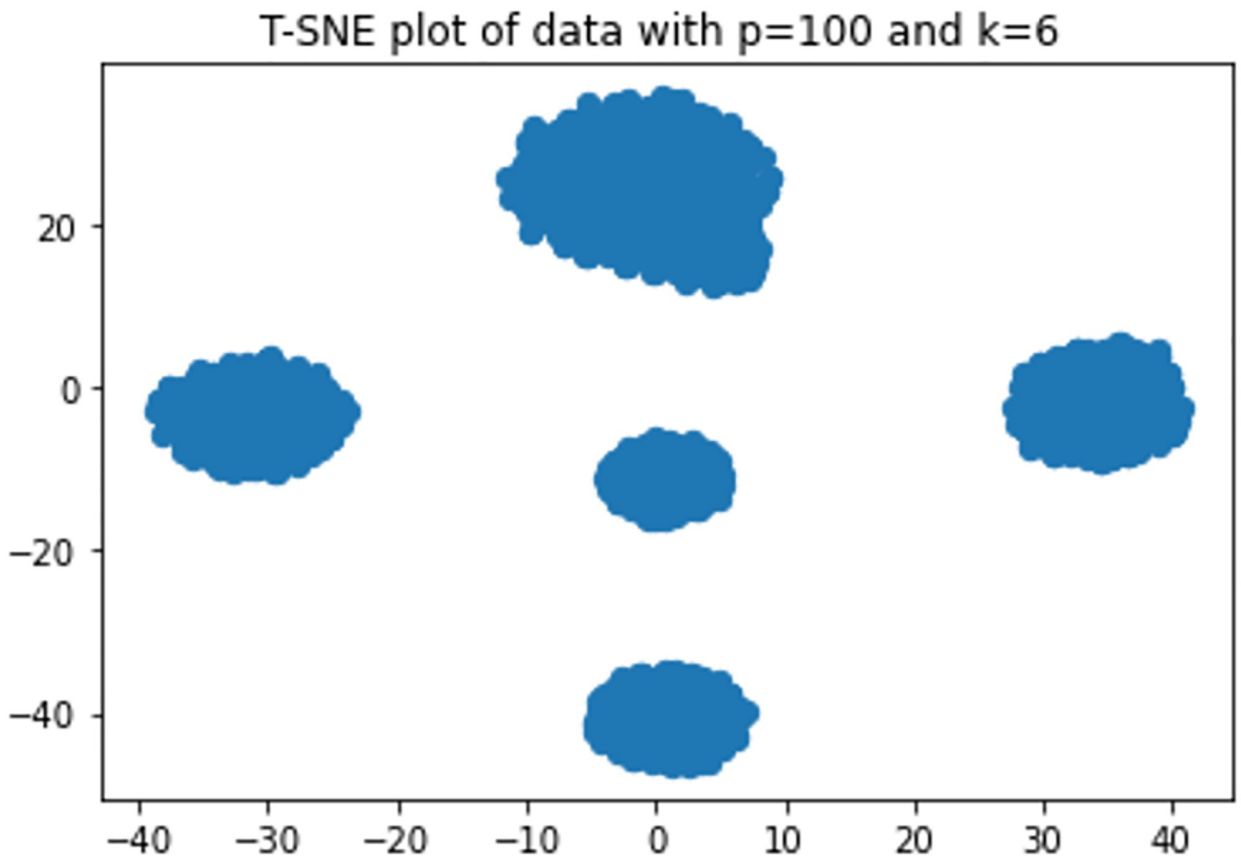

Fig 8. T-SNE plot of data with p = 100 and k = 6.

https://doi.org/10.1371/journal.pone.0268438.g008

"Curse of Dimensionality" with increasing dimension in the data. For example, in case of GMM, a p dimensional mean vector and $p \times p$ symmetric covariance matrix need to be estimated for each k clusters. In other words for a GMM with k clusters and p dimensions, $(k - 1) + kp(1 + (p/2 + 1/2))$ number of parameters need to be estimated. On the other hand, in case of a DMM with k clusters and p dimensions, we need only $(k - 1) + kp$ parameters to estimate. So for 10 clusters and 100 dimensional data GMM estimates 51509 parameters, whereas DMM estimates only 1009 parameters on the same situation. So, DMM has an added advantage over GMM when it comes to high dimensionality. That is why we have noticed in our study that increasing dimension has very little to none impact on the performance of Hard DMM. Also with increasing number of cluster, the number of parameters increase linearly. And with a good starting value in the EM algorithm, the model converges soon with satisfactory results. On the contrary, overlap in the data leads to misclassification, which in turn decreases the performance of Hard DMM significantly.

Real data applications

We have applied the proposed methods on two real data problems. Our main idea was to check how our model works for the given data and not to provide an optimum solution for the problems. We have checked three measures to evaluate the performance, namely: accuracy, precision and recall. All measures have been compared with hierarchical agglomerative

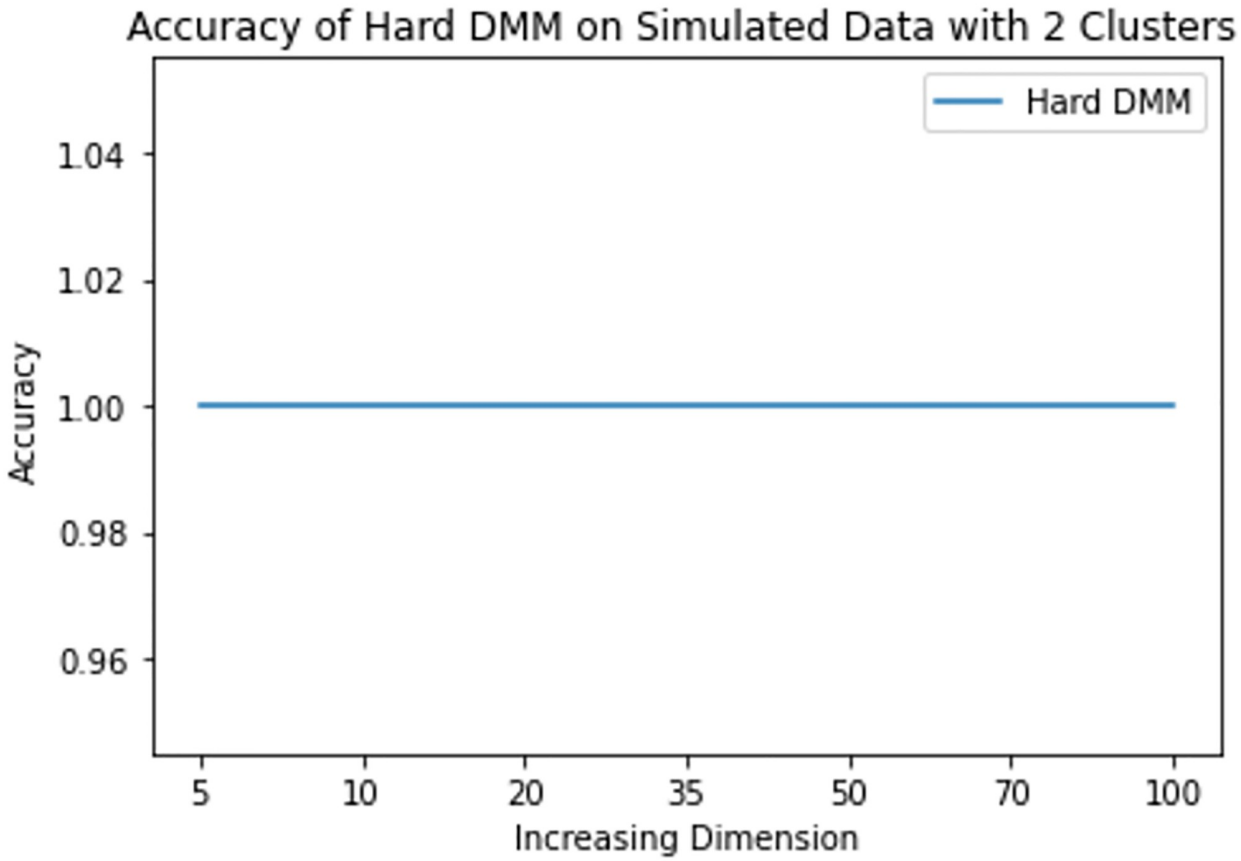

Fig 9. Mean accuracy of hard DMM 1 with 2 clusters and varied dimensions.

https://doi.org/10.1371/journal.pone.0268438.g009

clustering with linkage criteria ward, single, average and complete, PAM, CLARANS, Fuzzy Cmean, KMeans GMM, Hard GMM, Spectral clustering and DBSCAN algorithm. At first we have checked for missing data. In our study there was no missing data. The class labels were subsequently label encoded in order to make it compatible with python. Using L1 normalization [55] the data was then converted into compositional data and zero values were treated using multiplicative replacement which is available in scikit-bio [56], a python library.

Wholesale customers data

We have used a data from Marketing and Management domain for our first experiment. [57] has used this data for logical discriminant models. The dataset can be downloaded from UCI Machine Learning Repository. The data refers to 440 customers of a wholesale distributor, where 298 customers are from the Horeca (Hotel/Restaurant/Café) channel and the rest 142 customers are from the Retail channel. The wholesale customers are grouped in above two classes according to frequency spending degrees of four types:

- low frequency-low spending;
- high frequency-low spending;
- regular frequency-regular spending;
- high frequency-high spending.

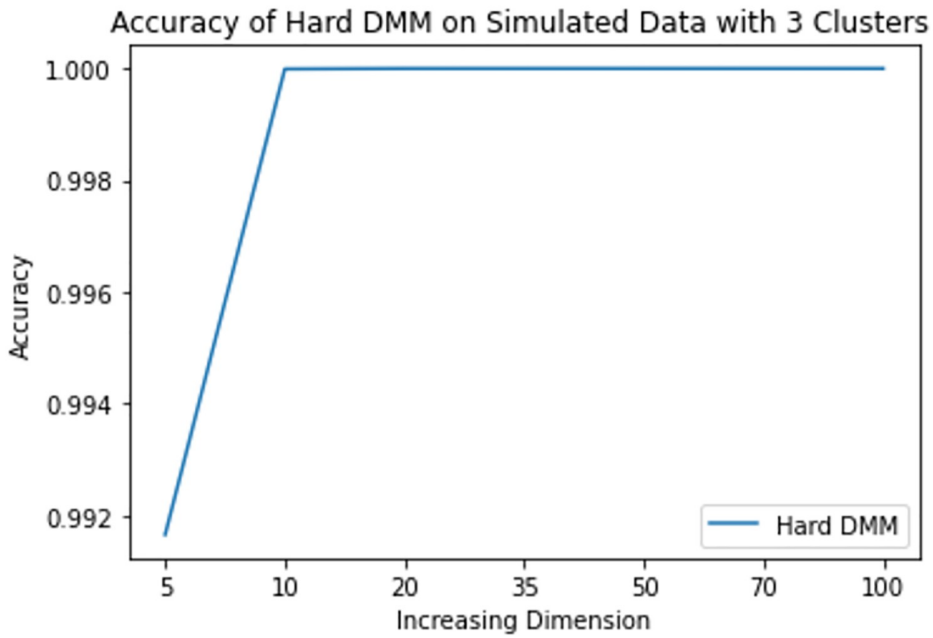

Fig 10. Mean accuracy of hard DMM 1 with 3 clusters and varied dimensions.

https://doi.org/10.1371/journal.pone.0268438.g010

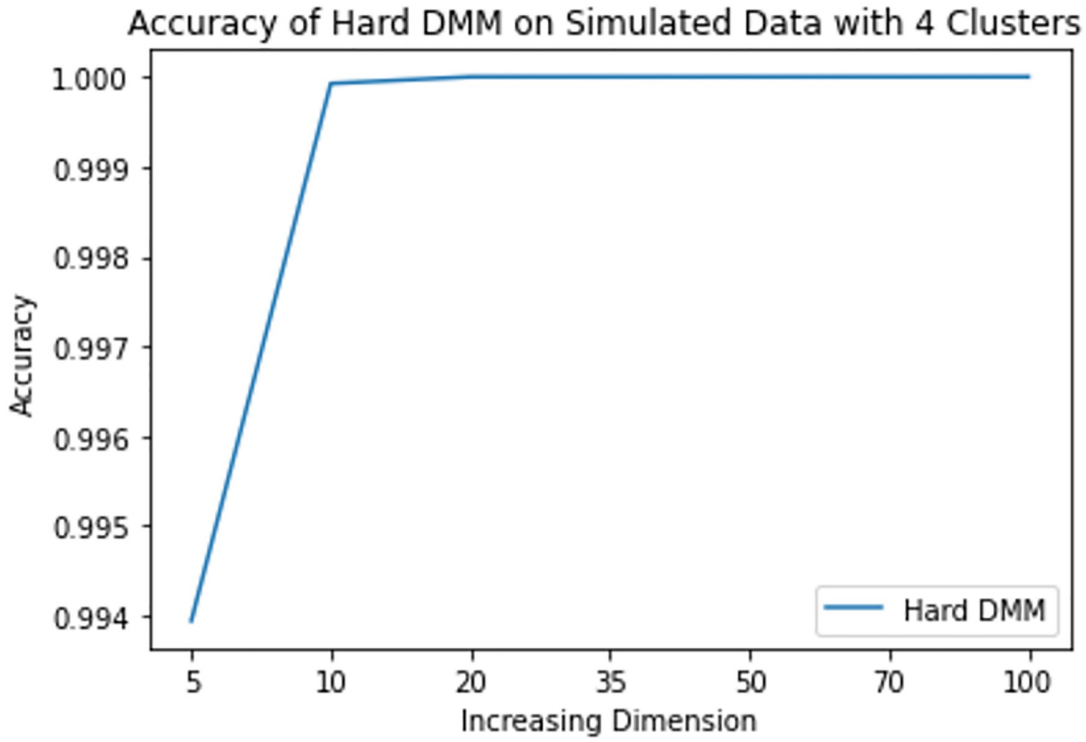

Fig 11. Mean accuracy of hard DMM 1 with 4 clusters and varied dimensions.

https://doi.org/10.1371/journal.pone.0268438.g011

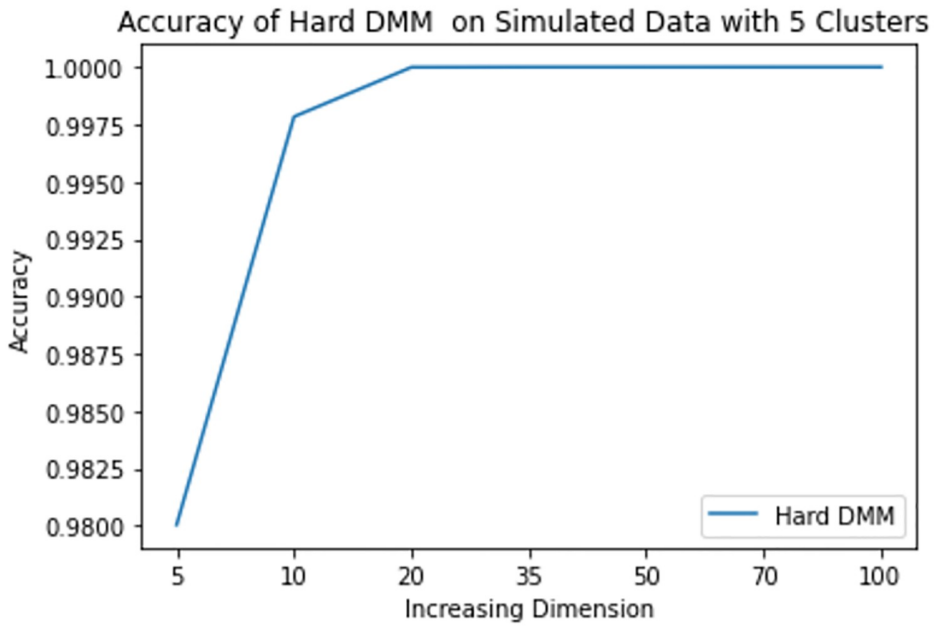

Fig 12. Mean accuracy of hard DMM 1 with 5 clusters and varied dimensions.

https://doi.org/10.1371/journal.pone.0268438.g012

The first two spending patterns are captured by the class Horeca and the second two patterns are captured by the class Retail. The wholesale data concerning the customers consists of annual spending in monetary units (m.u.) on different product categories, namely: fresh products, milk products, grocery, frozen products, detergents and paper products and delicatessen. The summary statistics of the data is shown in Table 4. The aim of the analysis is to find if there are different spending patterns for the two groups of customers and if so, maybe the wholesale could differentiate its marketing actions directed to these groups. For our study we will restrict ourselves to clustering analysis. In this case we have $p = 6$, $k = 2$ and $N = 440$. The T-SNE plots in Fig 14 shows complex distributional patterns. We have used different algorithms for clustering and checked their performances. The corresponding results are shown in Table 5.

From Fig 15 we see that both versions of Hard DMM work better than all other algorithms under consideration in terms of Accuracy. Precision and recall are found to be little better in GMM and Hard GMM than Hard DMM. As discussed before, when there are different algorithms with same level of accuracy, it is better to choose the algorithm with more precision and recall. In this experiment Hard DMM has the highest accuracy with comparatively good precision and recall value. So, Hard DMM can still be considered as a suitable choice in this situation.

Wine data

The dataset we heve chosen for our second experiment is from physical science domain. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. The 13 components are namely: Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenol, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines and Proline. This data

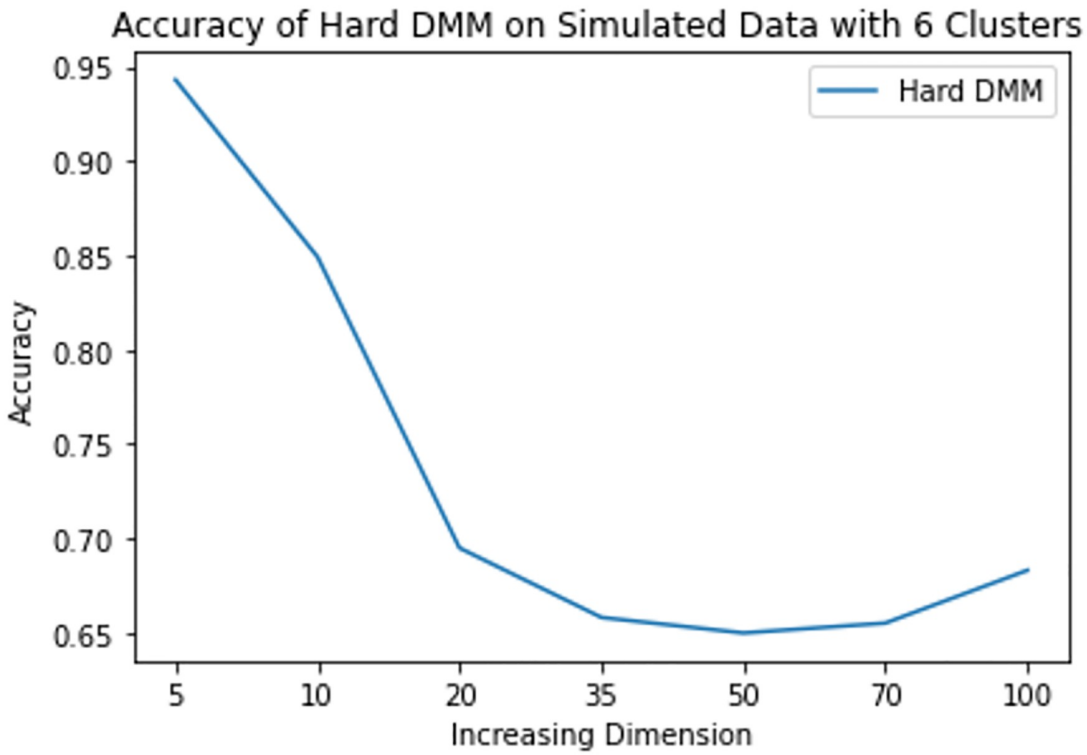

Fig 13. Mean accuracy of hard DMM 1 with 6 clusters, varied dimensions and increasing overlap.

https://doi.org/10.1371/journal.pone.0268438.g013

has been used by many researchers, (see [58–60]). This dataset can also be downloaded from UCI Machine Learning Repository. The aim of the analysis is to identify the different types of wines based on its components. But the attributes color intensity and hue do not constitute chemical components of wine. Hence, we have dropped those two variables for our compositional data analysis. We have used it for clustering purpose. In this case, we have $p = 11$, $k = 3$ and $N = 178$. The summary statistics of the data is shown in Table 6. Fig 16 displays the T-SNE plot of the data which shows difficult cluster patterns. Like before, we have performed clustering with different clustering algorithms. The results in detail is shown in Table 7.

From Fig 17, we see that, both versions of Hard DMM work better than all other clustering algorithms in terms of accuracy, precision and recall. For complex distributional pattern of compositional data, Hard DMM work better than other models as, compositional data can be naturally modelled using Dirichlet distribution.

Table 3. Mean accuracy of hard DMM 1 with varied dimensions and number of clusters.

k	p = 5	p = 10	p = 20	p = 35	p = 50	p = 70	p = 100
2	1.000000	1.000000	1.000000	1.00000	1.00000	1.000000	1.000000
3	0.991658	0.999992	1.000000	1.00000	1.00000	1.000000	1.000000
4	0.993945	0.999925	1.000000	1.00000	1.00000	1.000000	1.000000
5	0.980044	0.997838	0.999994	1.00000	1.00000	1.000000	1.000000
6	0.942976	0.849214	0.695095	0.65829	0.65009	0.655305	0.683252

https://doi.org/10.1371/journal.pone.0268438.t003

Table 4. Summary statistics of wholesale customers data.

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
count	440.000	440.000	440.000	440.000	440.000	440.000	440.000	440.000
mean	1.322	2.543	12000.297	5796.265	7951.277	3071.931	2881.493	1524.870
std	0.468	0.774	12647.328	7380.377	9503.162	4854.673	4767.854	2820.105
min	1.000	1.000	3.000	55.000	3.000	25.000	3.000	3.000
25%	1.000	2.000	3127.750	1533.000	2153.000	742.250	256.750	408.250
50%	1.000	3.000	8504.000	3627.000	4755.500	1526.000	816.500	965.500
75%	2.000	3.000	16933.750	7190.250	10655.750	3554.250	3922.000	1820.250
max	2.000	3.000	112151.000	73498.000	92780.000	60869.000	40827.000	47943.000

https://doi.org/10.1371/journal.pone.0268438.t004

Conclusion

In this paper we have shown a convenient way to build Dirichlet mixture model to cluster compositional data. The model can be used without any transformation. In this case, Dirichlet distribution is a natural choice as it works with the unit sum constraint. Researchers generally use generic algorithms for clustering compositional data, whereas Hard DMM offers an exclusive solution specially for compositional data considering both the spherical and non spherical cluster patterns. Dirichlet distribution is well known for modelling symmetric and asymmetric data. This advantage can be exploited using Hard DMM.

We wanted to use a distribution other than normal in the mixture model and check whether it works as par with predominantly used methods such as GMM and KMeans. From the simulation study and two real data problems we see that when there is a pattern in the composition (proportions), both versions of Hard DMM are able to identify the clusters with quite a satisfactory result. For clustering purpose we had to be cautious while using data used for classification, as not always classification and clustering done on same ground. For example, if images are classified based on presence of a dog in it and the data contains values of red, green and blue channel, clustering algorithm tries to find completely a different pattern.

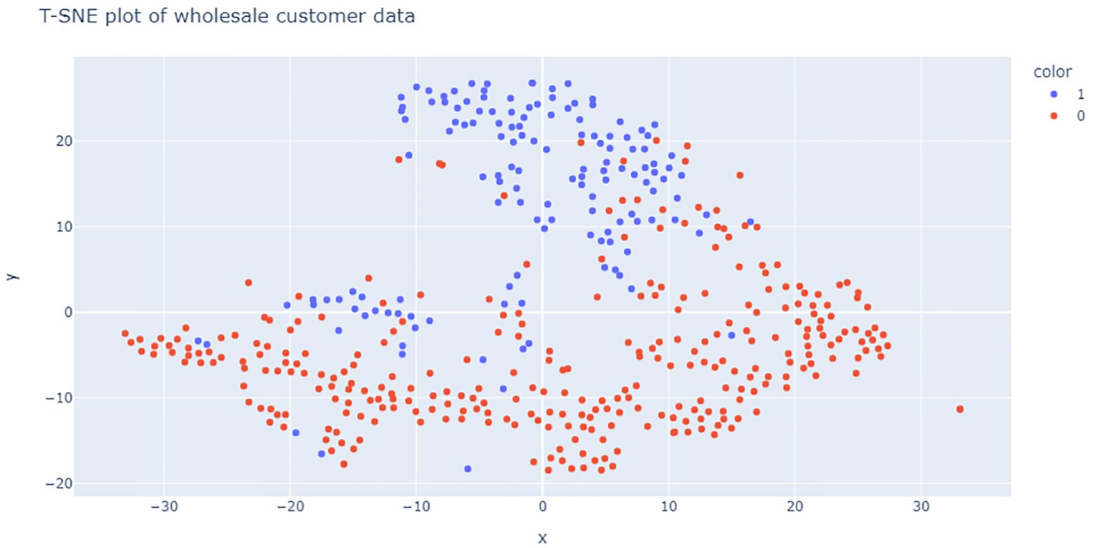

Fig 14. T-SNE plot of wholesale customer data.

https://doi.org/10.1371/journal.pone.0268438.g014

Table 5. Performance comparison of different model based clustering methods on wholesale customers data.

Methods	Accuracy	Precision	Recall
Hard DMM 1	**0.770455**	0.780769	0.750448
Hard DMM 2	**0.770455**	0.780769	0.750448
Ward	0.756818	0.757798	0.732705
Single	0.675000	0.498322	0.338269
Average	0.768182	0.764344	0.741682
Complete	0.406818	0.320612	0.319944
CLARANS	0.718182	0.744021	0.713506
PAM	0.740909	0.755270	0.725436
Fuzzy CMean	0.731818	0.748558	0.718779
KMeans	0.731818	0.748558	0.718779
GMM	0.740909	**0.790292**	**0.790292**
Hard GMM	0.754545	0.789300	0.752955
Sprectal	0.734091	0.750236	0.750236
DBSCAN	0.677273	0.500000	0.500000

https://doi.org/10.1371/journal.pone.0268438.t005

We have also done an extensive simulation study to evaluate the performance of our proposed method. We see that increasing number of dimensions (upto 100) and increasing number of clusters do not seem to have much effect on the performance. But increasing overlap makes the accuracy decrease accordingly. DMM can also be advantageous in high dimensional setting as it requires relatively less number of parameters to be estimated when compared to other mixture model. Due to the complexity of maximum likelihood (ML) estimation for Dirichlet parameters, Dirichlet distribution has long been ignored for clustering purpose. In our study, we have shown a novel way to use ML estimates of dirichlet parameters conveniently in mixture set up which can be used to cluster compositional data.

In our study, we have considered the number of clusters to be known in advance. But in reality, sometimes we need to estimate k before we can start clustering. We have also not

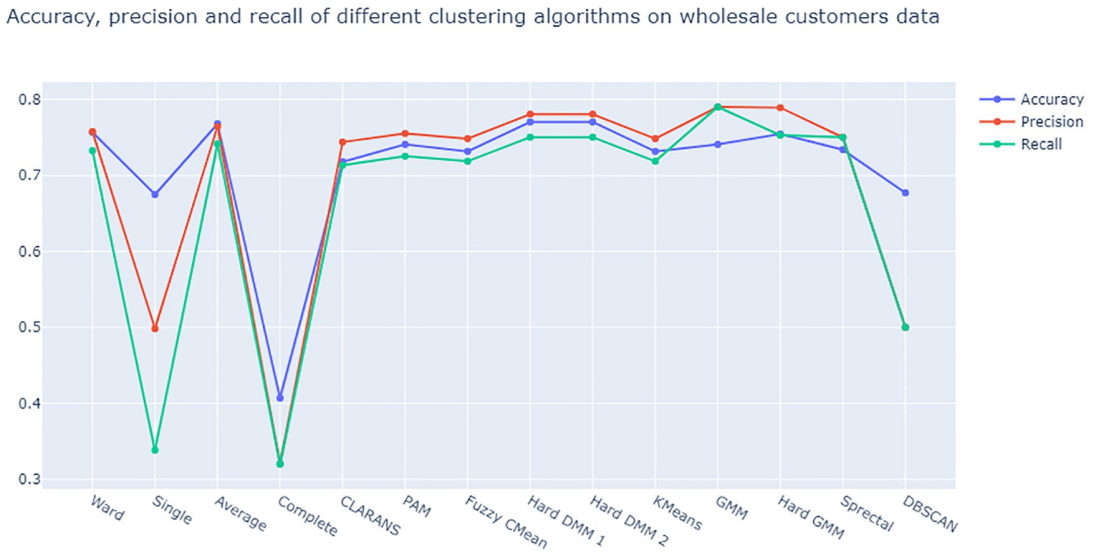

Fig 15. Plot of accuracy, precision and recall of different algorithms on on wholesale customers data.

https://doi.org/10.1371/journal.pone.0268438.g015

Table 6. Summary statistics of wine data.

	0	1	2	3	4	5	6	7	8	9	10	11
count	178.000	178.0000	178.000	178.000	178.000	178.000	178.000	178.000	178.000	178.000	178.000	178.000
mean	1.938	13.000	2.336	2.366	19.494	99.741	2.295	2.029	0.361	1.590	2.611	746.893
std	0.775	0.811	1.117	0.274	3.339	14.282	0.625	0.998	0.124	0.572	0.709	314.907
min	1.000	11.030	0.740	1.360	10.600	70.000	0.980	0.340	0.130	0.410	1.270	278.000
25%	1.000	12.362	1.602	2.210	17.200	88.000	1.742	1.205	0.270	1.250	1.937	500.500
50%	2.000	13.050	1.865	2.360	19.500	98.000	2.355	2.135	0.340	1.555	2.780	673.500
75%	3.000	13.677	3.082	2.557	21.500	107.000	2.800	2.875	0.437	1.950	3.170	985.000
max	3.000	14.830	5.800	3.230	30.000	162.000	3.880	5.080	0.660	3.580	4.000	1680.000

https://doi.org/10.1371/journal.pone.0268438.t006

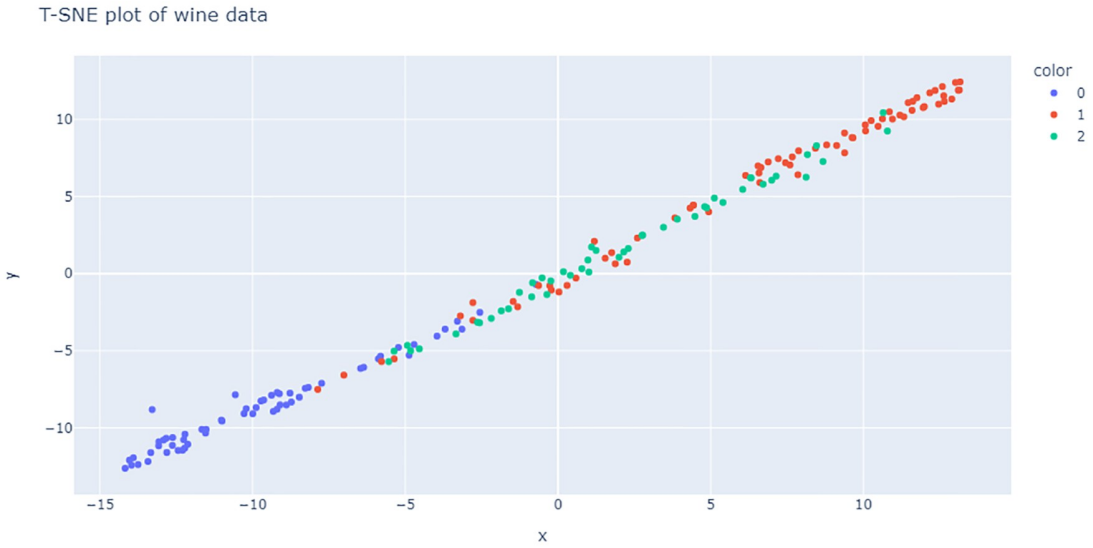

Fig 16. T-SNE plot of wine data.

https://doi.org/10.1371/journal.pone.0268438.g016

Table 7. Performance comparison of different model based clustering methods on wine data.

Methods	Accuracy	Precision	Recall
Hard DMM 1	**0.674157**	**0.693130**	**0.749177**
Hard DMM 2	**0.674157**	**0.693130**	**0.749177**
Ward	0.646067	0.642660	0.689409
Single	0.398876	0.389671	0.427611
Average	0.584270	0.532527	0.521509
Complete	0.516854	0.490513	0.404811
CLARANS	0.612360	0.597660	0.591908
PAM	0.668539	0.672008	0.692386
Fuzzy CMean	0.511236	0.487177	0.490715
KMeans	0.511236	0.487177	0.486869
GMM	0.528090	0.494173	0.494173
Hard GMM	0.539326	0.521899	0.545030
Sprectal	0.612360	0.594180	0.594180
DBSCAN	0.331461	0.333333	0.333333

https://doi.org/10.1371/journal.pone.0268438.t007

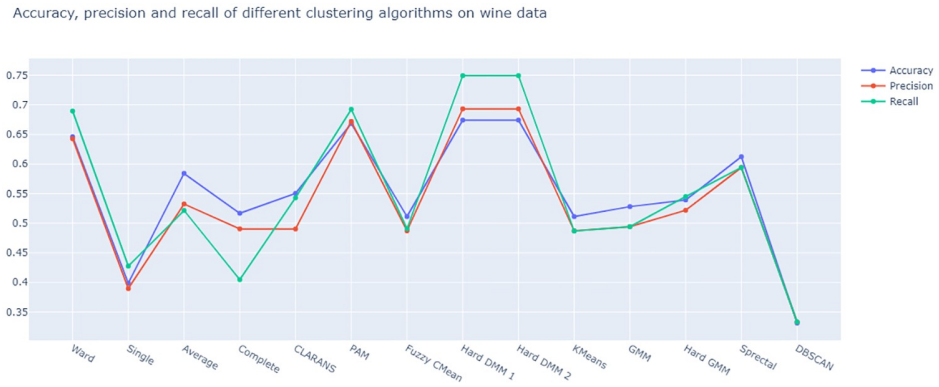

Fig 17. Plot of accuracy, precision and recall of different algorithms on on wine data.

https://doi.org/10.1371/journal.pone.0268438.g017

explored the situation when the data is very high dimensional and sparse. Both the issues would require further research and we keep that for our future work.

Compositional data is getting very popular in biology domain. We hope to see more future applications of Dirichlet distribution and DMM in compositional data analysis.

Author Contributions

Conceptualization: Samyajoy Pal.

Methodology: Samyajoy Pal.

Software: Samyajoy Pal.

Supervision: Christian Heumann.

Writing – original draft: Samyajoy Pal.

Writing – review & editing: Christian Heumann.

References

1. Aitchison J. The statistical analysis of compositional data. Journal of the Royal Statistical Society: Series B (Methodological). 1982; 44(2):139–160.
2. Smith PF, Renner RM, Haslett SJ. Compositional data in neuroscience: If you've got it, log it! Journal of neuroscience methods. 2016; 271:154–159. https://doi.org/10.1016/j.jneumeth.2016.07.008 PMID: 27450923
3. Buccianti A, Tassi F, Vaselli O. Compositional changes in a fumarolic field, Vulcano Island, Italy: a statistical case study. Geological Society, London, Special Publications. 2006; 264(1):67–77. https://doi.org/10.1144/GSL.SP.2006.264.01.06
4. Miesch A, Chapman R. Log transformations in geochemistry. Journal of the International Association for Mathematical Geology. 1977; 9(2):191–198. https://doi.org/10.1007/BF02312512
5. Friedman J, Alm EJ. Inferring correlation networks from genomic survey data. 2012;.
6. Gloor GB, Macklaim JM, Pawlowsky-Glahn V, Egozcue JJ. Microbiome datasets are compositional: and this is not optional. Frontiers in microbiology. 2017; 8:2224. https://doi.org/10.3389/fmicb.2017.02224 PMID: 29187837
7. Godichon-Baggioni A, Maugis-Rabusseau C, Rau A. Clustering transformed compositional data using K-means, with applications in gene expression and bicycle sharing system data. Journal of Applied Statistics. 2019; 46(1):47–65. https://doi.org/10.1080/02664763.2018.1454894
8. Aebischer NJ, Robertson PA, Kenward RE. Compositional analysis of habitat use from animal radio-tracking data. Ecology. 1993; 74(5):1313–1325. https://doi.org/10.2307/1940062

9. Bingham RL, Brennan LA, Ballard BM. Misclassified resource selection: compositional analysis and unused habitat. The Journal of wildlife management. 2007; 71(4):1369–1374. https://doi.org/10.2193/2006-072

10. Belles-Sampera J, Guillen M, Santolino M. Compositional methods applied to capital allocation problems. Journal of Risk, Forthcoming. 2016;. https://doi.org/10.21314/JOR.2016.345

11. DeSarbo WS, Ramaswamy V, Chatterjee R. Analyzing constant-sum multiple criterion data: A segment-level approach. Journal of Marketing Research. 1995; 32(2):222–232. https://doi.org/10.2307/3152050

12. Longford NT, Pittau MG. Stability of household income in European countries in the 1990s. Computational statistics & data analysis. 2006; 51(2):1364–1383. https://doi.org/10.1016/j.csda.2006.02.011

13. Pearson K. Mathematical Contributions to the Theory of Evolution. III. Regression. Heredity, and Panmixia Philosophical Transactions of;.

14. Chayes F. On correlation between variables of constant sum. Journal of Geophysical research. 1960; 65(12):4185–4193. https://doi.org/10.1029/JZ065i012p04185

15. McAlister D. XIII. The law of the geometric mean. Proceedings of the Royal Society of London. 1879; 29 (196-199):367–376. https://doi.org/10.1098/rspl.1879.0061

16. Kotz S, Balakrishnan N, Johnson NL. Continuous multivariate distributions, Volume 1: Models and applications. John Wiley & Sons; 2004.

17. Rehder S, Zier U. Letter to the Editor: Comment on "Logratio Analysis and Compositional Distance" by J. Aitchison, C. Barceló-Vidal, JA Martín-Fernández, and V. Pawlowsky-Glahn. Mathematical Geology. 2001; 33(7):845–848. https://doi.org/10.1023/A:1010902931554

18. Wang H, Liu Q, Mok HM, Fu L, Tse WM. A hyperspherical transformation forecasting model for compositional data. European journal of operational research. 2007; 179(2):459–468.

19. Butler A, Glasbey C. A latent Gaussian model for compositional data with zeros. Journal of the Royal Statistical Society: Series C (Applied Statistics). 2008; 57(5):505–520.

20. Martín-Fernández JA, Barceló-Vidal C, Pawlowsky-Glahn V. Dealing with zeros and missing values in compositional data sets using nonparametric imputation. Mathematical Geology. 2003; 35(3):253–278. https://doi.org/10.1023/A:1023866030544

21. Martín-Fernández JA, Thió-Henestrosa S. Rounded zeros: some practical aspects for compositional data. Geological Society, London, Special Publications. 2006; 264(1):191–201. https://doi.org/10.1144/GSL.SP.2006.264.01.14

22. Little RJ, Rubin DB. Statistical analysis with missing data. vol. 793. John Wiley & Sons; 2019.

23. Wang X, Wang H, Wang S, Yuan J. Convex clustering method for compositional data via sparse group lasso. Neurocomputing. 2021; 425:23–36. https://doi.org/10.1016/j.neucom.2020.10.105

24. Greenacre M. Amalgamations are valid in compositional data analysis, can be used in agglomerative clustering, and their logratios have an inverse transformation. Applied Computing and Geosciences. 2020; 5:100017. https://doi.org/10.1016/j.acags.2019.100017

25. McLachlan G, Peel D. Finite Mixture Models.,(John Wiley & Sons: New York.). 2000;.

26. Ward JH, Jr. Hierarchical grouping to optimize an objective function. Journal of the American statistical association. 1963; 58(301):236–244. https://doi.org/10.1080/01621459.1963.10500845

27. MacQueen J, et al. Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. vol. 1. Oakland, CA, USA; 1967. p. 281–297.

28. Rau A, Maugis-Rabusseau C. Transformation and model choice for RNA-seq co-expression analysis. Briefings in bioinformatics. 2018; 19(3):425–436. PMID: 28065917

29. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological). 1977; 39(1):1–22.

30. Baudry JP, Celeux G. EM for mixtures. Statistics and computing. 2015; 25(4):713–726. https://doi.org/10.1007/s11222-015-9561-x

31. McLachlan GJ, Peel D. Finite mixture models. John Wiley & Sons; 2004.

32. Ma J, Jiang X, Jiang J, Gao Y. Feature-guided Gaussian mixture model for image matching. Pattern Recognition. 2019; 92:231–245. https://doi.org/10.1016/j.patcog.2019.04.001

33. Gebru ID, Alameda-Pineda X, Forbes F, Horaud R. EM algorithms for weighted-data clustering with application to audio-visual scene analysis. IEEE transactions on pattern analysis and machine intelligence. 2016; 38(12):2402–2415. https://doi.org/10.1109/TPAMI.2016.2522425 PMID: 27824582

34. Chung H, Loken E, Schafer JL. Difficulties in drawing inferences with finite-mixture models: a simple example with a simple solution. The American Statistician. 2004; 58(2):152–158. https://doi.org/10.1198/0003130043286

35. Neal RM, Hinton GE. A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Learning in graphical models. Springer; 1998. p. 355–368.

36. Murphy KP. Machine learning: a probabilistic perspective. MIT press; 2012.

37. McLachlan GJ, Krishnan T. The EM algorithm and extensions. vol. 382. John Wiley & Sons; 2007.

38. Kearns M, Mansour Y, Ng AY. An information-theoretic analysis of hard and soft assignment methods for clustering. In: Learning in graphical models. Springer; 1998. p. 495–520.

39. Minka T. Estimating a Dirichlet distribution; 2000.

40. Zhanyu Ma. Bayesian estimation of the dirichlet distribution with expectation propagation. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 689–693. IEEE, 2012.

41. Mohammed M EnasAbidAlhafidh and Al-Tebawy Alaa Adnan Aoda. Bayesian estimation of the beta distribution parameter (α) when the parameter (β) is known. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(14):4879–4886, 2021.

42. Nielsen F. Hierarchical clustering. In: Introduction to HPC with MPI for Data Science. Springer; 2016. p. 195–211.

43. Kaufman L, Rousseeuw PJ. Finding groups in data: an introduction to cluster analysis. vol. 344. John Wiley & Sons; 2009.

44. Ng RT, Han J. CLARANS: A method for clustering objects for spatial data mining. IEEE transactions on knowledge and data engineering. 2002; 14(5):1003–1016. https://doi.org/10.1109/TKDE.2002.1033770

45. Bezdek JC, Ehrlich R, Full W. FCM: The fuzzy c-means clustering algorithm. Computers & geosciences. 1984; 10(2-3):191–203. https://doi.org/10.1016/0098-3004(84)90020-7

46. Bengio Y, Delalleau O, Roux NL, Paiement JF, Vincent P, Ouimet M. Learning eigenfunctions links spectral embedding and kernel PCA. Neural computation. 2004; 16(10):2197–2219. https://doi.org/10.1162/0899766041732396 PMID: 15333211

47. Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

48. Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. Information processing & management. 2009; 45(4):427–437. https://doi.org/10.1016/j.ipm.2009.03.002

49. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. Nature. 2020; 585(7825):357–362. https://doi.org/10.1038/s41586-020-2649-2 PMID: 32939066

50. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011; 12:2825–2830.

51. Warner J, Sexauer J, scikit fuzzy, twmeggs, alexsavio, Unnikrishnan A, et al. JDWarner/scikit-fuzzy: Scikit-Fuzzy version 0.4.2; 2019.

52. Novikov A. PyClustering: Data Mining Library. Journal of Open Source Software. 2019; 4(36):1230. https://doi.org/10.21105/joss.01230

53. Hinton G, Roweis ST. Stochastic neighbor embedding. In: NIPS. vol. 15. Citeseer; 2002. p. 833–840.

54. Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of machine learning research. 2008; 9(11).

55. Luo X, Chang X, Ban X. Regression and classification using extreme learning machine based on L1-norm and L2-norm. Neurocomputing. 2016; 174:179–186. https://doi.org/10.1016/j.neucom.2015.03.112

56. scikit-bio development team T. scikit-bio: A Bioinformatics Library for Data Scientists, Students, and Developers; 2020. Available from: http://scikit-bio.org.

57. Cardoso MG. Logical discriminant models. In: Quantitative Modelling In Marketing And Management. World Scientific; 2013. p. 223–253.

58. Zhong P, Fukushima M. Regularized nonsmooth Newton method for multi-class support vector machines. Optimisation Methods and Software. 2007; 22(1):225–236. https://doi.org/10.1080/10556780600834745

59. Fischer I, Poland J. Amplifying the block matrix structure for spectral clustering. In: Proceedings of the 14th annual machine learning conference of Belgium and the Netherlands. Citeseer; 2005. p. 21–28.

60. Basu S. Semi-supervised clustering with limited background knowledge. In: AAAI; 2004. p. 979–980.

Chapter 3

Gene Coexpression Analysis with Dirichlet Mixture Model: Accelerating Model Evaluation Through Closed-Form KL Divergence Approximation Using Variational Techniques

Summary

This study introduces a novel application of the Dirichlet Mixture Model (DMM) for clustering compositional data, with a focus on improving the efficiency and accuracy of model evaluation. The research addresses the computational challenges of using Kullback-Leibler (KL) Divergence in DMMs by proposing a new variational approach that offers a closed-form solution. This method significantly enhances computational speed and robustness compared to traditional Monte Carlo methods. Although applied to gene coexpression analysis, the statistical advancements presented can be broadly applied to various fields requiring efficient clustering of compositional data.

Contributing Article

Pal, Samyajoy, and Christian Heumann. "Gene coexpression analysis with Dirichlet mixture model: accelerating model evaluation through closed-form KL divergence approximation using variational techniques." In *International Workshop on Statistical Modelling*, pp. 134-141. Cham: Springer Nature Switzerland, 2024. https://doi.org/10.1007/978-3-031-65723-8_21.

Author Contributions

- Samyajoy Pal: Conceptualization, Methodology, Software, Writing – original draft
- Christian Heumann: Supervision, Writing – review & editing

Gene Coexpression Analysis with Dirichlet Mixture Model: Accelerating Model Evaluation Through Closed-Form KL Divergence Approximation Using Variational Techniques

Samyajoy Pal( ) and Christian Heumann

LMU Munich, Ludwigstr. 33, 80539 Munich, Germany
Samyajoy.Pal@stat.uni-muenchen.de

Abstract. Gene coexpression analysis poses unique challenges, particularly in clustering normalized gene profiles where dedicated algorithms are lacking. Compositional in nature, normalized gene profiles find a fitting solution in the Dirichlet Mixture Model (DMM). This study pioneers the application of DMM for clustering normalized gene profiles, recognizing the necessity for efficient model evaluation. Central to this evaluation is the Kullback-Leibler (KL) Divergence, a critical metric for DMMs. In addressing the computational challenges associated with KL Divergence in DMMs, we introduce a novel variational approach. This method provides a closed-form solution, markedly improving computational efficiency for rapid model comparisons and robust estimation evaluations. Through validation on real and simulated data, our approach demonstrates superior efficiency and accuracy compared to traditional Monte Carlo-based methods. This innovation opens new frontiers for expeditious exploration of diverse DMM models, propelling advancements in the statistical analysis of compositional gene expression data.

Keywords: Gene Coexpression · Dirichlet Mixture Model (DMM) · Normalized Gene Profiles · Kullback-Leibler Divergence · Variational Approach

1 Introduction

RNA-sequence (RNA-seq) data often come with read counts or pseudo-counts, representing the number of reads aligned to specific genes or genomic regions. When using normalized expression profiles, which indicate the proportion of normalized counts for each feature, the data become compositional. Dirichlet Mixture Model (DMM) has proven to be more effective than other clustering algorithms for compositional data [1]. The Kullback-Leibler (KL) Divergence [2]

J. Einbeck et al. (Eds.): IWSM 2024, CONTRIB.STAT., pp. 134–141, 2024.
https://doi.org/10.1007/978-3-031-65723-8_21

stands as a fundamental measure in statistics, quantifying the statistical distance between probability distributions. The Kullback-Leibler (KL) divergence (also recognized as relative entropy) between two probability density functions $f(x)$ and $g(x)$ is defined by the integral expression:

$$D(f\|g) \stackrel{\text{def}}{=} \int f(x) \log\left(\frac{f(x)}{g(x)}\right) \mathrm{dx} \ . \tag{1}$$

It operates as a measure of the dissimilarity between the probability distributions encoded by $f(x)$ and $g(x)$. In statistical inference, generally $f(x)$ is the distribution with true parameter values and $g(x)$ is the distribution with estimated parameter values. The KL divergence exhibits fundamental properties known as divergence properties: self-similarity, self-identification, and positivity. These properties underscore the significance of KL divergence in capturing the nuances of distributional disparities, making it a cornerstone in statistical analyses. While closed-form solutions for KL Divergence exist for the Dirichlet distribution, there is no analytically tractable solution available for DMM. This study addresses these challenges by proposing a variational approach to approximate KL Divergence in DMMs. Unlike other methods such as Monte Carlo based approximation [3], our approach provides a closed-form solution, substantially enhancing computational efficiency. Validation using real and simulated data demonstrates its superiority in efficiency and accuracy over traditional Monte Carlo-based approaches. The results underscore the potential of our variational approximation to accelerate the estimation process while improving the quality of estimates.

2 Methods

Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N$ denote a random sample of size N, where $\boldsymbol{X}_i$ is a p dimensional random vector with probability density function $f(\boldsymbol{x}_i)$ on $\mathbb{R}^p$.

The Dirichlet density is given by

$$f(\boldsymbol{x}_i) = \frac{\Gamma(\sum_{m=1}^{p} \alpha_m)}{\prod_{m=1}^{p} \Gamma(\alpha_m)} \prod_{m=1}^{p} x_{im}^{\alpha_m - 1} \ , \tag{2}$$

where $\sum_{m=1}^{p} x_{im} = 1, x_{im}\text{'s} > 0$, $\alpha_{jm}\text{'s} > 0$ and $\Gamma(\cdot)$ denotes a gamma function. The density of a mixture model with k mixture components for one observation $\boldsymbol{x}_i$ is given by the mixture density

$$p(\boldsymbol{x}_i) = \sum_{j=1}^{k} \pi_j f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j) \ , \tag{3}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$ contains the corresponding mixture proportions with $\sum_{j=1}^{k} \pi_j = 1$, $0 < \pi_j < 1$. The density component of mixture j is given by $f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j)$ and $\boldsymbol{\alpha}_j$, $j = 1, 2, ..., k$ is the vector of component specific parameters

for each density. Then $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)$ denotes the vector of all parameters of the model.

The parameters of Dirichlet mixture model, can be estimated using an Expectation-Maximization (EM) algorithm [4]. In our previous work [1] we presented a Hard version [5] of the EM algorithm to estimate the parameters. The log likelihood of the model for a sample of size N is then given by

$$\log p(\boldsymbol{x_1}, \ldots, \boldsymbol{x_N} | \boldsymbol{\alpha}, \pi) = \sum_{i=1}^{N} \log \left[\sum_{j=1}^{k} \pi_j f(\boldsymbol{x_i} | \boldsymbol{\alpha_j}) \right] . \tag{4}$$

Latent variables Z_i's are introduced, which are categorical variables taking on values $1, \ldots, k$ with probabilities $\pi_1, \ldots, \pi_k$ such that $Pr(\boldsymbol{X_i}|Z_i = j) = f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j)$, $j = 1, \ldots, k$. The cluster membership probabilities of data point i for cluster j can be obtained by γ_{ij}, where,

$$\gamma_{ij}(x_i) = Pr(Z_i = j | \boldsymbol{x_i}, \boldsymbol{\alpha_j}) = \frac{\pi_j f(\boldsymbol{x_i}|\boldsymbol{\alpha_j})}{\sum_{j=1}^{k} \pi_j f_(\boldsymbol{x_i}|\boldsymbol{\alpha_j})} . \tag{5}$$

Hard EM maximizes the classification log likelihood; it applies a delta function approximation to the posterior probabilities $Pr(Z_i = j|\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{\alpha})$, where $Z_i, i = 1, \ldots, N$ are the latent variables representing class labels. The approximation changes the E step as follows,

$$Pr(Z_i = j|\boldsymbol{x_i}, \boldsymbol{\alpha_j}) \approx \mathbb{I}(j = z_i^*), \tag{6}$$

where, $z_i^* = \underset{j}{\operatorname{argmax}}\, \gamma_{ij}$. γ_{ij}'s are nothing but the responsibilities (probabilities) for each data point that belongs to different clusters. In other words, it introduces a classification step, where all the data points are classified into different clusters based on the posterior probability. Let N_j be the number of data points in cluster j. Then, the ML estimates of π_j is obtained as $\frac{N_j}{N}$. And estimates of $\boldsymbol{\alpha_j}$'s are obtained by finding the MLE using N_j data points in cluster j. However, as the solution is not analytically tractable, some numerical methods need to be used. It is generally done by fixed-point iteration. The iterative equation that needs to be solved is given by

$$\Psi(\alpha_{jm}^{new}) = \Psi(\sum_{m=1}^{p} \alpha_{jm}^{old}) + \frac{1}{N_j} \sum_{i=1}^{N_j} \log(x_{im}) , \tag{7}$$

where, Ψ is the digamma function. The inverse of diggama function is used to get the estimates of α_{jm}. Now, let's explore various approximations to compute the KL distance for DMM.

2.1 KL Divergence: Monte Carlo Approach

Monte Carlo sampling is frequently employed by researchers to compute KL divergence. Let f_a and g_b be two DMMs (see Eq. 3). The approach involves

drawing a sample $\boldsymbol{x_i}$ from the probability density function f_a such that,

$$E_{f_a}[\log f_a(\boldsymbol{x_i})/g_b(\boldsymbol{x_i})] = D(f_a \| g_b) \ .$$

Utilizing N independent and identically distributed samples $\{\boldsymbol{x_i}\}_{i=1}^N$, the estimation is expressed as follows:

$$D_{\mathrm{MC}}(f_a \| g_b) = \frac{1}{N} \sum_{i=1}^{N} \log f_a(x_i)/g_b(x_i) \to D(f_a \| g_b) \ , \tag{8}$$

as $N \to \infty$. The variance of the estimation error is $\frac{1}{N}\mathrm{Var}_{f_a}[\log f_a/g_b]$. To compute $D_{\mathrm{MC}}(f\|g)$, it is necessary to generate the independent and identically distributed samples $\{\boldsymbol{x_i}\}_{i=1}^N$ from f_a. Drawing a sample $\boldsymbol{x_i}$ from the DMM f_a involves initially drawing a discrete sample a_i according to the probabilities π_a. Subsequently, a continuous sample $\boldsymbol{x_i}$ is drawn from the resulting Dirichlet component $f_{a_i}(\boldsymbol{x})$. The Monte Carlo method satisfies the similarity property; however, the positivity property does not hold. The identification property is likely to fail only under highly artificial circumstances and with very low probability.

2.2 KL Divergence: Variational Approach

Let us denote $\langle . \rangle$ as the inner product. In our context $\langle . \rangle_{\boldsymbol{x}}$ means expectation with respect to $\mathbf{x}$.

Theorem 1. *Let $\boldsymbol{X}$ be an $p \times 1$ random vector. Assume two Dirichlet distributions u and v specifying the probability distribution of $\boldsymbol{X}$ as,*

$$\begin{aligned} u: \ \boldsymbol{X} &\sim \mathrm{Dir}(\alpha_{11}, \ldots, \alpha_{1p}) \\ v: \ \boldsymbol{X} &\sim \mathrm{Dir}(\alpha_{21}, \ldots, \alpha_{2p}) \ . \end{aligned}$$

Then, the Kullback-Leibler divergence of u from v is given by,

$$\begin{aligned} \mathrm{D}(u \,||\, v) = & \log \frac{\Gamma\left(\sum_{i=1}^{p} \alpha_{1i}\right)}{\Gamma\left(\sum_{i=1}^{p} \alpha_{2i}\right)} + \sum_{i=1}^{p} \log \frac{\Gamma(\alpha_{2i})}{\Gamma(\alpha_{1i})} \\ & + \sum_{i=1}^{p} (\alpha_{1i} - \alpha_{2i}) \left[\psi(\alpha_{1i}) - \psi\left(\sum_{i=1}^{p} \alpha_{1i} \right) \right] \ . \end{aligned} \tag{9}$$

Proof. For Dirichlet distributions KL divergence is,

$$\begin{aligned} \mathrm{D}(u \,||\, v) &= \int_{\boldsymbol{X}^p} \mathrm{Dir}(\boldsymbol{x}; \alpha_{11}, \ldots, \alpha_{1p}) \, \log \frac{\mathrm{Dir}(\boldsymbol{x}; \alpha_{11}, \ldots, \alpha_{1p})}{\mathrm{Dir}(\boldsymbol{x}; \alpha_{21}, \ldots, \alpha_{2p})} \, \mathrm{d}\boldsymbol{x} \\ &= \left\langle \log \frac{\mathrm{Dir}(x; \alpha_{11}, \ldots, \alpha_{1p})}{\mathrm{Dir}(x; \alpha_{21}, \ldots, \alpha_{2p})} \right\rangle_{u(\boldsymbol{x})} \end{aligned} \tag{10}$$

We can do some algebraic manipulations using the probability density function of the Dirichlet distribution to get,

$$\mathrm{D}(u \,||\, v) = \left\langle \log \frac{\frac{\Gamma\left(\sum_{i=1}^{p} \alpha_{1i}\right)}{\prod_{i=1}^{p} \Gamma(\alpha_{1i})} \prod_{i=1}^{p} {x_i}^{\alpha_{1i}-1}}{\frac{\Gamma\left(\sum_{i=1}^{p} \alpha_{2i}\right)}{\prod_{i=1}^{p} \Gamma(\alpha_{2i})} \prod_{i=1}^{p} {x_i}^{\alpha_{2i}-1}} \right\rangle_{u(\boldsymbol{x})}$$
$$= \log \frac{\Gamma\left(\sum_{i=1}^{p} \alpha_{1i}\right)}{\Gamma\left(\sum_{i=1}^{p} \alpha_{2i}\right)} + \sum_{i=1}^{p} \log \frac{\Gamma(\alpha_{2i})}{\Gamma(\alpha_{1i})} + \sum_{i=1}^{p} (\alpha_{1i} - \alpha_{2i}) \cdot \langle \log x_i \rangle_{u(\boldsymbol{x})} \ .$$

Moreover,

$$\langle \log \boldsymbol{x}_i \rangle = \psi(\alpha_i) - \psi\left(\sum_{i=1}^{p} \alpha_i\right) \ .$$

Thus the Kullback-Leibler divergence of u from v becomes:

$$\mathrm{D}(u \,||\, v) = \log \frac{\Gamma\left(\sum_{i=1}^{p} \alpha_{1i}\right)}{\Gamma\left(\sum_{i=1}^{p} \alpha_{2i}\right)} + \sum_{i=1}^{p} \log \frac{\Gamma(\alpha_{2i})}{\Gamma(\alpha_{1i})} + \sum_{i=1}^{p} (\alpha_{1i} - \alpha_{2i}) \left[\psi(\alpha_{1i}) - \psi\left(\sum_{i=1}^{p} \alpha_{1i}\right)\right].$$

Proposition 1. *Let f_a and g_b be two DMMs such that,*

$$f_a = f(x) = \sum_a \pi_a Dir(\boldsymbol{x}; \boldsymbol{\alpha}_a)$$
$$g_b = g(x) = \sum_b \omega_b Dir(\boldsymbol{x}; \boldsymbol{\alpha}_b)$$

Then using a variational approach, an approximated KL divergence can be expressed as,

$$D_{\text{variational}}(f \| g) = \sum_a \pi_a \log \frac{\sum_{a'} \pi_{a'} e^{-D(f_a \| f_{a'})}}{\sum_b \omega_b e^{-D(f_a \| g_b)}}. \tag{11}$$

Proof. [6] has provided a similar proof for Gaussian Mixture Models. That approach can be adopted for DMMs as well. We can define the log-likelihood,$L_f(g)$ as, $L_f(g) = \langle [\log g(\boldsymbol{x})] \rangle_{f(\boldsymbol{x})}$. Then KL divergence can be written in terms of log-likelihood in the following way.

$$D(f \| g) = L_f(f) - L_f(g) \ . \tag{12}$$

We define variational parameters,$\phi_{b|a} > 0$ such that $\Sigma_b \phi_{b|a} = 1$. Using Jensen's inequality we can show,

$$
\begin{aligned}
L_f(g) &\stackrel{\text{def}}{=} \langle \log g(x) \rangle_{f(\boldsymbol{x})} \\
&= \left\langle \log \sum_b \omega_b g_b(x) \right\rangle_{f(\boldsymbol{x})} \\
&= \left\langle \log \sum_b \phi_{b|a} \frac{\omega_b g_b(x)}{\phi_{b|a}} \right\rangle_{f(\boldsymbol{x})} \\
&\geq \left\langle \sum_b \phi_{b|a} \log \frac{\omega_b g_b(x)}{\phi_{b|a}} \right\rangle_{f(\boldsymbol{x})} \\
&\stackrel{\text{def}}{=} \mathcal{L}_f(g, \phi) \ . \qquad (13)
\end{aligned}
$$

The above is a lower bound on $L_f(g)$. We can get the best bound by maximizing $\mathcal{L}_f(g, \phi)$ with respect to ϕ. The maximum value is obtained with:

$$
\hat{\phi}_{b|a} = \frac{\omega_b e^{-D(f_a \| g_b)}}{\sum_b' \pi_b' e^{-D(f_a \| g_{b'})}} \ . \qquad (14)
$$

Likewise, we define ,

$$
\mathcal{L}_f(f, \psi) \stackrel{\text{def}}{=} \left\langle \sum_{a'} \psi_{a'|a} \log \frac{\pi_{a'} f_{a'}(x)}{\psi_{a'|a}} \right\rangle_{f(\boldsymbol{x})} \ . \qquad (15)
$$

The optimal $\psi_{a'|a}$ is given by,

$$
\hat{\psi}_{a'|a} = \frac{\pi_{a'} e^{-D(f_a \| f_{a'})}}{\sum_{\hat{a}} \pi_{\hat{a}} e^{-D(f_a \| f_{\hat{a}})}} \qquad (16)
$$

Now, like Eq. 12, we can define $D_{\text{variational}}(f \| g) = \mathcal{L}_f(f, \hat{\psi}) - \mathcal{L}_f(g, \hat{\phi})$. After substituting $\hat{\phi}_{b|a}$ and $\hat{\psi}_{a'|a}$, we finally get,

$$
D_{\text{variational}}(f \| g) = \sum_a \pi_a \log \frac{\sum_{a'} \pi_{a'} e^{-D(f_a \| f_{a'})}}{\sum_b \omega_b e^{-D(f_a \| g_b)}} \ .
$$

$D_{\text{variational}}(f \| g)$ satisfies the self similarity and self identification property. However, it may not always satisfy the positivity property. In that case we may use the absolute value of the divergence.

3 Results

We conducted extensive experiments using both simulated and real data sets, including three simulated data sets (Data Set 1, Data Set 2, Data Set 3) drawn from different Dirichlet distributions and a real gene expression data set (Modencodefly Data [7]). Data Set 3 and Modencodefly Data are highdimensional in

nature. The Modencodefly Data has been pre-processed and top 75 genes were chosen based on variation. Each experiment was conducted only once. The primary metrics for comparison were the Kullback-Leibler (KL) Divergence and the time taken for each approach.

Table 1. Comparison of KL Divergence and Time Taken for Different Data Sets

Data Set	(N, p, k)	Metric	Variational	Monte Carlo $n = 10000$	Monte Carlo $n = 100000$	Monte Carlo $n = 1000000$
Data Set 1	(4200, 3,3)	KL Div	0.0017	0.0024	0.0021	0.0017
		Time	**0.0004**	1.7	17.3391	181.3682
Data Set 2	(4200, 3,3)	KL Div	0.0022	0.0024	0.0024	0.0022
		Time	**0.0005**	1.7283	17.4883	180.9482
Data Set 3	(230, 75,6)	KL Div	6.2723	6.1737	6.1815	6.1824
		Time	**0.0014**	3.6702	37.4568	384.2183
Modencodefly	(147, 75,6)	KL Div	34.7652	33.5581	33.5471	33.5579
		Time	**0.0014**	3.6353	36.0671	375.7836

From Table 1, it is evident that our proposed variational approach consistently provided faster solutions for all data sets. Notably, the Monte Carlo method yielded KL Divergence values closer to the variational method as we increase the number of samples, which increases the execution time to a great extent. When $N \to \infty$ Monte Carlo technique offers accurate KL divergence, as the variance of the estimation error becomes zero. From our results we see that with increasing N, the KL divergence obtained using Monte Carlo technique approaches that of variational technique, implying that variational technique offers more accurate results than Monte Carlo technique for moderate N with much less execution time, indicating that our solution is more robust and applicable for practical applications.

4 Conclusion

This study addresses the efficient estimation of Kullback-Leibler (KL) Divergence in Dirichlet Mixture Models (DMM). Despite the analytical tractability of KL Divergence for Dirichlet distributions, extending it to DMMs has been challenging, leading past research to rely on computationally intensive Monte Carlo methods. In response, we propose a novel variational approach, providing a closed-form solution that significantly enhances computational efficiency. The method is validated using both simulated and real-world datasets, demonstrating superior efficiency and accuracy over traditional Monte Carlo-based methods. Notably, our approach exhibits robustness by providing accurate solutions which can be achieved only by a very large number of samples using Monte Carlo based methods. This transformative solution opens avenues for rapid exploration of diverse DMM models, advancing statistical analyses of RNA-seq data in gene co-expression analysis.

References

1. Pal, S., Heumann, C.: Clustering compositional data using Dirichlet mixture model. Plos One **17**(5), e0268,438 (2022)
2. Csiszár, I.: I-divergence geometry of probability distributions and minimization problems. Ann. Probabil. 146–158 (1975)
3. Ma, Z., Rana, P.K., Taghia, J., Flierl, M., Leijon, A.: Bayesian estimation of Dirichlet mixture model with variational inference. Pattern Recogn. **47**(9), 3143–3157 (2014)
4. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc.: Ser. B (Methodol.) **39**(1), 1–22 (1977)
5. Celeux, G., Govaert, G.: A classification EM algorithm for clustering and two stochastic versions. Comput. Stat. Data Anal. **14**(3), 315–332 (1992)
6. Hershey, J.R., Olsen, P.A.: Approximating the kullback leibler divergence between gaussian mixture models. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP 2007, vol. 4, pp. IV–317. IEEE (2007)
7. Graveley, B.R., et al.: The developmental transcriptome of drosophila melanogaster. Nature **471**(7339), 473–479 (2011)

Chapter 4

Revisiting Dirichlet Mixture Model: Unraveling Deeper Insights and Practical Applications

Summary

This article enhances the Dirichlet Mixture Model (DMM) by proposing a new parametrization based on mean and precision, improving the interpretability and flexibility of parameter estimation. The study explores four estimation scenarios, deriving maximum likelihood estimates (MLE) using the Expectation-Maximization (EM) algorithm. For high-dimensional data, it introduces an innovative technique using Stirling's and moment approximations for faster, closed-form solutions. The article also demonstrates the identifiability of the DMM and uses a Kullback-Leibler (KL) divergence approximation to evaluate model fit, showcasing its practical utility through simulated and real datasets.

Contributing Article

Pal, Samyajoy, and Christian Heumann. "Revisiting Dirichlet Mixture Model: Unraveling Deeper Insights and Practical Applications." *Statistical Papers* 66, no. 1 (2025): 1-38. https://doi.org/10.1007/s00362-024-01627-0.

Author Contributions

- Samyajoy Pal: Conceptualization, Methodology, Software, Writing – original draft
- Christian Heumann: Supervision, Writing – review & editing

Statistical Papers (2025) 66:2
https://doi.org/10.1007/s00362-024-01627-0

REGULAR ARTICLE

Revisiting Dirichlet Mixture Model: unraveling deeper insights and practical applications

Samyajoy Pal[1] · **Christian Heumann**[1]

Received: 24 March 2024 / Revised: 30 July 2024

Abstract

This study revisits the Dirichlet Mixture Model (DMM), offering comprehensive insights into specific facets of parameter estimation. Estimating parameters of the DMM is challenging, with previous approaches focusing on standard parametrization, which lacks interpretability. We propose an alternative parametrization of the Dirichlet distribution using mean and precision, which provides critical insights into the distribution's location and peakedness. This parametrization is versatile, covering a wide range of scenarios with varying locations and precision levels, making it applicable to diverse datasets. Depending on whether one or both parameters are unknown, the estimation procedure varies, and estimates also differ when precision is identical across mixture components. In this article, we introduce this alternative parametrization and meticulously explore four distinct scenarios, deriving maximum likelihood estimates (MLE) for each using the Expectation-Maximization (EM) algorithm. For high-dimensional data, where standard methods often falter due to additional challenges, we present an innovative estimation approach utilizing Stirling's approximation and moment approximation, which provides closed-form solutions and faster execution times. Our study demonstrates the identifiability of the DMM and employs a closed-form approximation for Kullback–Leibler (KL) divergence to evaluate goodness of fit. Practical applications are illustrated through the analysis of both simulated and real datasets, showcasing the practical utility of the DMM.

Keywords Dirichlet Mixture Model · EM algorithm · KL divergence · Identifiability · High-dimensional data

Mathematics Subject Classification 62F10 · 62H30

Samyajoy Pal
Samyajoy.Pal@stat.uni-muenchen.de

Christian Heumann
chris@stat.uni-muenchen.de

[1] Department of Statistics, LMU Munich, Ludwigstr. 33, 80539 Munich, Bavaria, Germany

1 Introduction

Unsupervised learning, a pivotal domain in machine learning, has proven indispensable across diverse applications such as gene coexpression analysis (Van Dam et al. 2018; Ficklin et al. 2017), chemical analysis (Ghezelbash et al. 2020; Zhu et al. 2021), image segmentation (Rosyadi and Suciati 2020; Deeparani and Sudhakar 2021), and business studies (van Leeuwen and Koole 2022). Clustering techniques, a fundamental aspect of unsupervised learning, play an important role in organizing and structuring data without explicit labels. Among these techniques, two overarching categories stand out: those relying on similarity or distance measures, such as k-means clustering (MacQueen et al. 1967), hierarchical clustering (Nielsen 2016) etc., and model-based approaches (Wang et al. 2020; Liu et al. 2023; Zhan and Young 2023) typified by the Gaussian Mixture Model (GMM) (Peel and MacLahlan 2000). Methods based on similarity measures aim to group data points based on their proximity in the feature space, making them particularly useful for applications where the notion of similarity is well-defined. On the other hand, model-based approaches, as exemplified by the GMM, seek to represent the underlying structure of the data through probabilistic models, offering a different perspective on clustering by capturing the inherent distributional characteristics of the data.

Within the realm of model-based clustering, the Dirichlet Mixture Model (DMM) has emerged as a powerful tool with applications ranging from image processing (Bouguila et al. 2004) and text analysis (Blei 2004) to speech processing (Ma et al. 2013) and data mining (Fan et al. 2012). It finds application in audio data analysis (Miotto and Lanckriet 2011) and context modeling (Rasiwasia and Vasconcelos 2012) as well. It is worth mentioning that the DMM is particularly well-suited for compositional data analysis (Greenacre 2021; Fačevicová et al. 2023). Compositional data, which consist of quantitative descriptions of the parts of a whole, inherently contain relative information and adhere to the Aitchison geometry (Aitchison 1982) on the simplex. Measurements such as probabilities, proportions, percentages, and parts per million (ppm) can all be considered as compositional data. Compositional data analysis presents several challenges. One primary issue is the 'spurious correlation' between ratios of variables, initially identified by Pearson (Pearson 1896) and later expanded upon by Chayes (Chayes 1960). They demonstrated that some correlations between components of the composition must be negative due to the unit sum constraint. Over the years, various transformations, such as log transformation and log-ratio transformation, have been proposed to address the unit sum constraint. However, the debate continues regarding the optimal transformation. Typically, clustering compositional data is performed using distance or similarity-based algorithms (e.g., KMeans, hierarchical clustering) following some transformation (Comas-Cufí et al. 2020). One significant advantage of the DMM is that it inherently works with the unit sum constraint without requiring additional transformations. Furthermore, the DMM has demonstrated superior performance in certain scenarios when compared to models like the KMeans, hierarchical clustering and Gaussian Mixture Model (GMM) (Pal and Heumann 2022).

Estimating the parameters of the Dirichlet Mixture Model (DMM) poses a formidable challenge due to its analytical intractability, stemming from integral expressions involving the gamma function and its derivatives. Previous attempts to tackle this challenge have taken different paths. Ma et al. (2014) implemented a Bayesian estimation strategy to determine the posterior distribution of the parameters of a DMM. By employing a gamma distribution as the prior for each Dirichlet parameter, they were able to approximate both the prior and posterior distributions as products of several mutually independent gamma distributions. Typically, in variational inference techniques, the factorized approximation (FA) method (Blei et al. 2017) is used for Bayesian estimation. However, for the DMM, the FA method does not yield an analytically tractable solution. Consequently, the extended factorized approximation (EFA) method (Jordan et al. 1999; Jaakkola 2001) is utilized to achieve an analytically tractable solution. Despite its utility, this approach, which we denote as VDMM, has a significant limitation: it assumes that the parameters in a Dirichlet distribution are mutually independent. This assumption overlooks the inherent correlations among the parameters. Furthermore, the approximations employed may result in discrepancies between the true posterior distribution and its approximation.

Rasiwasia and Vasconcelos (2012) utilized the Expectation-Maximization (EM) algorithm to estimate the parameters of the Dirichlet Mixture Model (DMM). The EM algorithm introduces a latent variable, representing the cluster labels, and during the E step, the cluster membership probabilities are computed. In the M step, the complete data log-likelihood function is decomposed into two parts: one containing the mixture proportion parameters and the other containing the Dirichlet parameters. These two parts are maximized separately. Given the lack of an analytically tractable solution for estimating the Dirichlet parameters, the authors employed the Newton-Raphson technique to numerically find the solution. We denote this method as NDMM.

Miotto and Lanckriet (2011) employed a generalized Expectation-Maximization (GEM) algorithm to estimate the parameters of the DMM. Instead of maximizing the log-likelihood at the M step, it can be shown that convergence is ensured by merely updating the parameter values in a way that increases the log-likelihood at each step. This approach, known as the generalized EM algorithm, is particularly advantageous when maximizing the log-likelihood at the M step is challenging. The E steps remain unchanged from the standard EM algorithm. The authors used an approximation similar to the Newton–Raphson method, but without the need to invert the Hessian matrix, thereby providing an efficient means to update the parameters at each M step. We denote this method as GDMM.

In our previous work, we contributed to the understanding of the DMM by presenting a variant using a Hard EM approach (Pal and Heumann 2022). In this approach, we introduced an additional classification step following the E step, where each data point is assigned to respective clusters based on their cluster membership probabilities. Subsequently, during the M step, the maximum likelihood (ML) estimates of the Dirichlet component parameters are obtained for each cluster separately, using only the data points available in that cluster, through standard ML estimation techniques. This Hard EM method is significantly faster than the usual EM algorithm and offers an innovative solution by utilizing readily available ML estimates of the distribution parameters. We denote this method as Hard DMM.

Building upon this foundation, the present study delves deeper into the complexities of the DMM. Since the standard parametrization of the Dirichlet distribution lacks interpretability, we introduce an alternative parametrization using mean and precision parameters. The mean parameter controls the location of the distribution, whereas the precision parameter determines the pickedness. When the precision is large, the dirichlet random variable is likely to be near the means and for a small precision, the dirichlet variable is distributed more diffusely. This parametrization offers a clearer interpretation of the model parameters, enabling a more comprehensive understanding of the distribution based on the parameter values. A nuanced interpretation of these parameters provides insights into situations where fixing one and optimizing the other may be advantageous. Notably, mean and precision exhibit a degree of decoupling in the maximum-likelihood objective, allowing for simplifications and speedups through alternate optimization. This alternative parametrization not only augments the model's interpretability, but also offers increased flexibility in fitting the DMM to the data, providing a broader spectrum of options for model fitting and optimizations. This parametrization is versatile, encompassing a broad spectrum of scenarios with varying locations and precision levels, making it applicable to diverse datasets. The estimates differ depending on whether one or both parameters are unknown, and the estimates also vary when precision is uniform across mixture components.

To facilitate parameter estimation, we employ the Expectation Maximization (EM) algorithm, deriving maximum likelihood estimates (MLE) under four distinct scenarios: mean unknown, precision unknown, both mean and precision unknown, and identical precision across clusters. On the other hand, high-dimensional data presents additional challenges where standard methods often fail, as the curse of dimensionality significantly increases the time required to fit mixture models. Recognizing these demands, we propose an innovative estimation approach utilizing Stirling's approximation and moment approximation, offering closed-form solutions and faster execution times. Our research brings novel perspectives to previously unexplored topics, particularly in addressing the identifiability of the DMM. In a noteworthy contribution, our study conclusively establishes the identifiability of the DMM, shedding light on a dimension that has received limited attention in prior research.

Additionally, our investigation extends to the use of Kullback–Leibler (KL) divergence for DMM, a crucial measure in information theory. Traditionally, obtaining KL divergence for DMM involved the resource-intensive Monte Carlo method, known for its time-consuming nature. In contrast, we take advantage of an innovative variational approach, providing a closed-form approximation of KL divergence. This method not only enhances computational efficiency but also presents a more accessible and practical alternative to the laborious Monte Carlo based techniques. By doing so, our study not only expands the discourse on DMM but also enables fast comparison between different DMM variants.

Practical applications are demonstrated through the analysis of both simulated and real datasets, showcasing the versatility and effectiveness of the proposed DMM in capturing intricate structures in complex data. The remaining sections of the paper are as follows. In Sect. 2 we discuss MLE of DMM with the new parameterization under 4 scenarios. Section 3 provides estimates for high dimensional data. Sections 4 and 5 deal with identifiability and KL divergence respectively. Section 6 displays the results

of simulated and real data experiments, Sect. 7 lays down the limitations and finally Sect. 8 concludes our contribution.

2 Methods

Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N$ denote a random sample of size N, where $\boldsymbol{X}_i$ is a p dimensional random vector with probability density function $f(\boldsymbol{x}_i)$ on $\mathbb{R}^p$. An observed random sample is denoted by $\boldsymbol{x} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_N^T)^T$, where $\boldsymbol{x}_i$ is the observed value of the random vector $\boldsymbol{X}_i$. Throughout this paper, vectors are represented in bold font.

The density of a mixture model with k mixture components for one observation $\boldsymbol{x}_i$ is given by the mixture density

$$p(\boldsymbol{x}_i) = \sum_{j=1}^{k} \pi_j f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j) \,, \tag{1}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$ contains the corresponding mixture proportions with $\sum_{j=1}^{k} \pi_j = 1$, $0 < \pi_j < 1$. The density component of mixture j is given by $f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j)$ and $\boldsymbol{\alpha}_j$, $j = 1, 2, ..., k$ is the vector of component specific parameters for each density. Then $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)$ denotes the vector of all parameters of the model.

The log-likelihood of the model for a sample of size N is then given by

$$\log p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \mid \boldsymbol{\alpha}, \boldsymbol{\pi}) = \sum_{i=1}^{N} \log \left[\sum_{j=1}^{k} \pi_j f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j) \right] . \tag{2}$$

The parameters of mixture model can be estimated using EM algorithm (Dempster et al. 1977). For the E step, we introduce latent categorical variables Z_i, assuming values $1, \ldots, k$ with probabilities $\pi_1, \ldots, \pi_k$ such that $Pr(\boldsymbol{X}_i \mid Z_i = j) = f(\boldsymbol{x}_i)$, $j = 1, \ldots, k$. The posterior probability that the data point i belongs to cluster j is computed using Bayes rule as

$$\gamma_{ij}(\boldsymbol{x}_i) = Pr(Z_i = j \mid \boldsymbol{X}_i = \boldsymbol{x}_i, \boldsymbol{\alpha}, \boldsymbol{\pi}) = \frac{\pi_j f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j)}{\sum_{r=1}^{k} \pi_r f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_r)} \,. \tag{3}$$

The expected complete data log likelihood for the current iteration number t can be decomposed as follows (Murphy 2022),

$$\begin{aligned} Q(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{t-1}) &= \mathbb{E}\left[\sum_{i=1}^{N} \log(p(\boldsymbol{x}_i, z_i \mid \boldsymbol{\alpha})) \mid \boldsymbol{x}, \boldsymbol{\alpha}^{t-1} \right] \\ &= \sum_{i=1}^{N} \sum_{j=1}^{k} \gamma_{ij} \log \pi_j + \sum_{i=1}^{N} \sum_{j=1}^{k} \gamma_{ij} \log f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j) \,. \end{aligned} \tag{4}$$

The two parts of Eq. 4 can be optimized separately at the M step to estimate the parameters of the model. We denote

$$Q(\boldsymbol{\pi}) = \sum_{i=1}^{N}\sum_{j=1}^{k}\gamma_{ij}\log\pi_j \text{ and } Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\sum_{j=1}^{k}\gamma_{ij}\log f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j)\ . \tag{5}$$

The Dirichlet density component j is given by

$$f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j) = \frac{\Gamma(\sum_{m=1}^{p}\alpha_{jm})}{\prod_{m=1}^{p}\Gamma(\alpha_{jm})}\prod_{m=1}^{p}x_{im}^{\alpha_{jm}-1}\ , \tag{6}$$

where $\sum_{m=1}^{p}x_{im} = 1, x_{im}\text{'s} > 0\ , \alpha_{jm}\text{'s} > 0$ and $\Gamma(\cdot)$ denotes a gamma function. Thus, for mixture component j, $j = 1,\ldots,k$, the parameter $\boldsymbol{\alpha}_j = (\alpha_{j1},\ldots,\alpha_{jp})$ is a p-dimensional vector. In our study, we employ an alternative parametrization of Dirichlet parameters using mean and precision. For component j, we denote the mean parameter as $\boldsymbol{M}_j$ and the precision parameter as S_j. Here, $\boldsymbol{M}_J = (M_{j1},\ldots,M_{jp})$ is a p dimensional vector.

Let us consider the following reparameterization of Dirichlet parameters.

$$S_j = \sum_{m=1}^{p}\alpha_{jm} \text{ and } M_{jm} = \mathbb{E}[X_{jm}] = \frac{\alpha_{jm}}{S_j}\ .$$

Hence, we denote $\alpha_{jm} = S_j M_{jm}$.

- **Case 1.** M_{jm} **known**, S_j **unknown:**
  Let the known value of M_{jm} be M^*_{jm}. We rewrite $Q(\boldsymbol{\alpha})$ from Eq. 5 as

$$\begin{aligned} Q(S_j) &= \sum_{i=1}^{N}\sum_{j=1}^{k}\gamma_{ij}\log f(\boldsymbol{x}_i \mid S_j) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{k}\gamma_{ij}\log\left[\frac{\Gamma(S_j)}{\prod_{m=1}^{p}\Gamma(S_j M^*_{jm})}\prod_{m=1}^{p}x_{im}^{S_j M^*_{jm}-1}\right]\ . \end{aligned} \tag{7}$$

  The first and second derivatives with respect to S_j can be obtained subsequently.

$$\begin{aligned} Q'(S_j) &= N_j\Psi(S_j) - N_j\sum_{m=1}^{p}M^*_{jm}\Psi(S_j M^*_{jm}) + \sum_{i=1}^{N}\sum_{m=1}^{p}M^*_{jm}\gamma_{ij}\log x_{im} \\ Q''(S_j) &= N_j\Psi'(S_j) - N_j\sum_{m=1}^{p}{M^*_{jm}}^2\Psi'(S_j M^*_{jm}) \end{aligned} \tag{8}$$

  Here $N_j = \sum_{i=1}^{N}\gamma_{ij}$, $\Psi(\cdot)$ is the di-gamma function and $\Psi'(\cdot)$ is the tri-gamma function. When $Q''(S_j) < 0$, the estimates of S_j can be obtained using Newton's

method (Isaacson and Keller 2012).

$$S_j^{new} = S_j^{old} - \frac{Q'(S_j)}{Q''(S_j)} \tag{9}$$

The Newton's method maximizes a local quadratic approximation to the objective function. When Newton's method fails, we can use a non-quadratic approximation (Minka 2000a). If $Q'(S_j) + S_j Q''(S_j) < 0$ the updates are given as

$$\frac{1}{S_j^{new}} = \frac{1}{S_j^{old}} + \frac{1}{S_j^{old\,2}} \frac{Q'(S_j)}{Q''(S_j)} \,. \tag{10}$$

- **Case 2.** M_{jm} **unknown,** S_j **known:**
  Let the known value of S_j be S_j^*. We can estimate M_{jm} in the following way.

$$Q(M_{jm}) \propto \sum_{i=1}^{N} \sum_{j=1}^{k} \gamma_{ij} \log \left[\prod_{m=1}^{p} \frac{\exp(S_j^* M_{jm} \log x_{im})}{\Gamma(S_j^* M_{jm})} \right] . \tag{11}$$

The first derivative of $Q(M_{jm})$ is given below.

$$\begin{aligned} Q'(M_{jm}) = & \left(\frac{1}{N_j} \sum_{i=1}^{N} \gamma_{ij} \log(x_{im}) \right) - \Psi(S_j^* M_{jm}) \\ & - \sum_{m=1}^{p} M_{jm} \left(\frac{1}{N_j} \sum_{i=1}^{N} \gamma_{ij} \log x_{im} - \Psi\left(S_j^* M_{jm}\right) \right) + \text{Constant} \,. \end{aligned} \tag{12}$$

There is no analytically tractable solution to obtain the estimate of M_{jm}. The estimate can be obtained using a fixed-point iteration. The iterative equation that needs to be solved can be written as,

$$\begin{aligned} \Psi(S_j^* M_{jm}^{new}) = & \frac{1}{N_j} \sum_{i=1}^{N} \gamma_{ij} \log(x_{im}) \\ & - \sum_{m=1}^{p} M_{jm}^{old} \left(\frac{1}{N_j} \sum_{i=1}^{N} \gamma_{ij} \log x_{im} - \Psi\left(S_j^* M_{jm}^{old}\right) \right) \,. \end{aligned} \tag{13}$$

Let us denote the obtained solution as, $\Psi(S_j^* \hat{M}_{jm})$, which can be further written as $\Psi(\hat{\alpha}_{jm})$. Subsequently, after inverting the di-gamma function the updates of M_{jm} can be obtained as

$$M_{jm}^{new} = \frac{\hat{\alpha}_{jm}}{\sum_{m=1}^{p} \hat{\alpha}_{jm}} \,. \tag{14}$$

- **Case 3.** M_{jm}, S_j **both unknown:**
  If M_{jm} and S_j are both unknown, we first estimate S_j considering M_{jm} as fixed and then estimate M_{jm} using the estimated value of S_j. We continue the process until convergence. For this method, we would need an initial guess of M_{jm}. Let that initial guess be M_{jm}^{init}. Then we can write,

$$\begin{aligned} Q(S_j) &= \sum_{i=1}^{N}\sum_{j=1}^{k} \gamma_{ij} \log f(\boldsymbol{x}_i \mid S_j) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{k} \gamma_{ij} \log \left[\frac{\Gamma(S_j)}{\prod_{m=1}^{p} \Gamma(S_j M_{jm}^{init})} \prod_{m=1}^{p} x_{im}^{S_j M_{jm}^{init}-1} \right]. \end{aligned} \tag{15}$$

  Now, S_j is estimated using Eqs. 9 and 10. Let the estimated value of S_j be $\hat{S}_j$. Thus,

$$Q(M_{jm}) \propto \sum_{i=1}^{N}\sum_{j=1}^{k} \gamma_{ij} \log \left[\prod_{m=1}^{p} \frac{\exp(\hat{S}_j M_{jm} \log x_{im})}{\Gamma(\hat{S}_j M_{jm})} \right]. \tag{16}$$

  M_{jm} is then estimated using Eq. 14.
- **Case 4.** S_j**'s are identical:**
  When S_j's are identical for all j, $j = 1, \ldots, k$, we can denote $S_j = S$. In this scenario, we estimate S only once and estimate M_{jm}'s for all j using that value.

$$\begin{aligned} Q(S) &= \sum_{i=1}^{N}\sum_{j=1}^{k} \gamma_{ij} \log f(\boldsymbol{x}_i \mid S) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{k} \gamma_{ij} \log \left[\frac{\Gamma(S)}{\prod_{m=1}^{p} \Gamma(S M_{jm}^{init})} \prod_{m=1}^{p} x_{im}^{S M_{jm}^{init}-1} \right]. \end{aligned} \tag{17}$$

  S is estimated using Eqs. 9 and 10. Using the estimated value of S, M_{jm}'s are estimated as in case 3.

For all the above scenarios, π_j is estimated as follows,

$$Q(\boldsymbol{\pi}) = \sum_{i=1}^{N}\sum_{j=1}^{k} \gamma_{ij} \log \pi_j \,. \tag{18}$$

Maximizing $Q(\boldsymbol{\pi})$, we get,

$$\hat{\pi_j} = \frac{N_j}{N}\,, \tag{19}$$

where $N_j = \sum_{i=1}^{N} \gamma_{ij}$.

3 Estimates for high-dimensional data

When we have high-dimensional data, estimating Dirichlet parameters becomes difficult. As p becomes very large, the time required to estimate the parameters also increases. Moreover, as there are no closed form solutions for the updates of the parameters at the M-step, it often leads to computational errors. In our study, we introduce two approximations, which provide closed form solution at the M-step such that it does not depend on p. When employing numerical methods such as the Newton–Raphson or non-quadratic approximations at the M step of the EM algorithm in high-dimensional settings, these methods may fail. For instance, the Hessian matrix might become non-invertible, or the specific conditions required for non-quadratic approximations might not be satisfied. Our proposed approximations bypass the iterative algorithm at each M step, significantly reducing execution time and avoiding computational errors. While our proposed parametrization offers distinct advantages, optimizing mean and precision separately incurs a notable increase in execution time. This becomes especially critical in high-dimensional settings, where efficient computational strategies are paramount for practical applications. As such, we opt for the standard parametrization, leveraging Stirling's approximation and moment approximation to provide estimates tailored for high-dimensional scenarios.

3.1 Stirling's approximation

When, p increases, we can say that $\sum_{m=1}^{p} \alpha_{jm}$ also increases as $\alpha_{jm} > 0$ for all j, $j = 1, \ldots, k$ and $m, m = 1, \ldots, p$. Thus, we can assume that as $p \to \infty$, $\sum_{m=1}^{p} \alpha_{jm} \to \infty$, where α_{jm}'s are not necessarily large in value. Let us first look at a result regarding Stirling's approximation of gamma function.

Result 1 *When* $x \to \infty$, $\Gamma(x+\alpha) = \Gamma(x)x^{\alpha}$, $\alpha \in \mathbb{C}$.

Proof Using, Stirling approximation Artin (2015):

$$\begin{aligned}\Gamma(x+\alpha) &\underset{x\to+\infty}{\sim} \sqrt{2\pi(x+\alpha)}\left(\tfrac{x+\alpha}{e}\right)^{x+\alpha}\\ &\underset{x\to+\infty}{\sim} \sqrt{2\pi x}\left(\tfrac{x+\alpha}{e}\right)^{x} x^{\alpha}e^{-\alpha}\end{aligned}$$

Moreover,

$$\begin{aligned}(x+\alpha)^{x}e^{-\alpha} &= x^{x}\exp\left(x\log\left(1+\frac{\alpha}{x}\right)-\alpha\right)\\ &= x^{x}e^{o(1)}\\ &\underset{x\to+\infty}{\sim} x^{x}\end{aligned}$$

Thus,

$$\Gamma(x+\alpha) \underset{x\to+\infty}{\sim} \sqrt{2\pi x}\left(\frac{x}{e}\right)^{x} x^{\alpha}$$

$$\underset{x\to+\infty}{\sim} \Gamma(x)x^{\alpha}$$

□

Using the above result, we provide estimates of $\boldsymbol{\alpha}$, suitable for high-dimensional data. From Eq. 4, we can write,

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\sum_{j=1}^{k} \gamma_{ij} \log f(\boldsymbol{x}_i \mid \boldsymbol{\alpha}_j) \,. \tag{20}$$

Maximizing $Q(\boldsymbol{\alpha})$ we get,

$$\Psi(\alpha_{jm}) = \Psi\left(\sum_{m=1}^{p}\alpha_{jm}\right) + \frac{1}{N_j}\sum_{i=1}^{N}\gamma_{ij}\log(x_{im}) \,, \tag{21}$$

Let $\sum_{r=1}^{p}(1-\delta_{r,m})\alpha_{jr} = \beta_{jm}$, where $\delta_{r,m}$ is the Kronecker delta, defined as:

$$\delta_{r,m} = \begin{cases} 1 & \text{if } r = m \\ 0 & \text{if } r \neq m \end{cases}$$

Then,

$$\begin{aligned}
\Psi\left(\sum_{m=1}^{p}\alpha_{jm}\right) &= \frac{\partial}{\partial\alpha_{jm}}\log\Gamma\left(\sum_{m=1}^{p}\alpha_{jm}\right) \\
&= \frac{\partial}{\partial\alpha_{jm}}\log\Gamma\left(\beta_{jm}+\alpha_{jm}\right)\big) \\
&\underset{\beta_{jm}\to+\infty}{\sim} \frac{\partial}{\partial\alpha_{jm}}\log\left(\Gamma(\beta_{jm})\beta_{jm}^{\alpha_{jm}}\right) \\
&= \frac{\partial}{\partial\alpha_{jm}}\left[\log\Gamma(\beta_{jm}) + \alpha_{jm}\log\beta_{jm}\right] \\
&= \log\beta_{jm} \\
&= \log\sum_{r=1}^{p}(1-\delta_{r,m})\alpha_{jr}
\end{aligned} \tag{22}$$

Now, Eq. 21 becomes,

$$\Psi(\alpha_{jm}) = \log\sum_{r=1}^{p}(1-\delta_{r,m})\alpha_{jr} + \frac{1}{N_j}\sum_{i=1}^{N}\gamma_{ij}\log(x_{im}) \tag{23}$$

$$\alpha_{jm} = \Psi^{-1}\left(\log\sum_{r=1}^{p}(1-\delta_{r,m})\alpha_{jr} + \frac{1}{N_j}\sum_{i=1}^{N}\gamma_{ij}\log(x_{im})\right) \tag{24}$$

3.2 Moment approximation

We can now use moment approximation, to estimate $\sum_{r=1}^{p}(1-\delta_{r,m})\alpha_{jr}$. We know that for cluster j,

$$E[X_m] = \frac{\alpha_{jm}}{\sum_{m=1}^{p}\alpha_{jm}}$$
$$E[X_m^2] = E[X_m]\frac{1+\alpha_{jm}}{1+\sum_{m=1}^{p}\alpha_{jm}}$$

It can be shown that (Minka 2000b) for cluster j,

$$\sum_{m=1}^{p}\alpha_{jm} = \frac{E[X_1]-E[X_1^2]}{E[X_1^2]-E^2[X_1]} \tag{25}$$

Let us denote, $S_j = \sum_{m=1}^{p}\alpha_{jm}$. Then, S_j can be written as,

$$S_j = \frac{\mu_{j1}-[\sigma_{j1}^2+\mu_{j1}^2]}{\sigma_{j1}^2}\,. \tag{26}$$

where, $\mu_{j1} = E[X_1]$ and $\sigma_{j1}^2 = E[X_1^2]-E^2[X_1]$ for cluster j. Now, μ_{j1} and σ_{j1}^2 can be estimated by,

$$\hat{\mu_{j1}} = \frac{1}{\#x_{i1}\in j}\sum_{x_{i1}\in j}x_{i1}$$
$$\hat{\sigma_{j1}^2} = \frac{1}{\#x_{i1}\in j}\sum_{x_{i1}\in j}(x_{i1}-\hat{\mu_{j1}})^2$$

Thus, the estimate of S_j is given by,

$$\hat{S_j} = \frac{\hat{\mu_{j1}}-[\hat{\sigma_{j1}^2}+\hat{\mu_{j1}^2}]}{\hat{\sigma_{j1}^2}} \tag{27}$$

Now, for iteration t, Eq. 24 can be rewritten as,

$$\alpha_{jm}^{t} = \Psi^{-1}\left(\log(\hat{S_j}-\alpha_{jm}^{t-1}) + \frac{1}{N_j}\sum_{i=1}^{N}\gamma_{ij}\log(x_{im})\right) \tag{28}$$

For visualization purposes, we employ T-SNE (Hinton and Roweis 2002; Hinton and van der Maaten 2008) plots to project the high-dimensional data into two dimensions. From a clustering standpoint, especially for high-dimensional data, using domain knowledge or data-driven dimensionality reduction techniques prior to clustering is a pragmatic approach. As the dimensionality increases, models such as the Gaussian Mixture Model (GMM) face an exponential increase in the number of parameters to estimate. In contrast, one advantage of the DMM is that the number of parameters to estimate increases linearly with the number of dimensions. Moreover, we develop estimates specifically tailored for high-dimensional scenarios to avoid analytical intractability and reduce execution time. In our analysis of both simulated and real high-dimensional datasets, we utilize DMM without any dimensionality reduction techniques.

4 Identifiability

The identifiability of a statistical model is crucial because it ensures that the model parameters can be uniquely determined from the observed data, preventing ambiguities in parameter estimation. A well-identified model is essential for reliable and interpretable statistical inferences, providing a solid foundation for drawing meaningful conclusions from empirical observations. Although DMM has been used by many researchers, identifiability of DMM has not been discussed before. In this section, we show that DMMs are identifiable.

Definition 1 A random variable X follows an exponential family (Andersen 1970) of distribution if its probability density function can be written in the following form,

$$p(x|\boldsymbol{\eta}) = h(x)exp\{\boldsymbol{\eta}^T T(x) - A(\boldsymbol{\eta})\} \tag{29}$$

where,

- $\boldsymbol{\eta}$ is a vector of parameters
- $T(x)$ is the sufficient statistics
- $A(\boldsymbol{\eta})$ is the cumulant function

Remark 1 Dirichlet distribution follows an exponential family of distributions.

Proof The Dirichlet density can be written as,

$$\begin{aligned} f(\boldsymbol{x}|\boldsymbol{\alpha}) &= \frac{\Gamma\left(\sum_p \alpha_p\right)}{\prod_p \Gamma(\alpha_p)} \prod_p x_p^{\alpha_p - 1} \\ &= exp\left\{ \sum_p (\alpha_p - 1)\log x_p - \left[\sum_p \log\Gamma(\alpha_p) - \log\Gamma\left(\sum_p \alpha_p\right)\right]\right\} \end{aligned} \tag{30}$$

Thus, Dirichlet distribution is an exponential family of distributions with

- $\boldsymbol{\eta} = \boldsymbol{\alpha} - 1$
- $A(\boldsymbol{\eta}) = \sum_p \log \Gamma(\alpha_p) - \log \Gamma(\sum_p \alpha_p)$
- $T(\boldsymbol{x}) = \log \boldsymbol{x}$

□

Definition 2 Let, $\mathcal{F} = \{F(\boldsymbol{x}, \boldsymbol{\alpha}); \boldsymbol{\alpha} \in \mathbb{R}_1^m, \boldsymbol{x} \in \mathbb{R}^p\}$ be a family of p dimensional cdf's indexed by a point $\boldsymbol{\alpha}$ in a Borel subset $\mathbb{R}_1^m$ of the Euclidean m space $\mathbb{R}^m$ such that $F(\boldsymbol{x}, \boldsymbol{\alpha})$ is measurable in $\mathbb{R}^p \times \mathbb{R}_1^m$. Then $\mathcal{H}$, the set of all finite mixtures of a class of distributions $\mathcal{F}$ is defined as the convex hull of $\mathcal{F}$:

$$\mathcal{H} = \Bigg\{ H(\boldsymbol{x}) : H(\boldsymbol{x}) = \sum_{j=1}^{k} \pi_j F(\boldsymbol{x}, \boldsymbol{\alpha}_j), \pi_j > 0, \sum_{j=1}^{k} \pi_j = 1, F(\boldsymbol{x}, \boldsymbol{\alpha}_j) \in \mathcal{F}, \quad k = 1, 2, \ldots \Bigg\}. \tag{31}$$

$\mathcal{F}$ generates identifiable finite mixtures if and only if $\mathcal{H}$ has the uniqueness of representation property.

$$\sum_{j=1}^{k} \pi_j F(\boldsymbol{x}, \boldsymbol{\alpha}_j) = \sum_{j=1}^{q} \pi'_j F'(\boldsymbol{x}, \boldsymbol{\alpha}_j) \tag{32}$$

implies, $k = q$ and for each j, $1 < j < k$ there is some l, $1 < l < k$ such that $\pi_j = \pi_l^{'}$ and $F(\boldsymbol{x}, \boldsymbol{\alpha}_j) = F'(\boldsymbol{x}, \boldsymbol{\alpha}_l)$.

Theorem 1 *The class $\mathcal{H}$, of all finite mixtures of the family $\mathcal{F}$ is identifiable if and only if $\mathcal{F}$ is a linearly independent set over the field of real numbers.*

Proof The proof is given by Yakowitz and Spragins (1968). □

Proposition 1 *If $\mathcal{F}$ is the family of Dirichlet distributions, then the class of all finite mixtures of $\mathcal{F}$ is identifiable.*

Proof From Eq. 30 we can write,

$$f(\boldsymbol{x}, \boldsymbol{a}) = C(\boldsymbol{\alpha}) e^{\boldsymbol{a}^T \log \boldsymbol{x}}$$

where, $C(\boldsymbol{\alpha}) = e^{-\left[\sum_p \log\ \Gamma(\alpha_p) - \log\ \Gamma(\sum_p \alpha_p)\right]}$ and $\boldsymbol{a} = \boldsymbol{\alpha} - \mathbf{1}$. Let us assume a linear relation in $\mathcal{F}$,

$$\sum_{j=1}^{k} \pi_j f(\boldsymbol{x}, \boldsymbol{a}_j) = 0 \ , \boldsymbol{a}_j \in \mathbb{R}^p \ . \tag{33}$$

Now, $\boldsymbol{a}^T \log \boldsymbol{x}$ can be regarded as a linear functional on $\log \boldsymbol{x}$. Moreover, we can say that,

$$[\boldsymbol{a}_1 - \boldsymbol{a}_l]^T \log \boldsymbol{x} = \sum_{m=1}^{p} (a_{1m} - a_{lm}) \log x_m \tag{34}$$

is a non-zero linear functional if $l \neq 1$. As, the kernel of a non-zero linear functional is a hyper-plane, there is some point $\boldsymbol{u} \in \mathbb{R}^p, u_l > 0, l = 1, \ldots, p$ such that $0 < \boldsymbol{a}_j^T \boldsymbol{u} \equiv \xi_j$ and $\xi_j \neq \xi_1 > 0$, $j = 2, \ldots, k$. Thus for all vectors $b\boldsymbol{u}, b > 0$ Eq. 33 can be written as,

$$\sum_{j=1}^{k} \pi_j C(\boldsymbol{a}_j) e^{b\xi_j} = 0 \,, \tag{35}$$

where, $\xi_j \neq \xi_1$ if $j \neq 1$. Now,as $\xi_j \neq \xi_1 > 0$, $j > 1$, the relationship in Eq. 35 does not hold if $\pi_1 \neq 0$. Continuing like this, we can show that Eq. 33 has only trivial solution $\pi_1 = \cdots = \pi_k = 0$. Thus, $\mathcal{F}$ is a linearly independent set over the field of real numbers. Hence, $\mathcal{F}$ is identifiable. □

5 Kullback–Leibler divergence

The KL divergence, also known as relative entropy, serves as a widely adopted statistical metric to assess the similarity between two probability density functions. Let us assume that we have two pdfs $f(x)$ and $g(x)$, then its formulation is expressed as:

$$D(f \| g) \overset{\text{def}}{=} \int f(x) \log \frac{f(x)}{g(x)} \mathrm{dx}. \tag{36}$$

This divergence, commonly employed in statistics, adheres to three key properties known as divergence properties:

- Self similarity: $D(f \| f) = 0$
- Self identification: $D(f \| g) = 0$ only if $f = g$
- Positivity: $D(f \| g) \geq 0$ for all f, g.

For Dirichlet distributions, KL divergence can be derived in closed form (Rezek and Roberts 2005). However, for DMM there is no closed form solution available. In our previous research (Pal and Heumann 2024), we introduced an innovative closed-form solution for computing KL divergence through a variational approach. Through an extensive array of simulations and real-world data analyses, our findings consistently affirm the reliability and robustness of the variational approach in accurately estimating KL divergence. Notably, our method boasts significantly reduced execution times compared to conventional Monte Carlo techniques, offering a compelling advantage in computational efficiency. Intriguingly, our investigations reveal an interesting trend: as sample sizes increase, the Monte Carlo method gradually converges to KL

divergence values akin to those obtained through the variational approach. However, this convergence comes at the expense of exponentially escalating execution times. Consequently, our findings unequivocally advocate for the adoption of the variational approximation over Monte Carlo-based methods, providing researchers with a swift and reliable means for model comparison and goodness-of-fit evaluation.

Theorem 2 *Let $\boldsymbol{X}$ be an $p \times 1$ random vector. Assume two Dirichlet distributions u and v specifying the probability distribution of $\boldsymbol{X}$ as,*

$$\begin{aligned} u: \ \boldsymbol{X} &\sim \text{Dir}(\alpha_{11}, \ldots, \alpha_{1p}) \\ v: \ \boldsymbol{X} &\sim \text{Dir}(\alpha_{21}, \ldots, \alpha_{2p}). \end{aligned}$$

Then, the KL divergence of u from v is given by,

$$\begin{aligned} \text{D}(u \,||\, v) = \log \frac{\Gamma\left(\sum_{i=1}^{p} \alpha_{1i}\right)}{\Gamma\left(\sum_{i=1}^{p} \alpha_{2i}\right)} + \sum_{i=1}^{p} \log \frac{\Gamma(\alpha_{2i})}{\Gamma(\alpha_{1i})} \\ + \sum_{i=1}^{p} (\alpha_{1i} - \alpha_{2i}) \left[\psi(\alpha_{1i}) - \psi\left(\sum_{i=1}^{p} \alpha_{1i} \right) \right]. \end{aligned} \tag{37}$$

Proposition 2 *Let f_a and g_b be two DMMs such that,*

$$\begin{aligned} f_a = f(x) = \sum_a \pi_a Dir(\boldsymbol{x}; \boldsymbol{\alpha}_a) \\ g_b = g(x) = \sum\nolimits_b \omega_b Dir(\boldsymbol{x}; \boldsymbol{\alpha}_b) \end{aligned}$$

Then using a variational approach, an approximated KL divergence can be expressed as,

$$D_{\text{variational}}(f \| g) = \sum_a \pi_a \log \frac{\sum_{a'} \pi_{a'} e^{-D(f_a \| f_{a'})}}{\sum_b \omega_b e^{-D(f_a \| g_b)}}. \tag{38}$$

For the proof of the aforementioned theorem and proposition, please refer to Pal and Heumann (2024). $D_{\text{variational}}(f \| g)$ satisfies the self similarity and self identification property. However, it may not always satisfy the positivity property. In that case we may use the absolute value of the divergence.

6 Results

In this section, we present a comprehensive exploration encompassing both simulation studies and real data analyses, providing a robust evaluation of our proposed methodology. For the simulation study, we meticulously examine four distinct scenarios utilizing the mean-precision parametrization, comparing true and estimated parameter values.

Additionally, we assess the goodness-of-fit through the computation of KL divergence (KL Div.) using Eq. 38.

To validate the efficacy of our proposed model, we conduct thorough comparisons with alternative DMM variants, including Hard DMM, VDMM, NDMM, and GDMM. We denote our proposed model as Soft DMM. The evaluation involves a diverse set of metrics, such as Accuracy (ACC) (Sokolova and Lapalme 2009), Adjusted Rand Index (ARI) (Hubert and Arabie 1985), Normalized Mutual Information (NMI) (Ana and Jain 2003; Kreer 1957), and Homogeneity Score (HMS) (Rosenberg and Hirschberg 2007), offering a comprehensive understanding of the model's performance across various criteria.

Furthermore, our exploration extends to high-dimensional data, where we conduct a dedicated simulation study. In the realm of real data analyses, we utilize a geochemical dataset and a wine chemical composition dataset, providing practical insights into the model's application in distinct domains. For high-dimensional real datasets, we delve into two RNA-sequence (RNA-seq) datasets, characterized by different types of cancer cells, further showcasing the versatility and relevance of our proposed methodology across diverse data landscapes. The entirety of the data analysis is conducted using the Python programming language (Van Rossum and Drake 2009).

6.1 Simulation study

Now we provide comprehensive exploration involving three experiments for each of the four scenarios, including a dedicated examination in a high-dimensional setting which consists of six experiments. A detailed exposition of the data generation techniques employed in these experiments can be found in the appendix section A, providing transparency to our methodology. In our comparisons with other DMM variants, we use Soft DMM considering both mean and precision to be unknown. While there are no existing high-dimensional variants of DMM (Bouguila and Ziou (2006) has provided high dimensional estimates for mixture models that use generalized Dirichlet density) for direct comparisons, we rigorously investigated our model's performance using diverse metrics, enhancing our understanding of its efficacy in various scenarios. The results of the experiments, as presented in the following tables, are conducted on different data sets, precluding direct comparisons across different scenarios. However, in Table 5, we provide a comparison of the performance of the Soft DMM under various scenarios using the same data sets. This allows for a direct comparison of performance across different scenarios, providing clearer insights into the efficacy of the Soft DMM under each condition.

The results presented in Tables 1 to 6 provide a detailed examination of various experiments within the simulation study. In Table 1, where precision is known and the mean is unknown, the comparison of estimated and true parameter values reveals a close alignment, accompanied by minimal KL divergence, indicating a robust fit. The notably high Adjusted Rand Index (ARI) further attests to the effectiveness of clustering. Similarly, Tables 2, 3, and 4 showcase comparable findings across experiments with different configurations of mean and precision parameters such as known mean, both mean and precision unknown and identical precision; consistently demonstrating

Table 1 True and estimated parameter values along with KL Div. (KL divergence) and ARI when S is known

Data set	Parameter	True value	Estimated value	KL Div	ARI
Experiment 1	$\boldsymbol{\pi}$	(0.3333, 0.3, 0.3667)	(0.3188, 0.288, 0.3932)	0.1768	0.8929
	$\boldsymbol{M}_1$	(0.6,0.3,0.1)	(0.6051, 0.2975, 0.0974)		
	$\boldsymbol{M}_2$	(0.1,0.7,0.2)	(0.0928, 0.71, 0.1972)		
	$\boldsymbol{M}_3$	(0.3,0.4,0.3)	(0.3017, 0.4005, 0.2978)		
Experiment 2	$\boldsymbol{\pi}$	(0.4762, 0.2857, 0.2381)	(0.4787,0.2861,0.2351)	0.0101	0.9719
	$\boldsymbol{M}_1$	(0.7,0.2,0.1)	(0.6984, 0.1996, 0.102)		
	$\boldsymbol{M}_2$	(0.1,0.7,0.3)	(0.0899, 0.6366, 0.2735)		
	$\boldsymbol{M}_3$	(0.4,0.1,0.5)	(0.3923, 0.1005, 0.5072)		
Experiment 3	$\boldsymbol{\pi}$	(0.4444, 0.3333, 0.2222)	(0.4446,0.3336,0.2218)	0.001	0.9964
	$\boldsymbol{M}_1$	(0.8,0.1,0.1)	(0.7989, 0.1006, 0.1005)		
	$\boldsymbol{M}_2$	(0.1,0.7,0.2)	(0.0985, 0.7009, 0.2007)		
	$\boldsymbol{M}_3$	(0.3,0.2,0.5)	(0.2953, 0.1976, 0.5071)		

Table 2 True and estimated parameter values along with KL Div. (KL divergence) and ARI when M is known

Data set	Parameter	True value	Estimated value	KL Div	ARI
Experiment 1	$\boldsymbol{\pi}$	(0.4444, 0.3333, 0.2222)	(0.4446,0.3335,0.222)	0.0007	0.9962
	S_1	60	61.6694		
	S_2	80	82.22		
	S_3	20	21.0003		
Experiment 2	$\boldsymbol{\pi}$	(0.4762, 0.2857, 0.2381)	(0.476, 0.2858, 0.2381)	0.0006	0.9934
	S_1	60	60.9738		
	S_2	40	42.1422		
	S_3	100	102.7868		
Experiment 3	$\boldsymbol{\pi}$	(0.4444, 0.3333, 0.2222)	(0.4443, 0.3337, 0.222)	0.0006	0.9957
	S_1	50	51.1708		
	S_2	40	41.1029		
	S_3	200	210.3716		

Table 3 True and estimated parameter values along with KL Div. (KL divergence) and ARI when S and M both unknown

Data set	Parameter	True value	Estimated value	KL Div	ARI
Experiment 1	$\boldsymbol{\pi}$	(0.5556, 0.2222, 0.2222)	(0.5567, 0.2179, 0.2254)	0.0009	0.9425
	S_1	80	80.6552		
	S_2	90	95.5528		
	S_3	100	101.7908		
	$\boldsymbol{M}_1$	(0.1,0.1,0.8)	(0.1002, 0.1003, 0.7995)		
	$\boldsymbol{M}_2$	(0.2,0.2,0.6)	(0.1976, 0.2004, 0.602)		
	$\boldsymbol{M}_3$	(0.3,0.3,0.4)	(0.299, 0.2996, 0.4013)		
Experiment 2	$\boldsymbol{\pi}$	(0.4444, 0.3333, 0.2222)	(0.4442, 0.3332, 0.2227)	0.0009	0.9976
	S_1	50	51.2383		
	S_2	40	41.6925		
	S_3	150	154.9036		
	$\boldsymbol{M}_1$	(0.7,0.1,0.2)	(0.6998, 0.1001, 0.2002)		
	$\boldsymbol{M}_2$	(0.2,0.5,0.3)	(0.1993, 0.4998, 0.3009)		
	$\boldsymbol{M}_3$	(0.5,0.4,0.1)	(0.4994, 0.3997, 0.1009)		
Experiment 3	$\boldsymbol{\pi}$	(0.4762, 0.2857, 0.2381)	(0.4759, 0.286, 0.2381)	0.0014	0.9786
	S_1	60	61.2284		
	S_2	40	41.9488		
	S_3	10	10.4203		
	$\boldsymbol{M}_1$	(0.7,0.2,0.1)	(0.7007, 0.1991, 0.1002)		
	$\boldsymbol{M}_2$	(0.1,0.8,0.1)	(0.0989, 0.7992, 0.102)		
	$\boldsymbol{M}_3$	(0.4,0.1,0.5)	(0.3929, 0.1014, 0.5057)		

Table 4 True and estimated parameter values along with KL Div. (KL divergence) and ARI when M unknown and S identical

Data set	Parameter	True value	Estimated value	KL Div	ARI
Experiment 1	$\boldsymbol{\pi}$	(0.4444, 0.3333, 0.2222)	(0.4447, 0.3333, 0.222)	0.0007	0.9671
	S	50	51.072		
	$\boldsymbol{M}_1$	(0.7,0.1,0.2)	(0.6997, 0.1002, 0.2001)		
	$\boldsymbol{M}_2$	(0.3,0.4,0.3)	(0.3008, 0.3987, 0.3006)		
	$\boldsymbol{M}_3$	(0.5,0.4,0.1)	(0.4966, 0.4033, 0.1)		
Experiment 2	$\boldsymbol{\pi}$	(0.5556, 0.2222, 0.2222)	(0.5552,0.2192,0.2256)	0.0008	0.9205
	S	80	82.8715		
	$\boldsymbol{M}_1$	(0.1,0.1,0.8)	(0.1, 0.1,0.8)		
	$\boldsymbol{M}_2$	(0.2,0.2,0.6)	(0.1993, 0.1977, 0.603)		
	$\boldsymbol{M}_3$	(0.3,0.3,0.4)	(0.2996, 0.2996, 0.4008)		
Experiment 3	$\boldsymbol{\pi}$	(0.4167, 0.3333, 0.25)	(0.4161,0.334,0.2499)	0.0008	0.9248
	S	35	34.1256		
	$\boldsymbol{M}_1$	(0.1,0.2,0.7)	(0.1003, 0.199, 0.7007)		
	$\boldsymbol{M}_2$	(0.3,0.3,0.4)	(0.302, 0.3, 0.3979)		
	$\boldsymbol{M}_3$	(0.4,0.5,0.1)	(0.3984, 0.4997, 0.1019)		

Table 5 Comparison of soft DMM under different scenarios

Data set	Scenario	KL Div	ACC	ARI	NMI	HMS	Time
Experiment 1	M Unknown	0.0005	0.9767	0.9405	0.8861	0.8857	**11.4958**
	S Unknown	**0.0003**	0.9769	0.9412	0.8873	0.886	13.1778
	M, S Both Unknown	0.0009	**0.9773**	**0.9425**	**0.8889**	**0.888**	20.0301
	Identical S	0.0078	0.9762	0.9382	0.8856	0.8863	15.211
Experiment 2	M Unknown	**0.0003**	**0.9991**	**0.9976**	**0.9937**	**0.994**	**5.7769**
	S Unknown	0.0007	**0.9991**	**0.9976**	0.9934	0.9936	12.277
	M, S Both Unknown	0.0009	**0.9991**	**0.9976**	**0.9937**	**0.994**	16.0856
	Identical S	0.2036	0.9958	0.9891	0.9766	0.9776	13.5253
Experiment 3	M Unknown	0.0008	0.9926	0.9779	0.9606	0.9597	**4.4319**
	S Unknown	**0.0007**	**0.9931**	**0.9791**	**0.9631**	**0.9623**	11.405
	M, S Both Unknown	0.0014	0.9929	0.9786	0.9616	0.9608	15.9828
	Identical S	0.1965	0.9781	0.9367	0.9155	0.9088	13.1586

Table 6 Comparison of different models on simulated data sets

Data set	Model	KL Div	ACC	ARI	NMI	HMS	Time
Experiment 1	Soft DMM	**0.0013**	**0.9988**	**0.9963**	**0.9917**	0.9918	14.9006
	Hard DMM	0.0014	**0.9988**	**0.9963**	**0.9917**	0.9918	1.5703
	VDMM	9.0399	0.4762	0.2131	0.3355	0.2324	24.2679
	NDMM	9.1063	0.4762	0.0	0.0	0.0	**0.0615**
	GDMM	0.0508	0.9969	0.9905	0.9833	**1.0**	7.9452
Experiment 2	Soft DMM	**0.0026**	0.9905	0.9796	0.9526	0.9516	41.0311
	Hard DMM	**0.0026**	**0.9907**	**0.98**	**0.9535**	**0.9524**	4.6035
	VDMM	1.5742	0.8167	0.8224	0.7992	0.7693	40.9632
	NDMM	5.2682	0.4545	0.0	0.0	0.0	**0.0044**
	GDMM	19.6616	0.4545	0.0	0.0	0.0	7.5183
Experiment 3	Soft DMM	**0.0013**	**0.8991**	**0.8569**	0.824	**0.8171**	139.5918
	Hard DMM	0.0808	0.7991	0.8314	**0.8406**	0.7917	24.9419
	VDMM	0.0041	0.7558	0.8542	0.8264	0.8151	89.9295
	NDMM	4.604	0.3125	0.0	0.0	0.0	**0.0086**
	GDMM	26.8711	0.3126	0.0	0.0002	0.0	8.3075

close alignment, low KL divergence, and strong clustering performance shown by high ARI.

Table 5 presents the comparison of Soft DMM performance under different scenarios. As expected, a better fit is observed when either mean or precision is unknown rather than when both mean and precision are unknown, as indicated by the KL divergence. Clustering performance is generally better when either mean or precision is unknown (except in Experiment 1). When both mean and precision are unknown, Soft DMM needs to estimate a greater number of parameters, resulting in the longest execution time. However, when precision is assumed to be identical across clusters, the execution time is shorter compared to the scenario where both mean and precision are unknown, while still providing moderately good results. In this case, the KL divergence is relatively higher, indicating a less accurate fit to the data. Thus, we can conclude that when some information about mean or precision is available, it is preferable to use estimates tailored to that specific scenario. When no information about mean and precision is available, the estimates for the scenario where both mean and precision are unknown should be used. However, if precise estimation is not critical and execution time is a concern, the estimates assuming identical precision can be employed.

Table 6 extends the analysis to a comparison of different DMM variants across three simulation experiments. Notably, Soft DMM consistently exhibits the lowest KL divergence across all experiments, suggesting superior model fit. In the first experiment, both Soft and Hard DMM yield the highest values for Accuracy (ACC), ARI, and Normalized Mutual Information (NMI), while GDMM achieves the highest homogeneity score. Despite NDMM's comparatively poor performance in the chosen metrics, it stands out as the fastest in terms of execution time. For the second experiment, Hard DMM stands out with the highest ACC, ARI, NMI, and homogeneity score. NDMM

excels in terms of speed. In the third experiment, Soft DMM again shows the lowest KL divergence, highest ACC, ARI, and homogeneity score, while Hard DMM leads in NMI, and NDMM maintains its status as the fastest variant. These detailed comparisons provide a nuanced understanding of the performance characteristics of different DMM variants across varied simulation scenarios.

In our investigation, we conduct six high-dimensional simulation experiments, where random numbers were drawn from distinct Dirichlet distributions with 10,000 dimensions. The parameter values are randomly sampled from a uniform distribution. We compare the performance of the method described in Sect. 2 (without approximation) with the method described in Sect. 3 (with approximation). Table 7 summarizes the outcomes of these experiments, revealing perfect scores (1.0) for Accuracy (ACC), Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), and Homogeneity Score across first four experiments. This exceptional performance can be attributed to the nature of high-dimensional space, where clusters being far apart often yield such results. For experiments 5, and 6, we deliberately drew random samples from Dirichlet distributions with very small parameter values to induce more challenging cluster structures. Despite these increased challenges, the metrics indicate commendable results.

As expected, we observe that the method without approximation provides a better goodness of fit for all the experiments compared to the method with approximations, as indicated by the Kullback-Leibler (KL) divergence. However, there is no significant pattern in the different clustering metrics between the two methods. In four out of six experiments, the clustering results are similar. In experiment 5, the results are better with the approximation, while in experiment 6, the results are better without the approximation.

Although the KL divergence with approximation appears relatively high, it is important to note that in high-dimensional spaces, small deviations from the true distribution can accumulate across numerous dimensions. The outstanding scores in ARI, ACC, NMI, and Homogeneity Score validate the effectiveness of the clustering results. The use of approximation for high-dimensional data may contribute to the higher KL divergence. Examining the execution times of the method with approximation, we observe duration of 144.8051 s, 162.1839s, and 164.3652 s, respectively. The next three experiments, with smaller data points, resulted in even lesser time taken, recording duration of 45.1354, 44.8841, and 67.7365 s. Given the substantial dimensionality of the datasets, these results are reasonable. Conversely, the method without approximation requires significantly more execution time. It takes several hours to fit the DMM to the high-dimensional datasets, posing a significant concern for practical applications. The challenges associated with high dimensions make it impractical to apply other Dirichlet Mixture Model (DMM) variants in such situations, even if we overlook potential computational errors. The fitting of the model could take several hours, highlighting the efficiency of our approach. The exceptional clustering results achieved on simulated datasets pave the way for a broad spectrum of applications in real-world scenarios.

Table 7 Performance of soft DMM on highdimensional simulated data sets with and without approximation (experiment is denoted as E)

Method	Data set	KL Div	ACC	ARI	NMI	HMS	Time
With approximation	E1	42.0101	1.0	1.0	1.0	1.0	144.8051
	E2	61.9818	1.0	1.0	1.0	1.0	162.1839
	E3	67.3562	1.0	1.0	1.0	1.0	164.3652
	E4	182.7654	1.0	1.0	1.0	1.0	45.1354
	E5	399.3344	0.8313	0.8411	0.8835	0.8225	44.8841
	E6	405.739	0.8064	0.7654	0.873	0.7746	67.7365
Without approximation	E1	14.445	1.0	1.0	1.0	1.0	7144.4689
	E2	14.3106	1.0	1.0	1.0	1.0	10882.3921
	E3	14.315	1.0	1.0	1.0	1.0	10984.2561
	E4	127.5433	1.0	1.0	1.0	1.0	4857.3305
	E5	255.2036	0.8438	0.8603	0.9026	0.8225	12286.8934
	E6	333.8578	0.7742	0.7315	0.8608	0.7557	13045.5093

6.2 Real data analysis

Dirichlet Mixture Model demonstrates optimal performance when applied to compositional data, characterized by expressions as parts or proportions of a whole. RNA-seq data generally comes with read counts (Anders et al. 2015; Liao et al. 2014) or pseudo counts (Liao et al. 2014; Li and Dewey 2011), which represent the number of reads that align to specific genes or genomic regions. When we use normalized expression profiles (Rau and Maugis-Rabusseau 2018; Godichon-Baggioni et al. 2019) for each feature which is nothing but the proportion of normalized counts observed for a given feature, it makes the data compositional. In our real data analysis section, we focus on two distinct real datasets, comparing various DMM variants. Furthermore, we delve into the complexities of two high-dimensional datasets, employing specialized estimates tailored for high-dimensional data. Brief descriptions of these datasets are provided below.

- **Geochemical Data:** Bachmann et al. (2019) studied major elements and PGE concentrations of LG and MG chromitites from the Bushveld Complex, renowned as the largest layered mafic-ultramafic intrusion globally. This complex hosts numerous chromitite layers that are laterally continuous and chemically similar. In geochemical data analysis, these layers are generally classified into lower (LG), middle (MG), and upper (UG) groups based on their stratigraphic position, which serve as the focal point. However, the studied data set only contains two stratigraphic layers LG and MG. The dataset comprises 13 chemical components, including 'Cr_2O_3', 'FeO', 'SiO_2', 'MgO', 'Al_2O_3', 'CaO', 'P', 'Au ICP', 'Pt ICP', 'Pd ICP', 'Rh ICP', 'Ir ICP', and 'Ru ICP'. The objective is to cluster these elements accurately within their respective stratigraphic layers. The dataset, encompassing two stratigraphic layers, is available for download on Kaggle. In this context, $N = 1123$, $p = 13$, and $k = 2$.
- **Wine Data:** Aeberhard et al. (1994) studied one data set among others specifically focusing on the results of a chemical analysis of wines. These wines originate from the same region in Italy but are derived from three different cultivars. The dataset includes the quantities of 13 constituents found in each type of wine, including attributes like Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenol, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, and Proline. This well-known dataset, widely used by researchers, is accessible from the UCI Machine Learning Repository (Aeberhard and Forina 1991). The primary goal of our analysis is to identify the different types of wines based on their components (Basu 2004), with a focus on clustering. Notably, the attributes Color intensity and Hue, not constituting chemical components, have been excluded from our compositional data analysis. The dataset involves 178 instances, with 11 variables (p) and 3 clusters (k).
- **Brain Cancer Data (Highdimensional):** Griesinger et al. (2013) studied this data set revealing an uncharted territory concerning the characteristics of immune cell infiltration in pediatric brain tumors. A profound comprehension of these traits serves as a fundamental step towards developing immunotherapy for pediatric

brain tumors. The study employed gene expression profiles to pinpoint differentially expressed immune marker genes across various brain tumor types. The overall design involved generating gene expression profiles from 130 surgical tumor and normal brain samples, utilizing Affymetrix HG-U133plus2 chips (Platform GPL570). Filtering gene expression profiles of different brain tumors and normal brain yielded key immune cell marker expressions. Comparative analyses between diverse brain tumors and normal brain samples were conducted to identify the differential immune characteristics of these tumors. The dataset encompasses five types of tumors, implying $k=5$. After eliminating zero values and features with low variance, 8776 genes were retained ($p=8776$). With 130 tumor samples we have $N=130$. The dataset can be accessed at Gene Expression Omnibus with accession number GSE50161.

- **Breast cancer data (high-dimensional):** Gruosso et al. (2016) studied expression data derived from various subtypes of Breast cancer. Within a cohort study focusing on primary invasive breast cancer, encompassing TN, HER2, Luminal A, and Luminal B subtypes, along with normal tissue samples and cell lines, tumor specimens were acquired during surgery before any patient treatment. Total RNA extraction was performed on all samples, and the entire transcriptome was quantified using Affymetrix U133 Plus 2.0 Chips. The dataset encompasses six types of cells, denoted by $k=6$. After the removal of zero values and features exhibiting low variance, the dataset retains 11,816 genes, corresponding to $p=11{,}816$. There are a total of 151 tumor samples, resulting in $N=151$. The dataset can be accessed at Gene Expression Omnibus with accession number GSE45827.

Given the additional challenges of using labeled real datasets (typically used for classification) for clustering purposes, we incorporate an additional metric, the F-score (Van Rijsbergen 1979), for our evaluation. The F-score, which is the harmonic mean of precision and recall, is a robust metric for evaluating classification performance. For multi-class classification problems, there are three popular methods for calculating the F-score: macro, micro, and weighted. In the macro approach, the F-score is calculated for each class independently and then averaged. This method assumes that all classes are equally important, which may not always be the case. In the micro approach, the contributions of all classes are aggregated to compute the average F-score, making it useful for datasets with class imbalance as smaller classes are given equal weight as larger classes. The weighted F-score calculates the F-score for each class independently and then averages them using a weight that depends on the number of true instances for each class. This approach accounts for class imbalance, thus offering a more realistic and representative metric. For our study, we have employed the weighted approach to ensure a balanced and accurate evaluation of our classification performance. To apply the F-score in the context of clustering, it is essential to address the label switching problem. This issue is efficiently managed using the Hungarian algorithm, also known as the Kuhn-Munkres algorithm (Kuhn 1955). Similarly, we have used Cohen's kappa (Cohen 1960) and Jaccard score (Jaccard 1912) to accurately assess classification performance. Cohen's kappa measures inter-rater agreement for categorical items, accounting for the possibility of agreement occurring by chance. The Jaccard score, also known as the Jaccard index, evaluates the similarity between

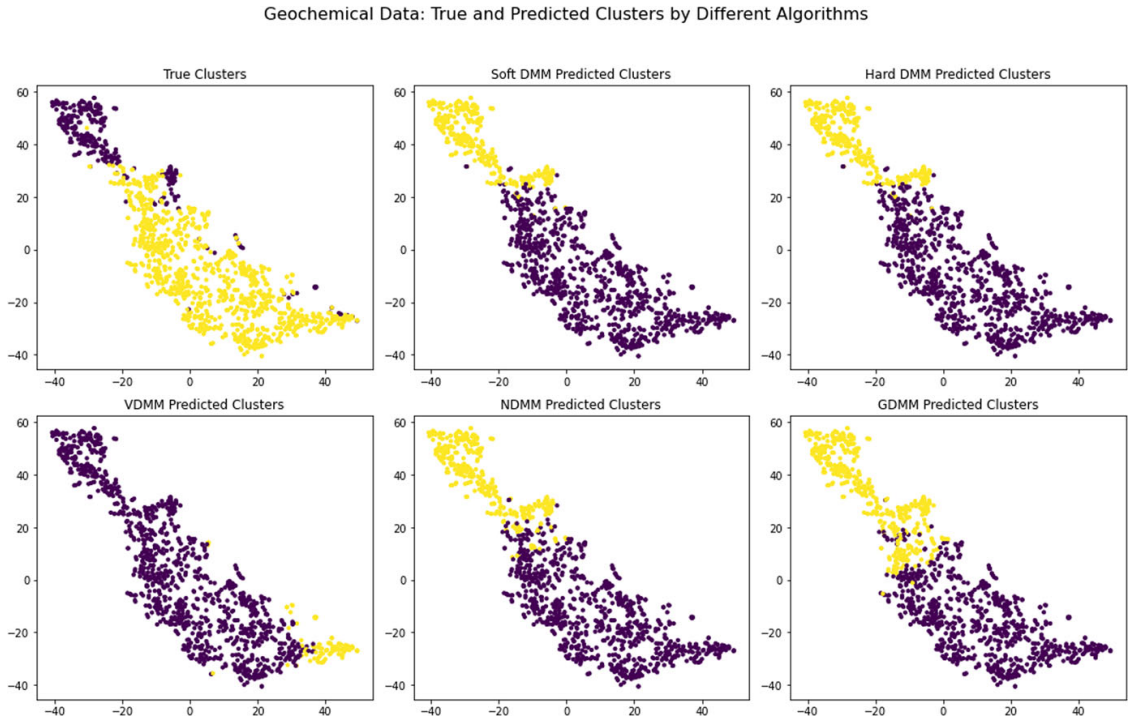

Fig. 1 True clusters and clusters predicted by different DMM algorithms on geochemical data

finite sample sets by comparing the intersection and union of the predicted and true classes.

Table 8 presents the comprehensive results for both Geochemical (D1) and Wine (D2) datasets. In the context of Geochemical data, the Hard DMM exhibits superior performance across key metrics such as ACC, ARI, NMI, and HMS compared to other models. However, Soft DMM demonstrates the best classification result as shown by the F-score, Cohen's kappa measure and Jaccard score. Notably, NDMM provides the fastest execution time. Figure 1 showcases a two-dimensional T-SNE plot depicting the true and predicted clusters by different models. Remarkably, the patterns identified by Soft DMM, Hard DMM, and NDMM closely resemble the true cluster pattern.

Shifting focus to the Wine dataset, results indicate that Hard DMM performs better than other models, in terms of ACC, ARI, and NMI, while GDMM excels in HMS. On the other hand, Soft DMM provides the best classification performance as revealed by the F-score, Cohen's kappa measure and Jaccard score. NDMM, once again, proves to be the fastest for this dataset. However, the observations from Fig. 2 reveal that Hard DMM, VDMM, and NDMM struggle to accurately capture the true cluster patterns. Despite ranking second in terms of clustering metrics, Soft DMM emerges as the model providing the most faithful representation of cluster patterns. The highest F-score, Cohen's kappa measure and Jaccard score of the Soft DMM further justify this observation. While Hard EM maximizes the classification log likelihood, potentially leading to improved clustering outcomes, Soft EM maximizes the mixture log-likelihood during model fitting and estimation. As a result, Soft DMM is recommended over Hard DMM for enhanced accuracy and robust parameter estimation.

Table 9 presents the outcomes for Brain Cancer and Breast Cancer datasets. Notably,

Table 8 Comparison of different models on real data sets

Data	Model	ACC	ARI	NMI	HMS	F-score	Cohen	Jaccard	Time
D1	Soft DMM	0.9181	0.6774	0.5163	0.5191	**0.9184**	**0.779**	**0.8526**	9.9939
	Hard DMM	**0.9216**	**0.6896**	**0.5292**	**0.5301**	0.9068	0.7503	0.8336	1.3788
	VDMM	0.2939	−0.0211	0.0024	0.0018	0.6458	−0.0354	0.5425	5.4193
	NDMM	0.8949	0.6035	0.4516	0.4674	0.8973	0.7286	0.8182	**0.1645**
	GDMM	0.8362	0.4386	0.3419	0.3709	0.8448	0.6134	0.7369	5.9379
D2	Soft DMM	0.6685	0.3109	0.4146	0.412	**0.6873**	**0.508**	**0.5387**	14.1661
	Hard DMM	**0.6742**	**0.4088**	**0.5029**	0.3924	0.5827	0.4755	0.4938	0.1249
	VDMM	0.3315	0.0	0.0	0.0	0.2275	0.0	0.1591	1.6103
	NDMM	0.5449	0.113	0.203	0.156	0.4644	0.2687	0.3166	**0.0027**
	GDMM	0.4944	0.3405	0.444	**0.4324**	0.4534	0.2695	0.3189	0.0784

Bold signifies the best result corresponding to the metric obtained by the model among all the models considered for the data

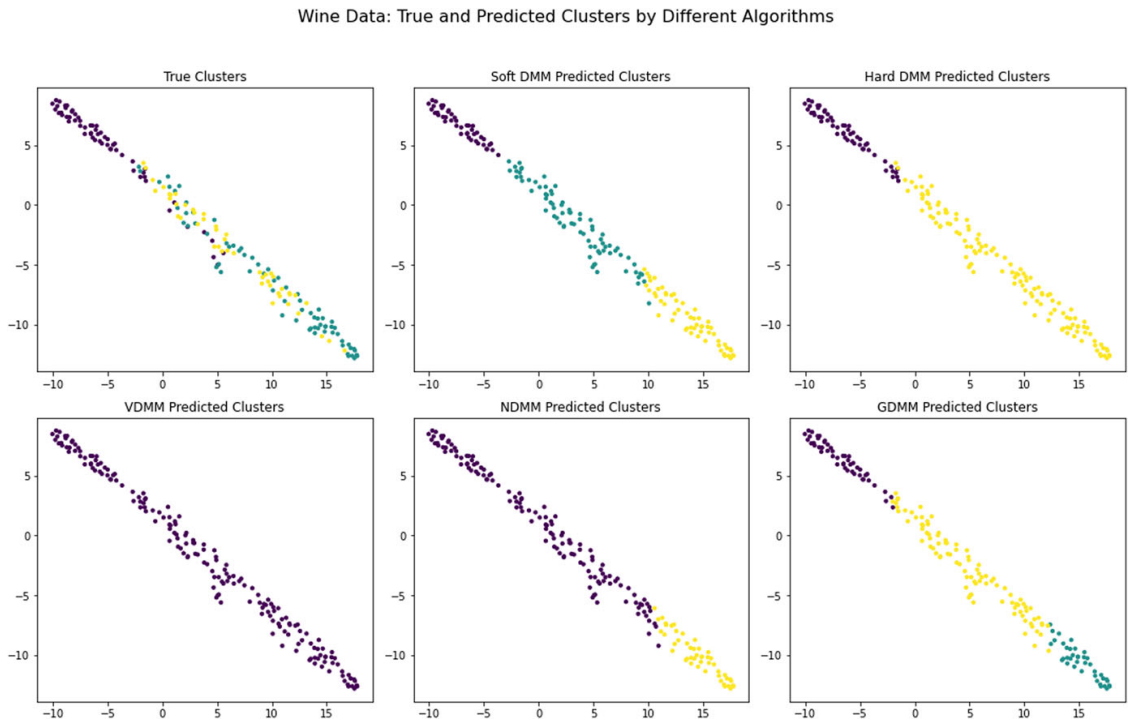

Fig. 2 True clusters and clusters predicted by different DMM algorithms on wine data

Table 9 Performance of DMM on high-dimensional real data sets

	Brain cancer data	Breast cancer data
ACC	0.7846	0.7152
F-Score	0.7378	0.6684
ARI	0.6127	0.6187
NMI	0.6810	0.7753
HMS	0.6162	0.7456
Cohen	0.7042	0.6464
Jaccard	0.6395	0.5852
Time	107.5818	167.2218

Soft DMM, leveraging high-dimensional estimates, demonstrated commendable performance across ACC, ARI, NMI, HMS, F-score, Cohen's kappa measure and Jaccard score. The execution time, despite the high dimensions of the data, remains quite reasonable. Additionally, Figs. 3 and 4 showcase 2D T-SNE plots for Brain Cancer and Breast Cancer data, depicting the alignment of predicted cluster patterns with the true ones. This visual representation reinforces the effectiveness of Soft DMM in capturing accurate cluster structures. It is worth highlighting that traditional DMM faces limitations in its application to high-dimensional RNA-seq data. However, our proposed approximation now enables the utilization of DMM in high-dimensional scenarios, unlocking its potential for broader applications.

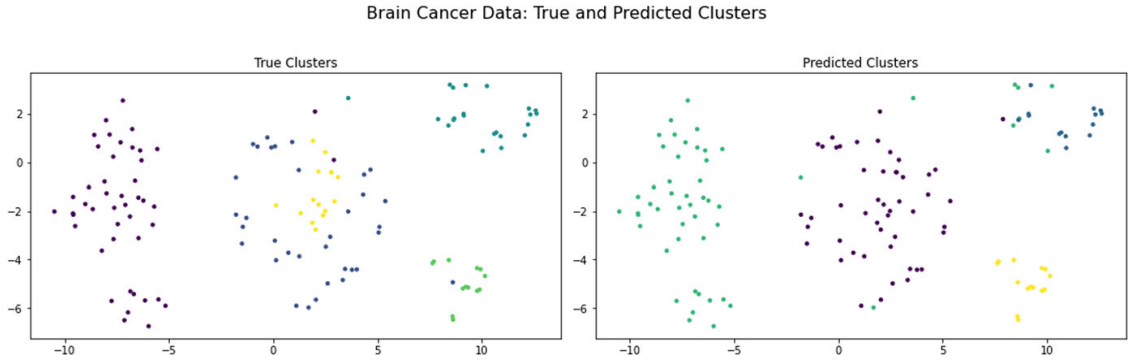

Fig. 3 True clusters and predicted clusters on brain cancer data

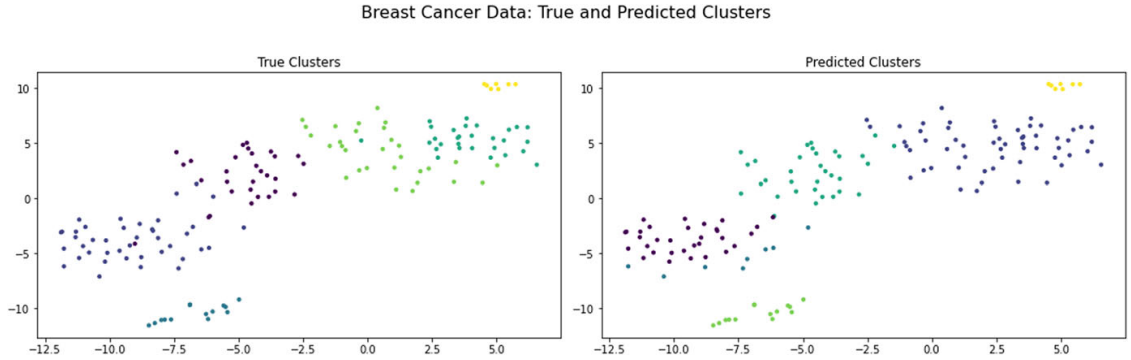

Fig. 4 True clusters and predicted clusters on breast cancer data

7 Limitations

While our study brings forth several strengths, including the provision of estimates for a diverse range of scenarios involving mean and precision parameters, capabilities for high-dimensional settings, and a closed-form approximation of KL divergence, it is imperative to recognize certain limitations. The adoption of Soft EM, though yielding accurate estimates, does come at the cost of increased computational time in comparison to alternatives such as Hard EM and Generalized EM. Researchers operating under constraints related to computational resources may find this aspect noteworthy.

A significant challenge arises in scenarios where both mean and precision parameters are unknown. The numerical optimization required at every M step contributes to prolonged execution times, particularly because closed form solutions are not readily available. Furthermore, while our mean-precision optimization theoretically extends to high-dimensional cases, practical implementation becomes intricate due to the considerable time it demands. The utilization of DMM for compositional data in high-dimensional scenarios introduces additional challenges, such as the presence of very small data values, potentially leading to computational errors. Our study also acknowledges the impact of zero values inside logarithmic functions, a common challenge in many clustering algorithms. Additionally, the use of approximation for KL divergence, although providing closed-form solutions, introduces potential limitations, including imprecise results at times and the risk of negative values that violate expected properties.

In our study, we utilized datasets with available labels commonly employed for testing supervised learning methods. These labels are often treated as a form of ground truth, providing a reference for model training and evaluation. However, it is crucial to highlight that labels used to train supervised models may not necessarily serve as a true ground truth when applied to clustering methods. The distinction arises from the nature of the learning objectives in supervised versus unsupervised scenarios.

Labels used for supervised learning are typically assigned with a specific class or category in mind, focusing on the discriminative aspects of the data. In contrast, clustering methods aim to identify inherent structures or patterns within the data itself, without the guidance of predefined categories. Consequently, while labeled datasets can be beneficial for certain aspects of model development, they may not accurately capture the intrinsic groupings or associations that clustering methods seek to unveil. The assumptions underlying supervised and unsupervised learning objectives differ, leading to a potential mismatch between labeled ground truth and the underlying structures that clustering algorithms attempt to reveal. This emphasizes the importance of cautious interpretation when applying labeled datasets traditionally used for supervised tasks to the evaluation of clustering methods. One advantage of using mixture models for clustering is that they provide a probability for each observation to belong to a cluster, enabling a nuanced interpretation of the results.

Despite these limitations, our work contributes valuable insights into the DMM, paving the way for further exploration and refinement across various applications.

8 Conclusion

In conclusion, our revisit of the DMM has contributed valuable insights, introducing an alternative parametrization that incorporates mean and precision parameters. The exploration of four distinct scenarios, coupled with the derivation of MLEs through the Expectation Maximization (EM) algorithm, has enhanced our understanding of the model's flexibility and applicability. Our study delves into specific facets, addressing challenges in estimating DMM parameters, particularly in high-dimensional settings. The introduction of an estimate tailored for high-dimensional scenarios, leveraging Stirling's approximation and moment approximation, signifies a crucial advancement. This adaptation extends the utility of DMM to a broader range of datasets, showcasing its adaptability to complex, real-world applications.

Our proposed model enjoys good convergence properties. Soft DMM uses an EM algorithm which is guaranteed to converge to a local optimum (Dempster et al. 1977). At the M step, it requires inversion of the digamma function. The digamma inversion algorithm is based on Newton's method (Goldstine 2012). Furthermore, the mean precision optimization technique uses Newton's method and non-quadratic approximation to obtain the estimates of S_j. Newton's method is said to have quadratic convergence (Hamming 2012). Although all of the algorithms mentioned above have good convergence properties, they suffer when a zero value occurs inside the logarithms during calculation. Moreover, if the initial value is far from the true value, the algorithm may fail to converge within some specified number of iterations. Thus, it is recommended to use k-means clustering for initializing the starting values of the parameters.

The identifiability of the DMM, elucidated in our study, underscores the model's robustness in capturing underlying patterns within diverse datasets. This finding contributes to the theoretical foundation of DMM, offering clarity on its parameter identifiability, a critical aspect in statistical modeling. Furthermore, our utilization of a closed-form approximation for KL divergence addresses a longstanding challenge in DMM analysis. This closed-form solution, in contrast to computationally intensive methods such as Monte Carlo, enhances the efficiency of DMM applications, facilitating faster and more accessible analyses.

The culmination of our study is exemplified in the comprehensive analysis of both simulated and real datasets. Through meticulous examination of simulated scenarios, we have demonstrated the model's performance across different conditions, showcasing the robustness of our proposed parametrization. The application of DMM to real datasets, spanning geochemical, wine, brain cancer, and breast cancer data, emphasizes the practical utility and versatility of the model. Our simulation studies have unraveled insights into DMM's behavior under various scenarios, shedding light on its strengths and potential limitations. The comparison with other DMM variants, including Hard DMM, VDMM, NDMM, and GDMM, has provided a comprehensive understanding of our proposed model's relative advantages. The promising results from the real data analyses further bolster the practical significance of our study. From the geochemical dataset to the diverse range of cancer datasets, including brain cancer and breast cancer data, our proposed DMM variant has consistently exhibited competitive performance, capturing underlying structures in the data effectively.

In essence, our study not only contributes to the theoretical advancement of the DMM but also establishes its practical efficacy in diverse applications. The maximum likelihood estimation using the alternative parametrization including four different scenarios, mathematical proof of identifiability, and adaptability to high-dimensional data sets collectively position our proposed DMM variant as a valuable tool for statisticians, data scientists, and researchers across various domains. The insights gained from this study pave the way for future refinements and extensions of the DMM, enhancing its applicability to an even broader spectrum of data analysis challenges.

Appendix A: Simulation study

Here we provide detailed description of the simulated data sets. Let us denote, n_j = number of data points in cluster j, $j = 1, 2, \ldots, k$. Here S and $\boldsymbol{M}$ are the precision and mean respectively. $\boldsymbol{\pi}$ is the mixture proportion. For highdimensional experiments we provide the data generation mechanisms rather than the true parameter values.

A.1: *S* known, *M* unknown

The corresponding results are shown in Table 1.

- Experiment 1: $S_1 = 10$, $S_2 = 50$, $S_3 = 100$.
  $\boldsymbol{M}_1 = (0.6, 0.3, 0.1)$, $\boldsymbol{M}_2 = (0.1, 0.7, 0.2)$, $\boldsymbol{M}_3 = (0.3, 0.4, 0.3)$.

$\boldsymbol{\pi} = (0.3333, 0.3, 0.3667)$.
$n_1 = 1000, n_2 = 900, n_3 = 1100$.

- Experiment 2: $S_1 = 60$, $S_2 = 50$, $S_3 = 10$.
  $\boldsymbol{M}_1 = (0.7, 0.2, 0.1)$, $\boldsymbol{M}_2 = (0.1, 0.7, 0.3)$, $\boldsymbol{M}_3 = (0.4, 0.1, 0.5)$.
  $\boldsymbol{\pi} = (0.4762, 0.2857, 0.2381)$.
  $n_1 = 2000, n_2 = 1200, n_3 = 1000$.
- Experiment 3: $S_1 = 60$, $S_2 = 60$, $S_3 = 20$.
  $\boldsymbol{M}_1 = (0.8, 0.1, 0.1)$, $\boldsymbol{M}_2 = (0.1, 0.7, 0.2)$, $\boldsymbol{M}_3 = (0.3, 0.2, 0.5)$.
  $\boldsymbol{\pi} = (0.4444, 0.3333, 0.2222)$.
  $n_1 = 2000, n_2 = 1500, n_3 = 1000$.

A.2: *S* unknown, *M* known

The corresponding results are shown in Table 2.

- Experiment 1: $S_1 = 60$, $S_2 = 80$, $S_3 = 20$.
  $\boldsymbol{M}_1 = (0.8, 0.1, 0.1)$, $\boldsymbol{M}_2 = (0.1, 0.7, 0.2)$, $\boldsymbol{M}_3 = (0.3, 0.2, 0.5)$.
  $\boldsymbol{\pi} = (0.4444, 0.3333, 0.2222)$.
  $n_1 = 2000, n_2 = 1500, n_3 = 1000$.
- Experiment 2: $S_1 = 60$, $S_2 = 40$, $S_3 = 100$.
  $\boldsymbol{M}_1 = (0.7, 0.2, 0.1)$, $\boldsymbol{M}_2 = (0.1, 0.8, 0.1)$, $\boldsymbol{M}_3 = (0.4, 0.4, 0.2)$.
  $\boldsymbol{\pi} = (0.4762, 0.2857, 0.2381)$.
  $n_1 = 2000, n_2 = 1200, n_3 = 1000$.
- Experiment 3: $S_1 = 50$, $S_2 = 40$, $S_3 = 200$.
  $\boldsymbol{M}_1 = (0.7, 0.1, 0.2)$, $\boldsymbol{M}_2 = (0.3, 0.4, 0.3)$, $\boldsymbol{M}_3 = (0.5, 0.4, 0.1)$.
  $\boldsymbol{\pi} = (0.4444, 0.3333, 0.2222)$.
  $n_1 = 2000, n_2 = 1500, n_3 = 1000$.

A.3: *S*, *M* both unknown

The corresponding results are shown in Table 3.

- Experiment 1: $S_1 = 80$, $S_2 = 90$, $S_3 = 100$.
  $\boldsymbol{M}_1 = (0.1, 0.1, 0.8)$, $\boldsymbol{M}_2 = (0.2, 0.2, 0.6)$, $\boldsymbol{M}_3 = (0.3, 0.3, 0.4)$.
  $\boldsymbol{\pi} = (0.5556, 0.2222, 0.2222)$.
  $n_1 = 2500, n_2 = 1000, n_3 = 1000$.
- Experiment 2: $S_1 = 50$, $S_2 = 40$, $S_3 = 150$.
  $\boldsymbol{M}_1 = (0.7, 0.1, 0.2)$, $\boldsymbol{M}_2 = (0.2, 0.5, 0.3)$, $\boldsymbol{M}_3 = (0.5, 0.4, 0.1)$.
  $\boldsymbol{\pi} = (0.4444, 0.3333, 0.2222)$.
  $n_1 = 2000, n_2 = 1500, n_3 = 1000$.
- Experiment 3: $S_1 = 60$, $S_2 = 40$, $S_3 = 10$.
  $\boldsymbol{M}_1 = (0.7, 0.2, 0.1)$, $\boldsymbol{M}_2 = (0.1, 0.8, 0.1)$, $\boldsymbol{M}_3 = (0.4, 0.1, 0.5)$.
  $\boldsymbol{\pi} = (0.4762, 0.2857, 0.2381)$.
  $n_1 = 2000, n_2 = 1200, n_3 = 1000$.

A.4: Identical S

The corresponding results are shown in Table 4.

- Experiment 1: $S = 50$.
  $\boldsymbol{M}_1 = (0.7, 0.1, 0.2)$, $\boldsymbol{M}_2 = (0.3, 0.4, 0.3)$, $\boldsymbol{M}_3 = (0.5, 0.4, 0.1)$.
  $\boldsymbol{\pi} = (0.4444, 0.3333, 0.2222)$.
  $n_1 = 2000, n_2 = 1500, n_3 = 1000$.
- Experiment 2: $S = 80$.
  $\boldsymbol{M}_1 = (0.1, 0.1, 0.8)$, $\boldsymbol{M}_2 = (0.2, 0.2, 0.6)$, $\boldsymbol{M}_3 = (0.3, 0.3, 0.4)$.
  $\boldsymbol{\pi} = (0.5556, 0.2222, 0.2222)$.
  $n_1 = 2500, n_2 = 1000, n_3 = 1000$.
- Experiment 3: $S = 35$.
  $\boldsymbol{M}_1 = (0.1, 0.2, 0.7)$, $\boldsymbol{M}_2 = (0.3, 0.3, 0.4)$, $\boldsymbol{M}_3 = (0.4, 0.5, 0.1)$.
  $\boldsymbol{\pi} = (0.4167, 0.3333, 0.25)$.
  $n_1 = 2500, n_2 = 2000, n_3 = 1500$.

A.5: Comparison of soft DMM under different scenarios

The corresponding results are shown in Table 5.

- Experiment 1: $S_1 = 80$, $S_2 = 90$, $S_3 = 100$.
  $\boldsymbol{M}_1 = (0.1, 0.1, 0.8)$, $\boldsymbol{M}_2 = (0.2, 0.2, 0.6)$, $\boldsymbol{M}_3 = (0.3, 0.3, 0.4)$.
  $\boldsymbol{\pi} = (0.5556, 0.2222, 0.2222)$.
  $n_1 = 2500, n_2 = 1000, n_3 = 1000$.
- Experiment 2: $S_1 = 50$, $S_2 = 40$, $S_3 = 150$.
  $\boldsymbol{M}_1 = (0.7, 0.1, 0.2)$, $\boldsymbol{M}_2 = (0.2, 0.5, 0.3)$, $\boldsymbol{M}_3 = (0.5, 0.4, 0.1)$.
  $\boldsymbol{\pi} = (0.4444, 0.3333, 0.2222)$.
  $n_1 = 2000, n_2 = 1500, n_3 = 1000$.
- Experiment 3: $S_1 = 60$, $S_2 = 40$, $S_3 = 10$.
  $\boldsymbol{M}_1 = (0.7, 0.2, 0.1)$, $\boldsymbol{M}_2 = (0.1, 0.8, 0.1)$, $\boldsymbol{M}_3 = (0.4, 0.1, 0.5)$.
  $\boldsymbol{\pi} = (0.4762, 0.2857, 0.2381)$.
  $n_1 = 2000, n_2 = 1200, n_3 = 1000$.

A.6: Comparison of different models

The corresponding results are shown in Table 6.

- Experiment 1: $k = 3$, $p = 3$.
  $\boldsymbol{\alpha}_1 = (42.0, 15.0, 3.0)$, $\boldsymbol{\alpha}_2 = (2.0, 36.0, 2.0)$, $\boldsymbol{\alpha}_3 = (4.0, 1.0, 5.0)$.
  $\boldsymbol{\pi} = (0.4762, 0.2857, 0.2381)$
  $n_1 = 2000, n_2 = 1200, n_3 = 1000$.
- Experiment 2: $k = 4$, $p = 4$.
  $\boldsymbol{\alpha}_1 = (30.0, 15.0, 0.3, 14.7)$, $\boldsymbol{\alpha}_2 = (0.2, 20.0, 2.0, 17.8)$, $\boldsymbol{\alpha}_3 = (0.6, 0.2850, 0.015, 2.1)$,$\boldsymbol{\alpha}_4 = (2.5, 2.5, 2.5, 2.5)$.
  $\boldsymbol{\pi} = (0.45454545, 0.27272727, 0.09090909, 0.18181818)$
  $n_1 = 2500, n_2 = 1500, n_3 = 500, n_4 = 1000$.

- Experiment 3: $k=6, p=4$.
  $\boldsymbol{\alpha}_1=(30.0,\ 15.0,\ 0.3,\ 14.7)$, $\boldsymbol{\alpha}_2=(0.2,\ 20.0,\ 2.0,\ 17.8)$, $\boldsymbol{\alpha}_3=(0.6,\ 0.285,\ 0.015,\ 2.1)$,$\boldsymbol{\alpha}_4=(2.5,\ 2.5,\ 2.5,\ 2.5)$, $\boldsymbol{\alpha}_5=(0.5,\ 0.5,\ 0.5,\ 3.5)$, $\boldsymbol{\alpha}_6=(7.5,\ 7.5,\ 9.0,\ 6.0)$.
  $\boldsymbol{\pi}=(0.3125,\ 0.1875,\ 0.0625,\ 0.125,\ 0.15,\ 0.1625)$
  $n_1=2500, n_2=1500, n_3=500, n_4=1000, n_5=1200, n_6=1300$.

A.7: High-dimensional

The corresponding results are shown in Table 7.

- Experiment 1: $p=10{,}000$.
  $\boldsymbol{\alpha}_1=$randomly drawn from uniform (10,20), $\boldsymbol{\alpha}_2=$randomly drawn from uniform (20,200), $\boldsymbol{\alpha}_3=$randomly drawn from uniform (10,100).
  $\boldsymbol{\pi}=(0.4762,\ 0.2857,\ 0.2381)$
  $n_1=500, n_2=300, n_3=250$.
- Experiment 2: $p=10{,}000$.
  $\boldsymbol{\alpha}_1=$randomly drawn from uniform (10,40), $\boldsymbol{\alpha}_2=$randomly drawn from uniform (10,40), $\boldsymbol{\alpha}_3=$randomly drawn from uniform (10,100).
  $\boldsymbol{\pi}=(0.4762,\ 0.2857,\ 0.2381)$
  $n_1=500, n_2=300, n_3=250$.
- Experiment 3: $p=10{,}000$.
  $\boldsymbol{\alpha}_1=$randomly drawn from uniform (10,40), $\boldsymbol{\alpha}_2=$randomly drawn from uniform (10,70), $\boldsymbol{\alpha}_3=$randomly drawn from uniform (50,100).
  $\boldsymbol{\pi}=(0.4762,\ 0.2857,\ 0.2381)$
  $n_1=500, n_2=300, n_3=250$.
- Experiment 4: $p=10{,}000$.
  $\boldsymbol{\alpha}_1=$randomly drawn from uniform (10,20), $\boldsymbol{\alpha}_2=$randomly drawn from uniform (19,32), $\boldsymbol{\alpha}_3=$randomly drawn from uniform (19,22), $\boldsymbol{\alpha}_4=$randomly drawn from uniform (10,22).
  $\boldsymbol{\pi}=(0.3125,\ 0.1875,\ 0.15625,\ 0.34375)$
  $n_1=50, n_2=30, n_3=25, n_4=55$.
- Experiment 5: $p=10{,}000$.
  $\boldsymbol{\alpha}_1=$randomly drawn from uniform (0.5,6), $\boldsymbol{\alpha}_2=$randomly drawn from uniform (19,32), $\boldsymbol{\alpha}_3=$randomly drawn from uniform (19,22), $\boldsymbol{\alpha}_4=$randomly drawn from uniform (0.5,10).
  $\boldsymbol{\pi}=(0.3125,\ 0.1875,\ 0.15625,\ 0.34375)$
  $n_1=50, n_2=30, n_3=25, n_4=55$.
- Experiment 6: $p=10{,}000$.
  $\boldsymbol{\alpha}_1=$randomly drawn from uniform (0.1,6), $\boldsymbol{\alpha}_2=$randomly drawn from uniform (10,15), $\boldsymbol{\alpha}_3=$randomly drawn from uniform (10,22), $\boldsymbol{\alpha}_4=$randomly drawn from uniform (0.5,5).
  $\boldsymbol{\pi}=(0.3226,\ 0.2581,\ 0.1935,\ 0.2258)$
  $n_1=50, n_2=40, n_3=30, n_4=35$.

Acknowledgements The authors would like to extend their heartfelt gratitude to the Editor-in-Chief, the Associate Editor, and three anonymous reviewers for their insightful comments and invaluable suggestions, which have greatly enhanced the quality of this study.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Aeberhard S, Forina M (1991) Wine. UCI Mach Learn Repository. https://doi.org/10.24432/C5PC7J
Aeberhard S, Coomans D, De Vel O (1994) Comparative analysis of statistical pattern recognition methods in high dimensional settings. Pattern Recogn 27(8):1065–1077
Aitchison J (1982) The statistical analysis of compositional data. J R Stat Soc Ser B (Methodol) 44(2):139–160
Ana LF, Jain AK (2003) Robust data clustering. In: 2003 IEEE Computer Society conference on computer vision and pattern recognition, 2003. Proceedings, vol 2. IEEE
Anders S, Pyl PT, Huber W (2015) Htseq—a python framework to work with high-throughput sequencing data. Bioinformatics 31(2):166–169
Andersen EB (1970) Sufficiency and exponential families for discrete sample spaces. J Am Stat Assoc 65(331):1248–1255
Artin E (2015) The gamma function. Courier Dover Publications, Mineola
Bachmann K, Menzel P, Tolosana-Delgado R, Schmidt C, Hill M, Gutzmer J (2019) Multivariate geochemical classification of chromitite layers in the bushveld complex, South Africa. Appl Geochem 103:106–117
Basu S (2004) Semi-supervised clustering with limited background knowledge. In: AAAI, pp 979–980
Blei DM (2004) Probabilistic models of text and images. University of California, Berkeley
Blei DM, Kucukelbir A, McAuliffe JD (2017) Variational inference: a review for statisticians. J Am Stat Assoc 112(518):859–877
Bouguila N, Ziou D (2006) A hybrid SEM algorithm for high-dimensional unsupervised learning using a finite generalized Dirichlet mixture. IEEE Trans Image Process 15(9):2657–2668
Bouguila N, Ziou D, Vaillancourt J (2004) Unsupervised learning of a finite mixture model based on the Dirichlet distribution and its application. IEEE Trans Image Process 13(11):1533–1543. https://doi.org/10.1109/TIP.2004.834664
Chayes F (1960) On correlation between variables of constant sum. J Geophys Res 65(12):4185–4193
Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Meas 20(1):37–46
Comas-Cufí M, Martín-Fernández JA, Mateu-Figueras G, Palarea-Albaladejo J (2020) Modelling count data using the log ratio-normal-multinomial distribution. Stat Oper Res Trans (SORT) 44(1):99–126
Deeparani K, Sudhakar P (2021) Efficient image segmentation and implementation of k-means clustering. Mater Today Proc 45:8076–8079
Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J Roy Stat Soc Ser B (Methodol) 39(1):1–22
Fačevicová K, Filzmoser P, Hron K (2023) Compositional cubes: a new concept for multi-factorial compositions. Stat Pap 64(3):955–985
Fan W, Bouguila N, Ziou D (2012) Variational learning for finite Dirichlet mixture models and applications. IEEE Trans Neural Netw Learn Syst 23(5):762–774. https://doi.org/10.1109/TNNLS.2012.2190298
Ficklin SP, Dunwoodie LJ, Poehlman WL, Watson C, Roche KE, Feltus FA (2017) Discovering condition-specific gene co-expression patterns using gaussian mixture models: a cancer case study. Sci Rep 7(1):8617
Ghezelbash R, Maghsoudi A, Carranza EJM (2020) Optimization of geochemical anomaly detection using a novel genetic k-means clustering (GKMC) algorithm. Comput Geosci 134:104335
Godichon-Baggioni A, Maugis-Rabusseau C, Rau A (2019) Clustering transformed compositional data using k-means, with applications in gene expression and bicycle sharing system data. J Appl Stat 46(1):47–65
Goldstine HH (2012) A history of numerical analysis from the 16th through the 19th century. Springer, New York
Greenacre M (2021) Compositional data analysis. Annu Rev Stat Appl 8:271–299
Griesinger AM, Birks DK, Donson AM, Amani V, Hoffman LM, Waziri A, Wang M, Handler MH, Foreman NK (2013) Characterization of distinct immunophenotypes across pediatric brain tumor types. J Immunol 191(9):4880–4888

Gruosso T, Mieulet V, Cardon M, Bourachot B, Kieffer Y, Devun F, Dubois T, Dutreix M, Vincent-Salomon A, Miller KM et al (2016) Chronic oxidative stress promotes h2 ax protein degradation and enhances chemosensitivity in breast cancer patients. EMBO Mol Med 8(5):527–549
Hamming R (2012) Numerical methods for scientists and engineers. Courier Corporation, New York
Hinton G, Maaten L (2008) Visualizing data using t-SNE. J Mach Learn Res 9(2605):2579–2605
Hinton GE, Roweis S (2002) Stochastic neighbor embedding. In: Advances in neural information processing systems, vol 15, pp 833–840
Hubert L, Arabie P (1985) Comparing partitions. J Classif 2:193–218
Isaacson E, Keller HB (2012) Analysis of numerical methods. Courier Corporation, New York
Jaakkola TS (2001) Tutorial on variational approximation methods. In: Advanced mean field methods: theory and practice. MIT. https://doi.org/10.7551/mitpress/1100.003.0014
Jaccard P (1912) The distribution of the flora in the alpine zone. 1. New Phytol 11(2):37–50
Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK (1999) An introduction to variational methods for graphical models. Mach Learn 37:183–233
Kreer J (1957) A question of terminology. IRE Trans Inf Theory 3(3):208–208
Kuhn HW (1955) The Hungarian method for the assignment problem. Naval Res Logistics Q 2(1–2):83–97
Leeuwen R, Koole G (2022) Data-driven market segmentation in hospitality using unsupervised machine learning. Mach Learn Appl 10:100414
Li B, Dewey CN (2011) RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. BMC Bioinform 12:1–16
Liao Y, Smyth GK, Shi W (2014) featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. Bioinformatics 30(7):923–930
Liu Y, Kuang R, Liu G (2023) Penalized likelihood inference for the finite mixture of Poisson distributions from capture-recapture data. Stat Pap. https://doi.org/10.1007/s00362-023-01503-3
Ma Z, Leijon A, Kleijn WB (2013) Vector quantization of LSF parameters with a mixture of Dirichlet distributions. IEEE Trans Audio Speech Lang Process 21(9):1777–1790. https://doi.org/10.1109/TASL.2013.2238732
Ma Z, Rana PK, Taghia J, Flierl M, Leijon A (2014) Bayesian estimation of Dirichlet mixture model with variational inference. Pattern Recogn 47(9):3143–3157
MacQueen J et al (1967) Some methods for classification and analysis of multivariate observations. Berkeley Symp Math Stat Prob 1967:281–297
Minka TP (2000a) Beyond Newton's method. Technical report, Microsoft Research
Minka T (2000b) Estimating a Dirichlet distribution. Technical report, MIT
Miotto R, Lanckriet G (2011) A generative context model for semantic music annotation and retrieval. IEEE Trans Audio Speech Lang Process 20(4):1096–1108
Murphy KP (2022) Probabilistic machine learning: an introduction. MIT, Cambridge
Nielsen F (2016) Hierarchical clustering, pp. 195–211. Springer, Cham. https://doi.org/10.1007/978-3-319-21903-5_8
Pal S, Heumann C (2022) Clustering compositional data using Dirichlet mixture model. PLoS ONE 17(5):0268438
Pal S, Heumann C (2024) Gene coexpression analysis with Dirichlet mixture model: accelerating model evaluation through closed-form KL divergence approximation using variational techniques. In: International workshop on statistical modelling. Springer, pp 134–141
Pearson K (1896) VII. mathematical contributions to the theory of evolution. III. Regression, heredity, and panmixia. Philos Trans R Soc Lond Ser A 187:253–318
Peel D, MacLahlan G (2000) Finite mixture models. Wiley, Hoboken
Rasiwasia N, Vasconcelos N (2012) Holistic context models for visual recognition. IEEE Trans Pattern Anal Mach Intell 34(5):902–917
Rau A, Maugis-Rabusseau C (2018) Transformation and model choice for RNA-Seq co-expression analysis. Brief Bioinform 19(3):425–436
Rezek I, Roberts S (2005) Ensemble hidden Markov models with extended observation densities for biosignal analysis. Springer, London
Rosenberg A, Hirschberg J (2007) V-measure: a conditional entropy-based external cluster evaluation measure. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), pp 410–420
Rosyadi AW, Suciati N (2020) Image segmentation using transition region and k-means clustering. IAENG Int J Comput Sci 47(1):47–55

Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Inf Process Manag 45(4):427–437
Van Dam S, Vosa U, Graaf A, Franke L, Magalhaes JP (2018) Gene co-expression analysis for functional classification and gene-disease predictions. Brief Bioinform 19(4):575–592
Van Rijsbergen CJ (1979) Information retrieval, 2nd edn. Butterworth-Heinemann, Newton
Van Rossum G, Drake FL (2009) Python 3 reference manual. CreateSpace, Scotts Valley
Wang W-L, Jamalizadeh A, Lin T-I (2020) Finite mixtures of multivariate scale-shape mixtures of skew-normal distributions. Stat Pap 61(6):2643–2670
Yakowitz SJ, Spragins JD (1968) On the identifiability of finite mixtures. Ann Math Stat 39(1):209–214
Zhan D, Young DS (2023) Finite mixtures of mean-parameterized Conway–Maxwell–Poisson models. Stat Pap. https://doi.org/10.1007/s00362-023-01452-x
Zhu S, Shih H-C, Cui X, Yu C-Y, Ringer SP (2021) Design of solute clustering during thermomechanical processing of AA6016 Al–Mg–Si alloy. Acta Mater 203:116455

Chapter 5

Gaussian mixture model with modified hard EM algorithm in clustering problems

Summary

This article revisits the use of the Hard Expectation-Maximization (EM) algorithm, also known as Viterbi Training, for cluster analysis, particularly in the context of Gaussian Mixture Models (GMMs). While Hard EM is less computationally intensive and easier to implement than standard EM, it is generally viewed as inferior due to issues like biased estimates and lack of consistency. The study addresses these concerns by proposing modifications to the Hard EM algorithm to mitigate the problem of convergence to local optima. The modified algorithm's performance is evaluated across various scenarios, including different numbers of clusters, dimensions, overlaps, and data imbalances, using five benchmark datasets. The results are compared to those of the standard EM algorithm to assess whether Hard EM performs as poorly as often assumed. Additionally, the study includes an analysis of two real-world biological datasets to demonstrate the practical utility of the proposed modifications.

Contributing Article

Pal, Samyajoy, and Christian Heumann. "Gaussian mixture model with modified hard EM algorithm in clustering problems." In *Statistical Modeling and Applications on Real-Time Problems*, pp. 153-179. CRC Press, 2024. https://doi.org/10.1201/9781003356653-7.

Author Contributions

- Samyajoy Pal: Conceptualization, Methodology, Software, Writing – original draft
- Christian Heumann: Supervision, Writing – review & editing

Chapter 7

Gaussian mixture model with modified hard EM algorithm in clustering problems

Samyajoy Pal and Christian Heumann

7.1 INTRODUCTION

In many modern-day statistical and machine learning problems the expectation maximization (EM) algorithm [1] is widely used. Applications of the EM algorithm in unsupervised learning often involve mixture models [2,3]. Use of mixture models can be seen in many fields such as image matching [4] and audio and video scene analysis [5]. But there exists an alternative approach for estimating parameters of mixture models, which is known as Hard EM. Hard EM or Viterbi Training (VT) was first introduced by [6] for speech recognition technique. Since then, many researchers have used Hard EM for speech recognition problems [7–11]. In other fields such as natural language processing (NLP) [12–18], bioinformatics [19–21] and image analysis [22], it is also extensively used.

Hard EM is an unsupervised learning technique, which can be seen as a coordinate ascent procedure that locally optimizes a function. In the case of mixture models, Hard EM is often described as Classification EM (CEM) as CEM maximizes the classification likelihood instead of the mixture likelihood [23,24]. Neal, Hinton [25] has called another version of Hard EM as Sparse EM where, in the E step, the algorithm, instead of finding the marginals like standard EM, finds the modes of the hidden variables. Hard EM is also linked with KMeans [26] clustering algorithm. KMeans can be seen as a special case of Hard EM for a mixture of Gaussians with a common covariance matrix of the form $\sigma^2 I$ and unknown σ [24,27].

In many situations, the EM algorithm becomes slow and computationally expensive. On the other hand, Hard EM provides an easy and computationally less intensive solution by an appropriate maximization step [28]. It is also known for being more robust and faster than standard EM [29]. Despite having all these desired qualities, Hard EM has some theoretical disadvantages for which it is assumed to be less accurate than standard EM [30]. Contrary to standard EM, Hard EM does not increase the likelihood of the parameters given the observed data x. Instead, it increases the joint likelihood of latent variables and parameters. And that is why it lacks

DOI: 10.1201/9781003356653-7

consistency [31] and, in fact, can produce biased estimates [32]. Even with the above drawbacks, Hard EM still enjoys a fair share of applications in practice [33]. However, when and under what circumstances, Hard EM should be preferred over standard EM remains an open problem even today [29], which calls for further investigation. Another issue with EM algorithm is that sometimes it converges to local optimum instead of a global one [34]. To overcome the issue, generally two techniques are used. One involves using repeated random initialization and another uses KMeans centroids and empirical standard Deviations as starting values [35]. However, repeated run of EMs with random initialization consumes more time and in many situations (e.g., imbalanced data sets) KMeans work poorly. As a result, using values obtained from poorly fitted models lead to poor performance of EM algorithm.

In our study, we have revisited the problem of Hard EM in the case of mixture models, more precisely for its applications in clustering. Despite having theoretical disadvantages over standard EM, we wanted to investigate if Hard EM really performs worse than standard EM for clustering problems in different situations. The main objectives of our study are to:

- provide a modification to Hard EM to stop fast convergence to local optimums with one or more clusters being empty.
- assess the performance of Hard EM in clustering for different situations (e.g., increasing number of clusters, increasing dimensions, increasing overlap of clusters, imbalance in data points, etc.)
- compare the performance of Hard EM with standard EM to investigate if it really works worse as assumed.

We have used Hard EM with some modifications to build a Gaussian mixture model (Hard GMM). The model has been used on five benchmark data sets for Gaussian mixtures [36] which are often preferred to test novel clustering methods. We have evaluated the performance of the model in different situations and compared it with the standard EM (Usual GMM) at each stage. We have also used two real data sets from biology to evaluate its performance.

7.2 METHODOLOGY

In this section, we would like to introduce the mixture model in general using Hard EM.

Let $X_1, X_2 \ \ldots\ , X_N$ denote a random sample of size N, where X_i is a p dimensional random vector with probability density function $f(x_i)$ on R^p. We can write $X = (X_1^T, \ldots, X_N^T)^T$, where the superscript T denotes vector transpose.

Note that, the entire sample is represented by X, i.e., X is a N – tuple of points in R^p or an $N \times p$-matrix. $X = (x_1^T, \ldots, x_N^T)^T$ denotes an observed random sample where x_i is the observed value of the random vector X_i.

The density of a mixture model with k components for one observation x_i is given by the mixture density

$$p(x_i) = \sum_{j=1}^{k} \pi_j f_j(x_i|\alpha_j) \tag{7.1}$$

where $\pi = (\pi_1, \ldots, \pi_k)$ contains the corresponding mixture proportions with $\sum_{i=1}^{k} \pi_i = 1$ and $0 \leq \pi_i \leq 1$. $f_j(x_i|\alpha_j)$ is the density component of mixture j and α_j, $j = 1,2, \ldots, k$, are vectors of component specific parameters for each density. Then $\alpha = (\alpha_1, \ldots, \alpha_k)$ denotes the vector of all parameters (except π) of the model. The log likelihood of the model for a sample of size N is then given by

$$\log p(x_1, \ldots, x_N|\alpha, \pi) = \sum_{i=1}^{N} \log\left[\sum_{j=1}^{k} \pi_j f_j(x_i|\alpha_j)\right]. \tag{7.2}$$

The parameters can be estimated using the EM algorithm with some modifications. For that purpose, let us introduce latent variables Z_i, which are categorical variables taking on values $1, \ldots, k$ with probabilities $\pi_1, \ldots, \pi_k$ such that $Pr(X_i|Z_i = j) = f_j(x_i)$, $j = 1, \ldots, k$.

Further, probabilities γ_{ij} are introduced (conditional on the observed data $X = x$ and the parameter α):

$$\gamma_{ij}(x_i) = Pr(Z_i = j|X = x, \alpha) = \frac{\pi_j f_j(x_i|\alpha_j)}{\sum_{j=1}^{k} \pi_j f_j(x_i|\alpha_j)}. \tag{7.3}$$

Equation 7.3 can be seen as the probability of cluster membership j for a data point x_i. Now, we must note that, Hard EM and standard EM optimize two different objective functions. In case of a Hard EM the following objective function is optimized.

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \max_{z_1, \ldots, z_N} P_\Theta(x_1, \ldots, x_N, z_1, \ldots, z_n) \tag{7.4}$$

where, Θ denotes all parameters (π,α). But in case of a standard EM, the objective function is

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \sum_{z_1, \ldots, z_N} P_\Theta(x_1, \ldots, x_N, z_1, \ldots, z_n) \tag{7.5}$$

Hard EM does not maximize the likelihood; instead it applies a delta function approximation to the posterior probabilities $Pr(Z_i = j|X = x,\alpha)$, where $Z_i, i = 1, \ldots, N$ are the latent variables representing class labels. X and α are the data and model parameters, respectively. The approximation changes the E step as follows,

$$Pr(Z_i = j|X = x, \alpha) \approx I(j = z_i^*) \tag{7.6}$$

where, z_i^* = argmax γ_{ij}. γ_{ij}'s are nothing but the responsibilities (probabilities) for j, each data point belonging to different clusters. After this step, for a standard hard EM, the ratio of empirical cluster members and total observations serve as the new estimates of π_j . However, for our modified hard EM we do not propose to use that estimate. Instead, we optimize the expected complete data log likelihood with respect to π_j as usual like a standard EM. Our proposed technique to obtain the estimates of α_j is explained below.

The expectation of complete data log likelihood is given by

$$Q(\alpha, \alpha^{t-1}) = E\left[\sum_{i=1}^{N} \log(p(x_i, z_i|\alpha))|x, \alpha^{t-1}\right], \tag{7.7}$$

where t is the current iteration number. It can also be shown that [37]

$$Q(\alpha, \alpha^{t-1}) = \sum_{i=1}^{N}\sum_{j=1}^{k} \gamma_{ij} \log \pi_j + \sum_{i=1}^{N}\sum_{j=1}^{k} \gamma_{ij} \log f_j(x_i|\alpha_j) \tag{7.8}$$

At the M step, we optimize Q with respect to π and α. π_j is estimated in the usual way by $\frac{N_j}{N}$, where, $N_j = \sum_{i=1}^{N} \gamma_{ij}$ and for estimating α, we look at the part in Q which depends on α, which is given by

$$l(\alpha) = \sum_{i=1}^{N}\sum_{j=1}^{k} \gamma_{ij} \log f_j(x_i|\alpha_j) \tag{7.9}$$

Now, we choose α_j such that α_j^t = argmax $l(\alpha_j)$, which is obtained by the process of α_j assigning data points to respective clusters, given by argmax γ_{ij}, and estimate α_j by j some estimation method based on the assigned observations to that cluster. It can be seen as a Bayesian concept (although not strictly Bayesian) for learning where equation 7.3 provides the cluster membership probability. In that equation, π_j can be viewed as the prior probability of $Z_i = j$ and the quantity γ_{ij} as the corresponding posterior probability once x is observed. Hence, it is computed using Bayes rule. The idea of choosing the cluster based on maximum probability is the same as choosing the MAP estimate, the mode of the distribution of $Pr(Z_i = j|X,\alpha)$. The MAP estimate is given by

$$Z_i^* = \underset{j}{\operatorname{argmax}}\, \gamma_{ij} = \underset{j}{\operatorname{argmax}}\ \log\ p(x_i|Z_i = j, \alpha) + \log\ p(Z_i = j|\alpha) \quad (7.10)$$

In addition to the general setting of a Hard EM, we include an extra step at the M step of the algorithm as a modification. Instead of obtaining the MLE right away at M step, we propose to do a quality check of the model. For cluster j, $j = 1, \ldots, k$ we denote

$$\alpha_j^{new} = \begin{cases} \alpha_j^{init}, & \textit{if cluster } j \textit{ is empty} \\ \alpha_j^{MLE}, & \textit{otherwise} \end{cases} \quad (7.11)$$

where α_j^{init} is the initial value of the parameter α_j. Hard EM is well known for its greedy convergence; as a result, often, the algorithm converges with one or more clusters being empty. Hence, we would like to force the algorithm to re-iterate if one or more clusters are found to be empty at each M step.

At first, some trial values of the distribution parameters α and mixture proportions π are initialized to start the algorithm. Then the initial value of the log likelihood is evaluated. For different distributions, different techniques can be used to choose suitable initial values. It is known that EM algorithm is very sensitive to the choice of initial values [38]. Hard EM is no different in this regard. However, with proper initialization techniques, Hard EM can provide a robust performance. In the literature, we find different techniques of choosing starting values such as random initialization [39], iteratively constrained EM [40], KMeans clustering [41], Sum scores [42], etc. However, for better performance KMeans initialization and iteratively constrained EM with random initialization are most preferred [43]. There exist some robust versions of EM algorithm (see [44]) which take into account the number of clusters as well. For our study, we have taken, the centroids of KMeans as initial values of μ, and the empirical covariance matrix of each cluster is taken as an initial value of Σ_j. The initial values of π is computed using the ratio of cluster members obtained by KMeans algorithm and total observations.

At the E step, the values of the probabilities γ_{ij} are evaluated using the current parameter values. For a usual EM algorithm (e.g., in a GMM), at the M step, a weighted mean and a weighted covariance matrix are calculated using the γ_{ij} values. But for other distributions, where the model parameters are not mean and (co)variance, this technique can not be used. So, for different distributions, different techniques need to be used. Hard EM provides an easy and convenient solution where at the M step, each data point is assigned to a cluster depending on the probability of that data point belonging to each cluster. That cluster is assigned for which the probability is maximum. Now, if one or more clusters are found empty, then the initial value of the parameter α_j for cluster j is used. And for the

non-empty clusters, point estimates of the parameters of each parent distribution are obtained using only the data points available in each cluster. For faster convergence and convenience, maximum likelihood estimates can usually be recommended. The mixture component probabilities π_j are estimated as mentioned above by $\frac{N_j}{N}$. The new set of estimated values of the parameters is then used as an update over the previous one. After this step, the log likelihood is evaluated again using the updated parameter values. The process is then continued until convergence. Hard EM enjoys good convergence properties, which have been explained in detail by [45]. It is to be noted that, although estimates obtained through modified Hard EM can be seen as an approximated MLE and it means that MLE estimates from standard should be better, it is not guaranteed that standard EM would give better accuracy for assignments of data points to correct clusters. As Hard EM finds the MAP estimate i.e., the mode of the distribution of $Pr(Z_i = j|X,\alpha)$ to optimize the classification likelihood, which version of EM gives better accuracy should be investigated for a case-to-case basis.

Algorithm 1: Modified Hard EM Algorithm for Mixture Models

Initialize the model parameters, α and π. Evaluate the initial value of the log likelihood from equation (7.2);
while *loglikelihood difference* $\geq \in$ **do**
  Evaluate γ_{ij} From equation (7.3), using the parameter values and data
  $\pi_j^{new} = \frac{N_j}{N}$, where, $N_j = \mathrm{P}_{i=1}^{N}\ \gamma_{ij}$;
  for *i in1 to N* **do**
    cluster $z_i = \underset{j}{\operatorname{argmax}}\, \gamma_{ij}$;
    Assign data point x_i to cluster Z_i;
  end
  for *j in1 to k* **do**
    if *Cluster j is empty* **then**
      Use initial values of α_j as an update;
    else
      $\alpha_j^{new} = \alpha_j^{MLE}$;
    end
  end
  Re-evaluate log likelihood using the new values of the parameters.
end

For our experiments, we have used 0.0001 as the value of ϵ in Algorithm 1.

In case of a Gaussian mixture model, normal distribution can be used as the base distribution $f_j(.)$ in the model shown in equation 7.1.

For a $p \times 1$ continuous random vector X, the density of p variate multivariate normal distribution is given by

$$f(x|\mu, \Sigma) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right], \tag{7.12}$$

where μ is a $p \times 1$ vector, Σ is a $p \times p$ symmetric, positive definite matrix and the support of X is $R^{p\cdot}$

The maximum likelihood estimates of μ and Σ are given by

$$\widehat{\mu_{MLE}} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{7.13}$$

$$\widehat{\Sigma_{MLE}} = \frac{1}{N}\sum_i \left(x_i - \widehat{\mu_{MLE}}\right)\left(x_i - \widehat{\mu_{MLE}}\right)^T \tag{7.14}$$

The mixture model can be built the usual way and the model parameters can be estimated using the MLEs of a Gaussian distribution at the M step.

7.3 COMPARISON ON BENCHMARK DATA SETS

We have done an extensive experiment to observe the performance of Hard GMM in different conditions. Many authors [46] have argued that it may not be the best idea to test the clustering model only on synthetic data as the model is supposed to perform on real data problems. That is why we have decided to check the performance of our proposed method on both synthetic and real data. In this section, we are going to evaluate the performance of Hard GMM on basic benchmark data sets as proposed by [36]. The data sets are chosen in such a way that the sets are challenging enough for most typical heuristics to fail but easy enough for good clustering algorithms to identify the correct clusters. These data sets have been previously used by many other authors as well. A brief description of the data sets is given below in Table 7.1.

Figure 7.1, Figure 7.2 and Figure 7.3 show plots of data in data set A. In data set A, we have three sets of data containing distinct, separate clusters with the number of clusters 20, 35, and 50, respectively. We have four sets of data containing 15 clusters each in data set S, but with increasing overlap among clusters. Overlap has been increased by increasing the standard deviation of data points in each cluster. It is done in such a way that a good algorithm should still be able to identify the clusters. Figure 7.5, Figure 7.6, Figure 7.7 and Figure 7.8 show plots of data in data set S. Dim data sets contain six sets of data, each with distinct, separate clusters but with

Table 7.1 Descriptions of five basic benchmark data sets

Data set	*Variation*	*Size*	*Clusters*	*Dimension*	*Source*
A (3 sets)	Number of Clusters	3000–7500	20,30,50	2	Kärkkäinen, Fränti [47]
S (4 sets)	Overlap	5000	15	2	Fränti, Virmajoki [48]
Dim (6 sets)	Dimensions	1024	16	32–1024	Franti et al., [49]
G2 (100 sets)	Dimensions and overlap	2048	2	2–1024	Fränti et al., [50]
Unbalance (1 set)	Balance	6500	8	2	Rezaei, Fränti [51]

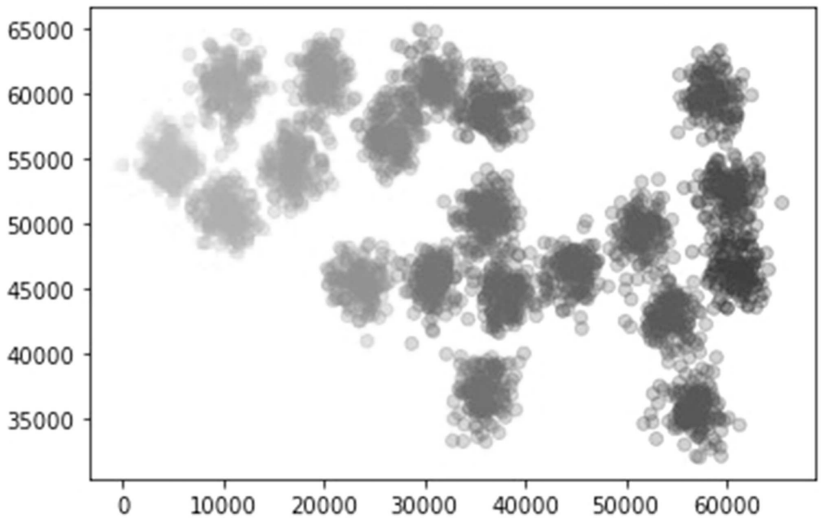

Figure 7.1 A1 data set.

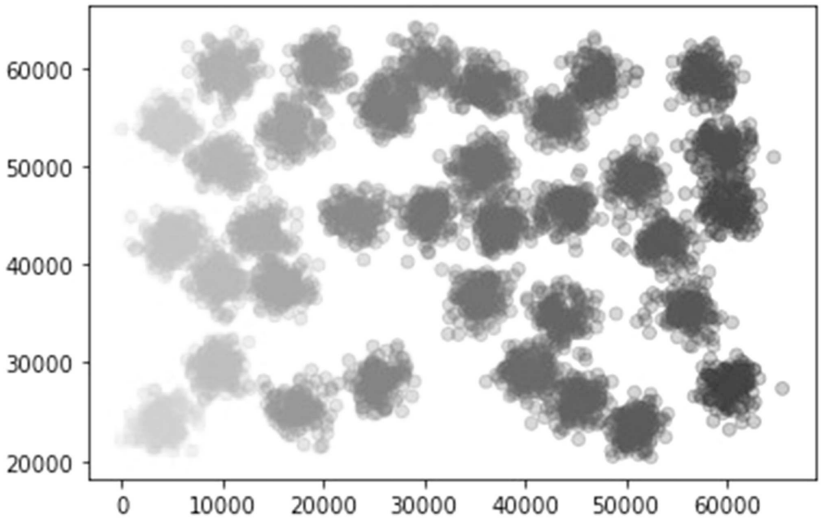

Figure 7.2 A2 data set.

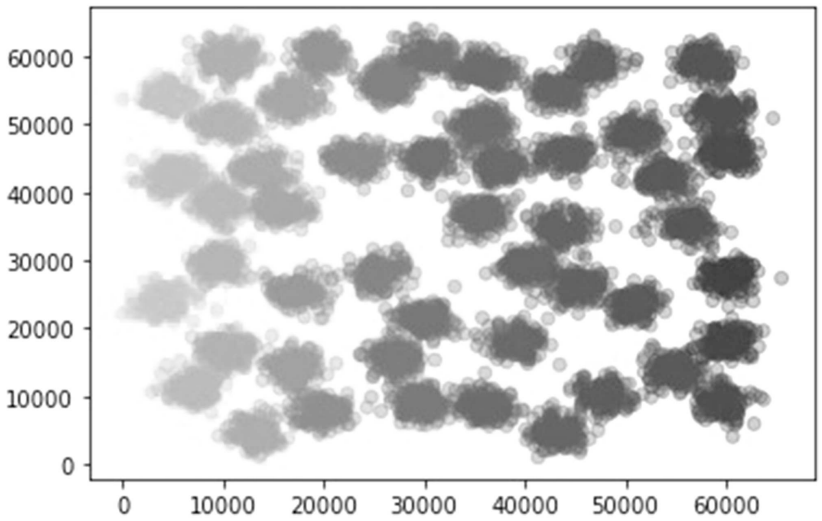

Figure 7.3 A3 data set.

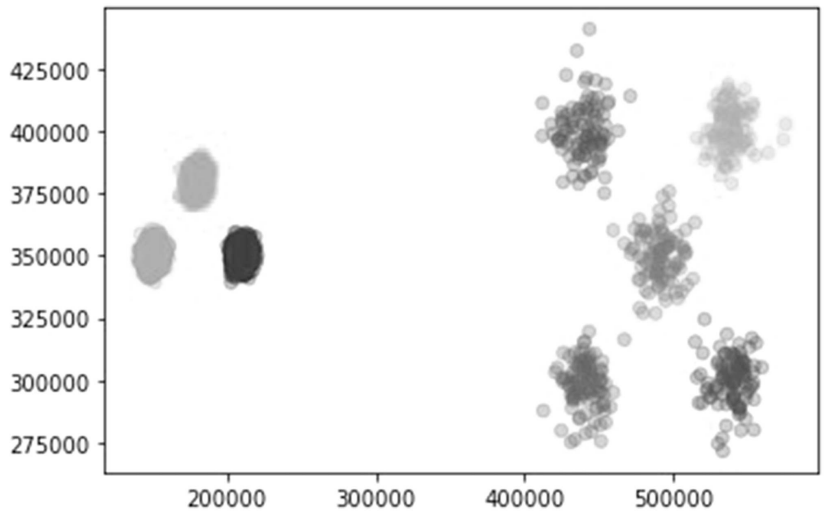

Figure 7.4 Unbalance data set.

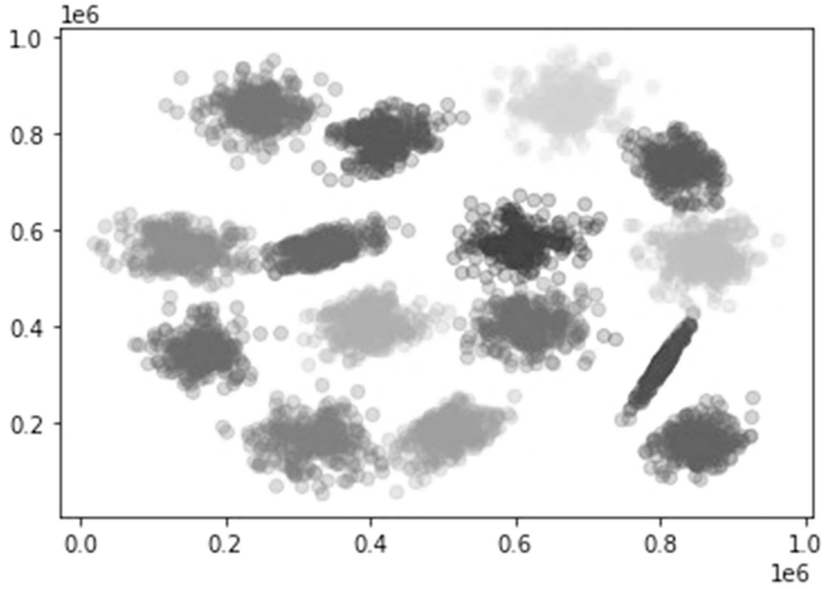

Figure 7.5 S1 data set.

increasing dimensions. The data sets have data with dimensions 32, 64, 128, 256, 512 and 1024, respectively. The G2 data sets have 100 sets of data with increasing dimensions and overlap of clusters. With an increase of dimension, the standard deviation of data points of each cluster has also been increased to introduce increasing overlap. The dimension of data sets ranges from 2 to 1024; at the same time, the standard deviation ranges from 10 to 100. For the data set Unbalance, we have eight clusters with an imbalance of data points in each cluster. In other words, a few clusters contain more data points, and a few clusters contain very few data points. Figure 7.4 shows the unbalanced data set.

We have run Hard GMM and Usual GMM algorithms 100 times on each data to measure and compare the mean performance of both models. The

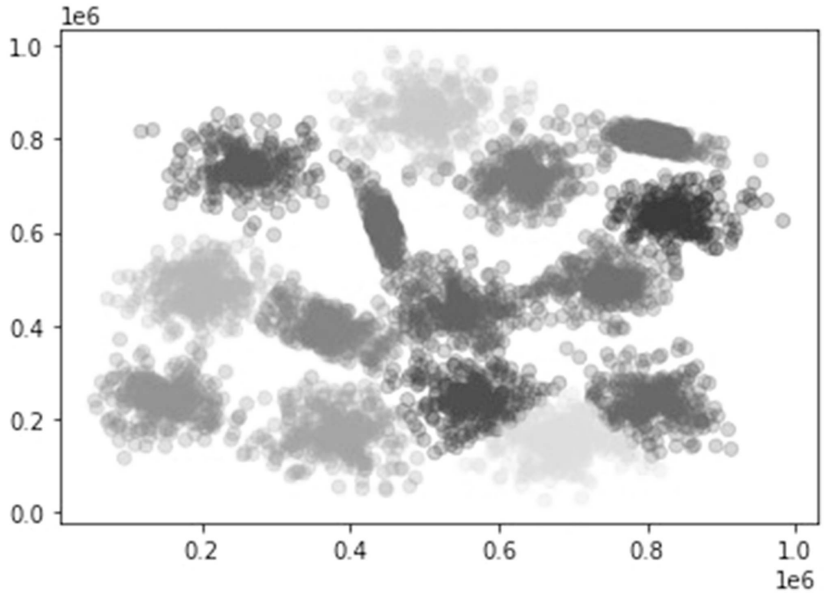

Figure 7.6 S2 data set.

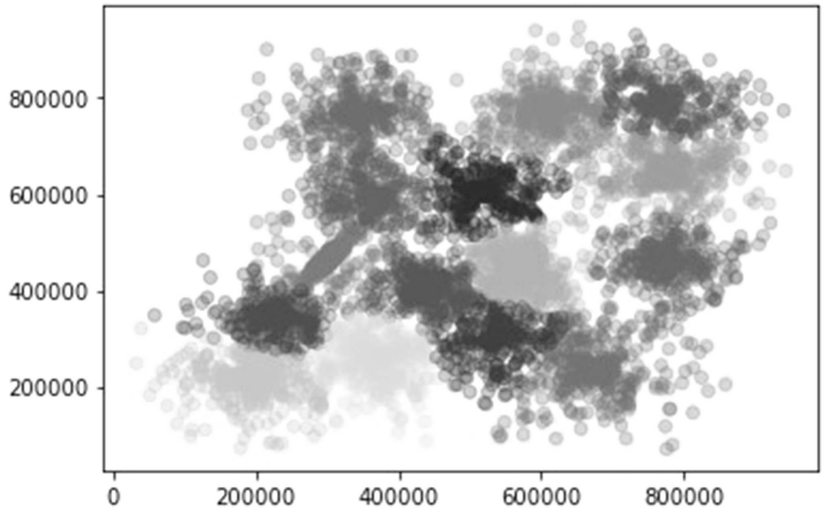

Figure 7.7 S3 data set.

algorithm for usual GMM is available in Scikit-learn [52], a machine learning library in Python. The algorithm for hard GMM is also written in Python. The experiments were done on a machine with 32 gigabytes of RAM and multi-threaded CPU. We have checked three measures to evaluate the performance.

- Accuracy: The total accurate classifications, divided by the number of observations.
- Precision: True positives, divided by sum of true positives and false positives.
- Recall: True positives, divided by the sum of true positives and false negatives.

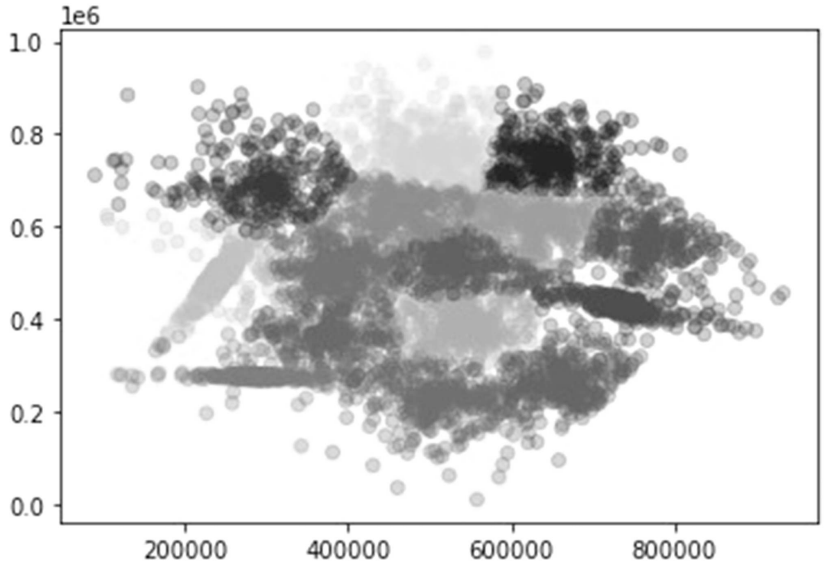

Figure 7.8 S4 data set.

A detailed description of all the measures can be found in [53]. Additionally, we have also provided computational run-time (in seconds) of the models where applicable. The results of the experiments on data sets A and S are given below. For datasets Dim, G2 and unbalance, both Hard GMM and Usual GMM are found to have given 100% success in terms of accuracy, precision and recall. Table 7.2, Table 7.3 and Table 7.4 display the detailed results of data sets A1, A2 and A3, respectively. Whereas the results of data sets S1, S2, S3 and S4 are shown in Table 7.5, Table 7.6, Table 7.7 and Table 7.8.

From the above results in Figure 7.9 and Figure 7.10, we see that Hard GMM works better than Usual GMM on Data sets A and S in terms of accuracy, precision, and recall. We have also observed that Hard GMM is much more consistent for data sets A and S on 100 runs, as the standard deviation is much less for Hard GMM for the data sets A and S. From Figure 7.9 we see that, with an increasing number of clusters, the performance of Hard GMM degrades in terms of accuracy, precision, and recall. However, Usual GMM does not show any significant pattern in performance in terms of accuracy, precision and recall when it comes to the increasing number of clusters in the data. From Figure 7.10 we see that the performance of both Hard GMM and Usual GMM drops significantly with increasing overlap of clusters in terms of accuracy, precision, and recall. But, we have seen no effect of increasing dimension in the performance of both Hard GMM and Usual GMM, as every time over 100 runs, the models have shown 100% success in terms of accuracy, precision, and recall on dim data sets. It is evident that increasing dimension has very little to do in terms of performance if the clusters are distinct. For G2 data sets, we had 100 sets of data with both increasing dimensions and overlap. But, in this case, also, we have observed 100% success of both models to identify the clusters on all 100 runs.

Table 7.2 Performance of Hard GMM and Usual GMM on dataset A1

Measures	*AHG*	*AUG*	*PHG*	*PUG*	*RHG*	*RUG*	*RTHG*	*RTUG*
Mean	0.996920	0.875690	0.996878	0.976668	0.996977	0.959366	0.178015	0.077974
Standard Deviation	0.000143	0.116462	0.000151	0.014935	0.000131	0.030430	0.032193	0.007880
Minimum	0.996667	0.626000	0.996610	0.942585	0.996745	0.882558	0.161511	0.062025
First Quartile	0.997000	0.796000	0.996962	0.966382	0.997050	0.938932	0.166006	0.074326
Median	0.997000	0.873333	0.996962	0.972818	0.997050	0.942855	0.168327	0.077453
Third Quartile	0.997000	0.992333	0.996962	0.992125	0.997050	0.992760	0.176055	0.083743
Maximum	0.997000	0.992333	0.996962	0.992125	0.997050	0.992760	0.443339	0.100296

AHG: Accuracy of Hard GMM, AUG: Accuracy of Usual GMM, PHG: Precision of Hard GMM, PUG: Precision of Usual GMM, RHG: Recall of Hard GMM, RUG: Recall of Usual GMM RTHG: Runtime of Hard GMM, RTUG: Run-time of Usual GMM.

Table 7.3 Performance of Hard GMM and Usual GMM on dataset A2

Measures	*AHG*	*AUG*	*PHG*	*PUG*	*RHG*	*RUG*	*RTHG*	*RTUG*
Mean	0.977640	0.863086	0.994175	0.977943	0.991563	0.957694	1.502774	0.185822
Standard Deviation	0.045927	0.073965	0.006402	0.008992	0.012301	0.018494	0.152975	0.033539
Minimum	0.783238	0.689143	0.973484	0.951427	0.961736	0.905962	1.356514	0.128678
First Quartile	0.996762	0.807000	0.996731	0.971669	0.996865	0.936981	1.427737	0.167785
Median	0.996952	0.869333	0.996916	0.978609	0.997036	0.963351	1.468551	0.177822
Third Quartile	0.997143	0.915571	0.997116	0.983020	0.997209	0.965244	1.524925	0.193811
Maximum	0.997333	0.993524	0.997305	0.993429	0.997393	0.993822	2.650507	0.394115

AHG: Accuracy of Hard GMM, AUG: Accuracy of Usual GMM, PHG: Precision of Hard GMM, PUG: Precision of Usual GMM, RHG: Recall of Hard GMM, RUG: Recall of Usual GMM RTHG: Runtime of Hard GMM, RTUG: Run-time of Usual GMM.

Table 7.4 Performance of Hard GMM and Usual GMM on dataset A3

Measures	*AHG*	*AUG*	*PHG*	*PUG*	*RHG*	*RUG*	*RTHG*	*RTUG*
Mean	0.934595	0.872537	0.990532	0.980345	0.983551	0.960637	1.786347	0.280679
Standard Deviation	0.056556	0.063956	0.005756	0.007250	0.010908	0.015886	0.315283	0.070037
Minimum	0.817733	0.680667	0.980392	0.959247	0.972147	0.909704	1.478938	0.196710
First Quartile	0.889633	0.835833	0.985925	0.976154	0.974674	0.952880	1.556423	0.240818
Median	0.929133	0.877400	0.988208	0.981027	0.977209	0.955084	1.683763	0.259527
Third Quartile	0.997467	0.919100	0.997456	0.985758	0.997513	0.974323	1.891815	0.298878
Maximum	0.997867	0.995333	0.997851	0.995275	0.997906	0.995483	3.232524	0.718938

AHG: Accuracy of Hard GMM, AUG: Accuracy of Usual GMM, PHG: Precision of Hard GMM, PUG: Precision of Usual GMM, RHG: Recall of Hard GMM, RUG: Recall of Usual GMM RTHG: Runtime of Hard GMM, RTUG: Run-time of Usual GMM.

Table 7.5 Performance of Hard GMM and Usual GMM on dataset S1

Measures	*AHG*	*AUG*	*PHG*	*PUG*	*RHG*	*RUG*	*RTHG*	*RTUG*
Mean	0.998200	0.974388	0.998186	0.995816	0.998180	0.988442	1.560566	0.165872
Standard Deviation	0.000000	0.063192	0.000000	0.005272	0.000000	0.023293	0.209326	0.041186
Minimum	0.998200	0.715400	0.998186	0.970759	0.998180	0.930539	1.304745	0.093759
First Quartile	0.998200	0.997800	0.998186	0.997787	0.998180	0.997793	1.445687	0.147343
Median	0.998200	0.997800	0.998186	0.997787	0.998180	0.997793	1.483772	0.151386
Third Quartile	0.998200	0.997800	0.998186	0.997787	0.998180	0.997793	1.593771	0.159478
Maximum	0.998200	0.997800	0.998186	0.997787	0.998180	0.997793	2.342927	0.347596

AHG: Accuracy of Hard GMM, AUG: Accuracy of Usual GMM, PHG: Precision of Hard GMM, PUG: Precision of Usual GMM, RHG: Recall of Hard GMM, RUG: Recall of Usual GMM RTHG: Runtime of Hard GMM, RTUG: Run-time of Usual GMM.

Table 7.6 Performance of Hard GMM and Usual GMM on dataset S2

Measures	*AHG*	*AUG*	*PHG*	*PUG*	*RHG*	*RUG*	*RTHG*	*RTUG*
Mean	0.992364	0.939530	0.992504	0.980173	0.992340	0.959469	1.678559	0.182416
Standard Deviation	0.000230	0.070728	0.000224	0.007245	0.000231	0.032618	0.378828	0.054915
Minimum	0.991400	0.703600	0.991574	0.950935	0.991350	0.908707	1.369272	0.117463
First Quartile	0.992200	0.908900	0.992336	0.977107	0.992189	0.920079	1.482488	0.157468
Median	0.992400	0.984600	0.992542	0.984818	0.992380	0.984725	1.543244	0.161766
Third Quartile	0.992600	0.984800	0.992733	0.985023	0.992571	0.984927	1.690107	0.180889
Maximum	0.992800	0.984800	0.992953	0.985038	0.992767	0.984927	3.640522	0.440569

AHG: Accuracy of Hard GMM, AUG: Accuracy of Usual GMM, PHG: Precision of Hard GMM, PUG: Precision of Usual GMM, RHG: Recall of Hard GMM, RUG: Recall of Usual GMM RTHG: Runtime of Hard GMM, RTUG: Run-time of Usual GMM.

Table 7.7 Performance of Hard GMM and Usual GMM on dataset S3

Measures	*AHG*	*AUG*	*PHG*	*PUG*	*RHG*	*RUG*	*RTHG*	*RTUG*
Mean	0.973476	0.850652	0.973608	0.936191	0.973091	0.905184	1.817396	0.240185
Standard Deviation	0.000627	0.092452	0.000622	0.014524	0.000655	0.038538	0.352305	0.118835
Minimum	0.972200	0.532600	0.972323	0.900290	0.971763	0.774384	1.417350	0.132710
First Quartile	0.973000	0.794100	0.973146	0.924001	0.972592	0.879216	1.532039	0.173736
Median	0.973400	0.846100	0.973554	0.938519	0.973041	0.891337	1.685317	0.206736
Third Quartile	0.974000	0.948000	0.974109	0.949817	0.973637	0.947924	2.026184	0.235066
Maximum	0.974800	0.949800	0.974872	0.951637	0.974501	0.949731	3.048964	0.718240

AHG: Accuracy of Hard GMM, AUG: Accuracy of Usual GMM, PHG: Precision of Hard GMM, PUG: Precision of Usual GMM, RHG: Recall of Hard GMM, RUG: Recall of Usual GMM RTHG: Runtime of Hard GMM, RTUG: Run-time of Usual GMM.

Table 7.8 Performance of Hard GMM and Usual GMM on dataset S4

Measures	*AHG*	*AUG*	*PHG*	*PUG*	*RHG*	*RUG*	*RTHG*	*RTUG*
Mean	0.967522	0.846350	0.967441	0.907800	0.966902	0.887625	1.615013	0.230549
Standard Deviation	0.002470	0.066463	0.002502	0.010089	0.002422	0.028414	0.186901	0.046512
Minimum	0.962200	0.709000	0.961969	0.868938	0.961627	0.835819	1.397856	0.174893
First Quartile	0.965400	0.784150	0.965274	0.908700	0.964847	0.863440	1.504002	0.202314
Median	0.967700	0.903000	0.967633	0.909992	0.967221	0.912310	1.564790	0.213907
Third Quartile	0.969600	0.904600	0.969546	0.912384	0.968908	0.913579	1.643767	0.251595
Maximum	0.972000	0.908400	0.972176	0.925609	0.971375	0.916840	2.434432	0.434772

AHG: Accuracy of Hard GMM, AUG: Accuracy of Usual GMM, PHG: Precision of Hard GMM, PUG: Precision of Usual GMM, RHG: Recall of Hard GMM, RUG: Recall of Usual GMM RTHG: Runtime of Hard GMM, RTUG: Run-time of Usual GMM.

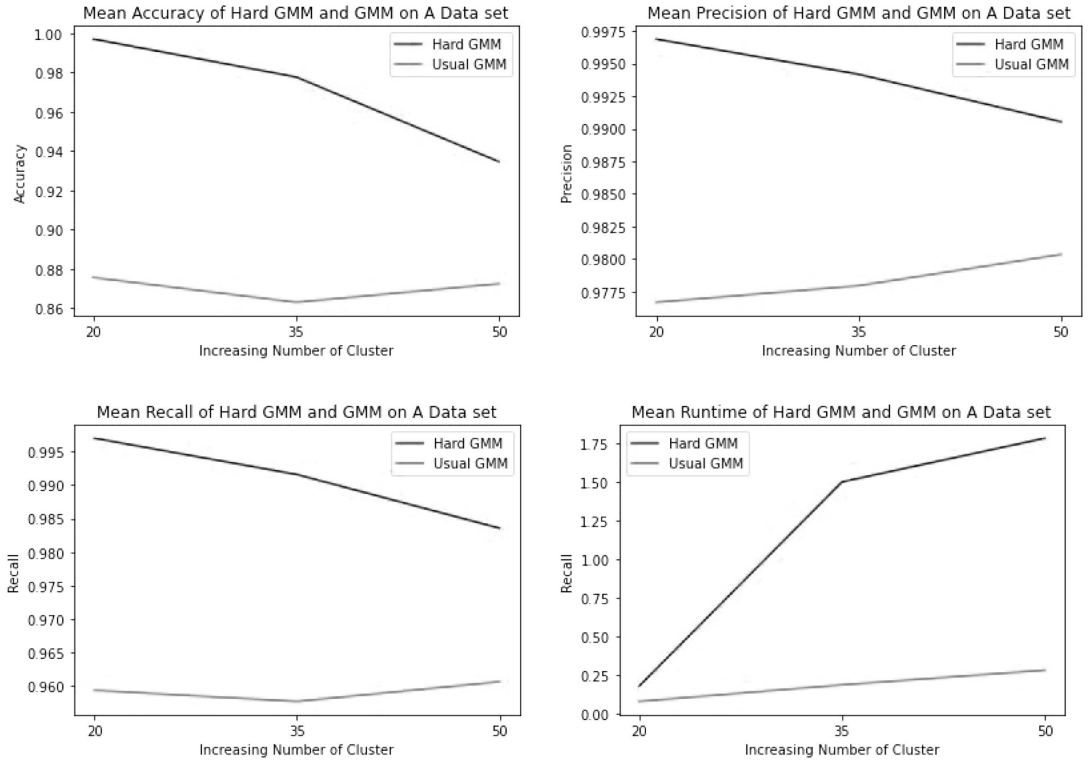

Figure 7.9 Mean accuracy, mean precision, mean recall and mean run-time with increasing number of clusters on A data sets.

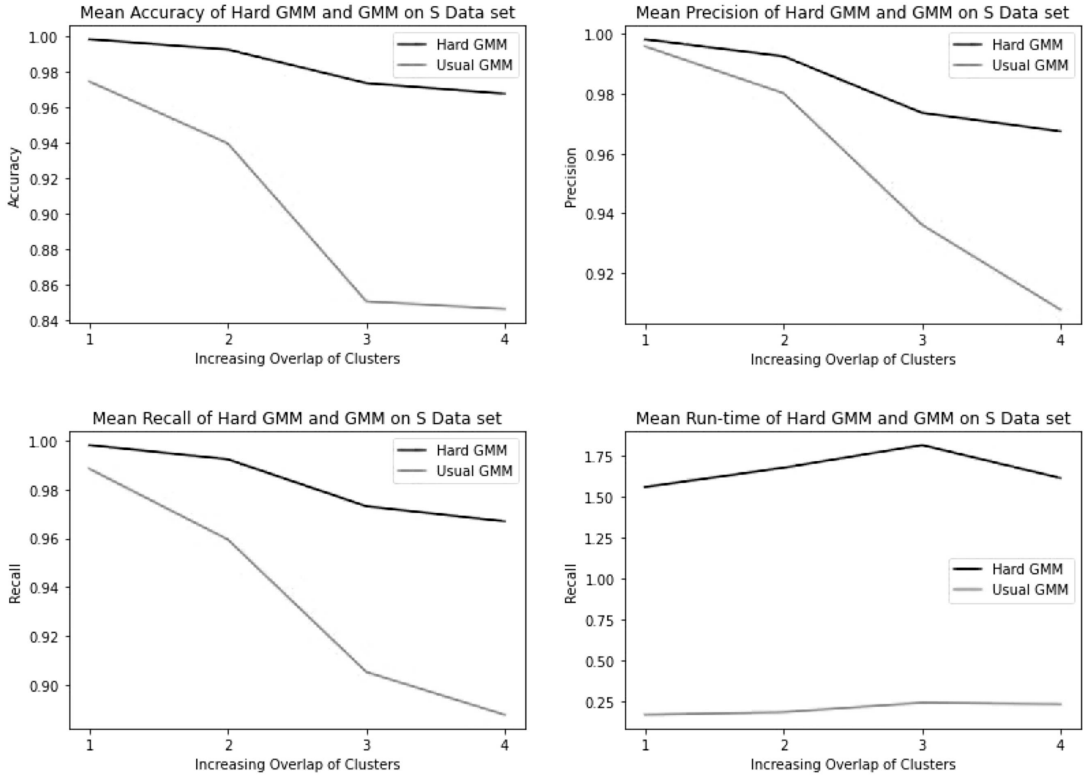

Figure 7.10 Mean accuracy, mean precision, mean recall and mean run-time with increasing overlap of clusters on S data sets.

The result occurs mostly because, in higher dimensional space, the data becomes sparse, and clusters keep on moving far away from each other. Surprisingly, on the unbalanced data set also, we have seen 100% success for both models. It is to be noted that the very popular KMeans algorithm works poorly if the data is imbalanced. Nevertheless, the Gaussian Mixture model, both with hard EM and Usual EM, seems to work very well on imbalanced data. When it comes to computational run time, we see that usual GMM is slightly faster. The clustering strategy of modified Hard EM stops early convergence of the algorithm with non-occupied clusters. From the results, we see that Hard EM offers better results in some situations with a little more run time.

7.4 REAL DATA APPLICATION

We have compared the performance of Hard GMM and Usual GMM on two real data sets. Please note that our study aims to compare the performance of the two methods and not to provide an optimum solution for the problems.

7.4.1 Breast cancer data

The relationship between several risk factors and breast cancer was studied by [54] by assessing hyperresistinemia and metabolic dysregulation in breast cancer. Between 2009 and 2013, women who had been newly diagnosed with breast cancer were recruited from the University Hospital Centre of Coimbra (CHUC). For each patient, the diagnosis was made by positive mammography, and it was histologically confirmed. Before surgery and treatment, all samples were collected, and all the patients with treatment before the consultation were excluded. Healthy female volunteers were selected and enrolled in the study as controls. All patients had no prior cancer treatment, and all participants were free from any infection or other acute diseases or comorbidities at the time of enrollment in the study. Later, [55] also used the data to build supervised learning models and provided an idea for a cheap and effective biomarker for breast cancer. In the dataset, we have nine clinical and biochemical factors, namely: Age (years), BMI (kg/m2),Glucose (mg/dL), Insulin (μU/mL), HOMA, Leptin (ng/mL), Adiponectin (μg/mL), Resistin (ng/mL) and MCP-1(pg/dL). The data can be downloaded from UCI Machine Learning Repository.

7.4.2 Yeast cell cycle data

The fluctuation of expression levels of approximately 6,000 genes over two cell cycles (17-time points) was shown by [56]. Later, [57] studied a subset of 384 genes where they had expression levels peaking at different times corresponding to the five phases of the cell cycle, namely: Early G1, Late

Table 7.9 Performance of Hard GMM and Usual GMM on Real Data

	Breast cancer data	*Yeast cell cycle data*
Accuracy of Hard GMM	0.543103	0.731771
Accuracy of Usual GMM	0.534482	0.559896
Precision of Hard GMM	0.578726	0.726260
Precision of Usual GMM	0.572716	0.623411
Recall of Hard GMM	0.642159	0.725608
Recall of Usual GMM	0.651250	0.585197
Run-time of Hard GMM	0.016630	0.075423
Run-time of Usual GMM	0.021233	0.076323

The aim of our study is to build a Gaussian Mixture Model using these features and cluster the data points into two categories, namely: healthy controls and patients. In this case, $k = 2$, $p = 9$ and $N = 116$.

G1, S, G2, and M. This data set has been previously used by other authors as well (see [58]). This data is also available at UCI Machine Learning Repository. The aim of the study is to cluster the gene expressions into five categories. In this case $k = 5$, $p = 17$ and $N = 384$.

From the above result in Table 7.9, we see that Hard GMM works better than Usual GMM on both data sets in terms of accuracy and precision. In terms of recall, Usual GMM works better in Breast Cancer data, and Hard GMM works better in cell cycle data. From the results, we understand that the performance of GMM depends mostly on overlap of data in different clusters than a number of clusters or dimensions. In terms of run time, we see that Hard EM was faster than Usual GMM with better results.

7.5 CONCLUSION

In our study, we wanted to address the question if Hard EM really works worse as it is assumed due to its theoretical disadvantages. We also wanted to verify how Hard EM performs with respect to Usual EM in different situations in clustering problems. In our study, we have implemented Hard EM with some modifications, and we have compared the performance of Hard EM and Usual EM on both basic benchmark data sets and real data sets. Furthermore, from our experiments, we have found that it can not be said that Hard EM works worse than Usual EM for clustering analysis. In fact, on many occasions, like an increasing number of clusters and increasing overlap, Hard EM has been found to perform better than Usual EM.

We have already discussed that Hard EM is computationally less intensive, and it is easy to implement, which fulfills the important criteria of choosing a suitable algorithm [59,60]. However, if the limitations of a suitable algorithm are not known, people tend to choose a less accurate algorithm whose limitations are well known beforehand. We have proposed

some modifications to Hard EM which stops faster convergence with poor results. Nevertheless, from our experiment we have seen that the difference in run time is not significant. On real data sets the run time of Hard EM was found to be less than that of Usual EM. Moreover, the performance of Hard EM has been enhanced due to the modifications. We have done an extensive experiment on different situations and ran the algorithm on each data 100 times. We have noticed that the performance of Hard EM is very consistent. The robustness of the algorithm can be seen in a varied range of situations. It is often assumed that increasing dimension has an inverse effect on the performance of clustering algorithms [61]. We have found that increasing dimension has almost nothing to do with the performance of Hard GMM if the clusters are distinct and separated. However, proper initialization technique must be observed in order to produce good results.

It is known that KMeans works poorly for imbalanced data [36]. However, for Gaussian Mixture Model, both hard and usual EM are found to work well for imbalanced data points in the cluster. The condition which affects the performance of Hard EM the most is the overlap of clusters. We have seen that the performance drops significantly for both Hard and Usual GMM when the overlap is increased. We have also noticed that for an increasing number of clusters, the performance of Hard GMM drops, whereas Usual GMM shows no significant pattern.

Our study has shown that despite having some disadvantages, Hard EM works at par with Usual GMM. The results on real data sets involving gene expression and biochemical analysis confirm the same. Thus, our study recommends the use of Modified Hard EM for clustering purposes. The proposed model is expected to yield a result at least as good as a standard EM algorithm in the situations we have considered in our study.

REFERENCES

1. 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.
2. Jean-Patrick Baudry and Gilles Celeux. Em for mixtures. Statistics and Computing, 25(4): 713–726, 2015.
3. Geoffrey J. McLachlan, Sharon X. Lee, and Suren I. Rathnayake. Finite mixture models. Annual Review of Statistics and Its Application, 6:355–378, 2019.
4. Jiayi Ma, Xingyu Jiang, Junjun Jiang, and Yuan Gao. Feature-guided Gaussian mixture model for image matching. Pattern Recognition, 92:231–245, 2019.
5. Israel Dejene Gebru, Xavier Alameda-Pineda, Florence Forbes, and Radu Horaud. Em algorithms for weighted-data clustering with application to audio-visual scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(12):2402–2415, 2016.

6. Frederick Jelinek. Continuous speech recognition by statistical methods. Proceedings of the IEEE, 64(4):532–556, 1976.
7. Lawrence Rabiner. Fundamentals of speech recognition. Fundamentals of Speech Recognition, 1993.
8. Hermann Ney, Volker Steinbiss, Reinhold Haeb-Umbach, B-H Tran, and Ute Essen. An overview of the Philips research system for large vocabulary continuous speech recognition. International Journal of Pattern Recognition and Artificial Intelligence, 8(01):33–70, 1994.
9. Biing Hwang Juang and Laurence R. Rabiner. Hidden Markov models for speech recognition. Technometrics, 33(3):251–272, 1991.
10. Nikko Str¨om, Lee Hetherington, Timothy J. Hazen, Eric Sandness, and James Glass. Acoustic modeling improvements in a segment-based speech recognizer. In Proceedings of IEEE ASRU Workshop, 1999.
11. Volker Steinbiss, Herman Ney, X. Aubert, Stefan Besling, Christian Dugast, Ute Essen, Daryl Geller, Reinhold Haeb-Umbach, Reinhard Kneser, H-G Meier, et al. The Philips research system for continuous-speech recognition. Philips Journal of Research, 49(4):317–352, 1995.
12. Yejin Choi and Claire Cardie. Structured local training and biased potential functions for conditional random fields with application to coreference resolution. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 65–72, 2007.
13. Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. What is the jeopardy model? a quasisynchronous grammar for qa. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, 2007.
14. Sharon Goldwater and Mark Johnson. Representational bias in unsupervised learning of syllable structure. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 112–119, 2005.
15. John DeNero and Dan Klein. The complexity of phrase alignment problems. In *Proceedings of ACL-08: HLT, Short Papers*, pages 25–28, 2008.
16. Valentin I. Spitkovsky, Hiyan Alshawi, Dan Jurafsky, and Christopher D. Manning. Viterbi training improves unsupervised dependency parsing. In Proceedings of the Fourteenth Conference on Computational Natural Language Learning, pages 9–17, 2010.
17. Gang Ji and Jeff Bilmes. Backoff model training using partially observed data: Application to dialog act tagging. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 280–287, 2006.
18. Franz Josef Och and Hermann Ney. Improved statistical alignment models. In *Proceedings of the 38th annual meeting of the association for computational linguistics*, pages 440–447, 2000.
19. Georg B. Ehret, Patrick Reichenbach, Ulrike Schindler, Curt M. Horvath, Stefan Fritz, Markus Nabholz, and Philipp Bucher. DNA binding specificity of different stat proteins: Comparison of in vitro specificity with natural target sites* 210. Journal of Biological Chemistry, 276 (9):6675–6688, 2001.

20. Uwe Ohler, Heinrich Niemann, Guo-chun Liao, and Gerald M. Rubin. Joint modeling of DNA sequence and physical properties to improve eukaryotic promoter recognition. Bioinformatics, 17(suppl 1):S199–S206, 2001.
21. Alexandre Lomsadze, Vardges Ter-Hovhannisyan, Yury O. Chernoff, and Mark Borodovsky. Gene identification in novel eukaryotic genomes by self-training algorithm. Nucleic Acids Research, 33(20):6494–6506, 2005.
22. Dhiraj Joshi, Jia Li, and James Ze Wang. A computationally efficient approach to the estimation of two-and three-dimensional hidden Markov models. IEEE Transactions on Image Processing, 15(7):1871–1886, 2006.
23. Gilles Celeux and G´erard Govaert. A classification em algorithm for clustering and two stochastic versions. Computational Statistics & Data Analysis, 14(3):315–332, 1992.
24. Chris Fraley and Adrian E. Raftery. Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association, 97(458):611–631, 2002.
25. Radford M. Neal and Geoffrey E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In Learning in Graphical Models, pages 355–368. Springer, 1998.
26. Stuart Lloyd. Least squares quantization in PCM. IEEE transactions on information theory, 28(2):129–137, 1982.
27. Philip A. Chou, Tom Lookabaugh, and Robert M. Gray. Entropy-constrained vector quantization. IEEE Transactions on acoustics, speech, and signal processing, 37(1):31–42, 1989.
28. Ju¨ri Lember and Alexey Koloydenko. Adjusted Viterbi training. Probability in the Engineering and Informational Sciences, 21(3):451–475, 2007.
29. Armen Allahverdyan and Aram Galstyan. Comparative analysis of Viterbi training and maximum likelihood estimation for hmms. In Advances in Neural Information Processing Systems, pages 1674–1682. Citeseer, 2011.
30. Alexey Koloydenko, Meelis K¨a¨arik, and Ju¨ri Lember. On adjusted Viterbi training. Acta Applicandae Mathematicae, 96(1):309–326, 2007.
31. Brian G. Leroux. Maximum-likelihood estimation for hidden Markov models. Stochastic processes and their applications, 40(1):127–143, 1992.
32. Yariv Ephraim and Neri Merhav. Hidden Markov processes. IEEE Transactions on Information Theory, 48(6):1518–1569, 2002.
33. Z. Hatala and F. Puturuhu. Viterbi algorithm and its application to Indonesian speech recognition. In Journal of Physics: Conference Series, volume 1752, page 012085. IOP Publishing, 2021.
34. Emilie M. Shireman, Douglas Steinley, and Michael J. Brusco. Local optima in mixture modeling. Multivariate Behavioral Research, 51(4):466–481, 2016.
35. Christophe Biernacki, Gilles Celeux, and G´erard Govaert. Choosing starting values for the em algorithm for getting the highest likelihood in multivariate Gaussian mixture models. Computational Statistics & Data Analysis, 41(3–4):561–575, 2003.
36. Pasi Fr¨anti and Sami Sieranoja. K-means properties on six clustering benchmark datasets. Applied Intelligence, 48(12):4743–4759, 2018.
37. Kevin P. Murphy. Machine learning: A probabilistic perspective. MIT Press, 2012.

38. V. Melnykov and I. Melnykov. Initializing the EM algorithm in Gaussian mixture models with an unknown number of components. Computational Statistics & Data Analysis, 56: 1381–1395, 2012.
39. J. Hipp and D. Bauer. Local solutions in the estimation of growth mixture models. Psychological Methods, 11: 36, 2006.
40. G. Lubke and B. Muth´en. Performance of factor mixture models as a function of model size, covariate effects, and class-specific parameters. Structural Equation Modeling: A Multidisciplinary Journal, 14: 26–47, 2007.
41. D. Steinley and M. Brusco. Evaluating mixture modeling for clustering: recommendations and cautions. Psychological Methods, 16: 63, 2011.
42. D. Bartholomew, M. Knott, and I. Moustaki. Latent variable models and factor analysis: A unified approach. John Wiley & Sons, 2011.
43. Shireman, E., Steinley, D., and Brusco, M. Examining the effect of initialization strategies on the performance of Gaussian mixture modeling. Behavior Research Methods, 49: 282–293, 2015, 12, 10.3758/s13428-015-0697-6
44. Yang, M., Lai, C., and Lin, C. A robust EM clustering algorithm for Gaussian mixture models. Pattern Recognition, 45, 3950–3961, 2012.
45. Amke Caliebe and U. Rosler. Convergence of the maximum a posteriori path estimator in hidden Markov models. IEEE Transactions on Information Theory, 48(7): 1750–1758, 2002.
46. Ulrike Von Luxburg, Robert C. Williamson, and Isabelle Guyon. Clustering: Science or art? In Proceedings of ICML workshop on unsupervised and transfer learning, pages 65–79. JMLR Workshop and Conference Proceedings, 2012.
47. Ismo K¨arkk¨ainen and Pasi Fr¨anti. Dynamic local search algorithm for the clustering problem. University of Joensuu Joensuu, Finland, 2002.
48. Pasi Fr¨anti and Olli Virmajoki. Iterative shrinking method for clustering problems. Pattern Recognition, 39(5):761–775, 2006.
49. Pasi Franti, Olli Virmajoki, and Ville Hautamaki. Fast agglomerative clustering using a knearest neighbor graph. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(11):1875–1881, 2006.
50. Pasi Fr¨anti, Radu Mariescu-Istodor, and Caiming Zhong. Xnn graph. In Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), pages 207–217. Springer, 2016.
51. Mohammad Rezaei and Pasi Fr¨anti. Set matching measures for external cluster validity. IEEE Transactions on Knowledge and Data Engineering, 28(8):2173–2186, 2016.
52. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.
53. Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4):427–437, 2009.
54. Joana Crisostomo, Paulo Matafome, Daniela Santos-Silva, Ana L. Gomes, Manuel Gomes, Miguel Patr´ıcio, Liliana Letra, Ana B. Sarmento-Ribeiro, Lelita Santos, and Raquel Sei¸ca. Hyperresistinemia and metabolic dysregulation: A risky crosstalk in obese breast cancer. Endocrine, 53(2):433–442, 2016.

55. Miguel Patr´ıcio, Jos´e Pereira, Joana Cris´ostomo, Paulo Matafome, Manuel Gomes, Raquel Sei¸ca, and Francisco Caramelo. Using resistin, glucose, age and BMI to predict the presence of breast cancer. BMC Cancer, 18(1):1–8, 2018.
56. Raymond J. Cho, Michael J. Campbell, Elizabeth A. Winzeler, Lars Steinmetz, Andrew Conway, Lisa Wodicka, Tyra G. Wolfsberg, Andrei E. Gabrielian, David Landsman, David J. Lockhart, et al. A genome-wide transcriptional analysis of the mitotic cell cycle. Molecular Cell, 2(1): 65–73, 1998.
57. Ka Yee Yeung, Chris Fraley, Alejandro Murua, Adrian E. Raftery, and Walter L. Ruzzo. Model-based clustering and data transformations for gene expression data. Bioinformatics, 17(10):977–987, 2001.
58. Zhe Liu, Yu-qing Song, Cong-hua Xie, and Zheng Tang. A new clustering method of gene expression data based on multivariate Gaussian mixture models. Signal, Image and Video Processing, 10(2):359–368, 2016.
59. Tomi Kinnunen, Ilja Sidoroff, Marko Tuononen, and Pasi Fr¨anti. Comparison of clustering methods: A case study of text-independent speaker modeling. Pattern Recognition Letters, 32(13):1604–1617, 2011.
60. Xiaojuan Huang, Li Zhang, Bangjun Wang, Fanzhang Li, and Zhao Zhang. Feature clustering based support vector machine recursive feature elimination for gene selection. Applied Intelligence, 48(3):594–607, 2018.
61. Carlotta Domeniconi, Dimitrios Gunopulos, Sheng Ma, Bojun Yan, Muna Al-Razgan, and Dimitris Papadopoulos. Locally adaptive metrics for clustering high dimensional data. Data Mining and Knowledge Discovery, 14(1):63–97, 2007.

Chapter 6

Flexible Multivariate Mixture Models: A Comprehensive Approach for Modeling Mixtures of Non-Identical Distributions

Summary

This paper introduces a novel and flexible approach to constructing mixture models that incorporate both identical and non-identical distributions, such as combinations of multivariate skew normal and multivariate generalized hyperbolic distributions. Unlike traditional models, which use mixtures of only identical distributions, this new framework allows for all possible permutations of distributions. It demonstrates that conventional mixture models are specific cases within this broader framework. The effectiveness of the proposed model is validated through its application to both simulated and real-world data, showing its ability to identify underlying patterns and accurately estimate parameters.

Contributing Article

Pal, Samyajoy, and Christian Heumann. "Flexible Multivariate Mixture Models: A Comprehensive Approach for Modeling Mixtures of Non-Identical Distributions." *International Statistical Review* (2024). https://doi.org/10.1111/insr.12593.

Author Contributions

- Samyajoy Pal: Conceptualization, Methodology, Software, Writing – original draft
- Christian Heumann: Supervision, Writing – review & editing

International Statistical Review

International Statistical Review (2024) doi: 10.1111/insr.12593

Flexible Multivariate Mixture Models: A Comprehensive Approach for Modeling Mixtures of Non-Identical Distributions

Samyajoy Pal and Christian Heumann

Department of Statistics, LMU Munich, Germany
Correspondence *Samyajoy Pal, Department of Statistics, LMU Munich, Germany. Email: Samyajoy.Pal@stat.uni-muenchen.de*

Summary

The mixture models are widely used to analyze data with cluster structures and the mixture of Gaussians is most common in practical applications. The use of mixtures involving other multivariate distributions, like the multivariate skew normal and multivariate generalised hyperbolic, is also found in the literature. However, in all such cases, only the mixtures of identical distributions are used to form a mixture model. We present an innovative and versatile approach for constructing mixture models involving identical and non-identical distributions combined in all conceivable permutations (e.g. a mixture of multivariate skew normal and multivariate generalised hyperbolic). We also establish any conventional mixture model as a distinctive particular case of our proposed framework. The practical efficacy of our model is shown through its application to both simulated and real-world data sets. Our comprehensive and flexible model excels at recognising inherent patterns and accurately estimating parameters.

Key words: EM algorithm; maximum likelihood estimates; mixture model; mixture of non-identical distributions; multivariate generalised hyperbolic distribution; multivariate skew normal distribution.

1 Introduction

The analysis of complex data structures exhibiting inherent clustering has been a persistent challenge across various fields, spanning from biology (Balaban *et al.*, 2019; Petegrosso *et al.*, 2020) and finance (Li *et al.*, 2021) to image processing (Kim *et al.*, 2020) and social sciences (Grimmer *et al.*, 2021). There are two prominent paradigms in clustering: model-based approaches and algorithms rooted in similarity or distance measures. In the former, such as the Gaussian mixture model (GMM) (McLachlan *et al.*, 2019), the focus is on extracting clusters by fitting a mixture of distributions to the data. Meanwhile, similarity-based algorithms such as hierarchical clustering (Ward Jr, 1963) and KMeans (MacQueen *et al.*, 1967) create clusters by quantifying the relationships or distances between data points.

The mixture models, in particular, have emerged as a versatile and widely adopted framework for addressing the challenge of clustering complex data sets. These models enable the identification of latent sub-populations within heterogeneous data sets, accommodating variations in observed data points while capturing the underlying distribution of the data. The Gaussian mixture model, a quintessential representative of mixture models, assumes that data within each

cluster follows a multivariate Gaussian distribution. Despite its prevalence, the practicality of mixture models extends beyond Gaussian distributions. The literature increasingly highlights the potential of alternative multivariate distributions, such as the multivariate t-distribution (MVT), the multivariate skew normal (MSN) and multivariate generalised hyperbolic (MGH) distributions, as promising candidates for modeling complex data structures. Browne & McNicholas (2015) have shown how to model multivariate data with mixtures of multivariate generalised hyperbolic distributions. On the other hand, Lin (2009) and Abe *et al.* (2021) provided vivid descriptions for fitting mixtures of multivariate skew normal distributions. We find mixture models with different special forms of multivariate generalised hyperbolic distribution as well. Mixtures involving the multivariate normal inverse Gaussian (MNIG) distribution (O'Hagan *et al.*, 2016), the skew t distribution (Lee & McLachlan, 2014; Vrbik & McNicholas, 2012) and the variance-Gamma distribution (McNicholas *et al.*, 2013) have also been used. Cabral *et al.* (2012) explored a versatile class of models comprising finite mixtures of multivariate skew normal independent distributions. Their investigation specifically emphasised finite mixtures encompassing skew normal, skew t, skew slash and skew contaminated normal distributions. Conversely, Zehra Doru *et al.* (2021) introduced finite mixtures of multivariate skew Laplace distributions as a means to effectively capture both skewness and heavy-tailedness within heterogeneous data sets.

However, the evolution of mixture models extends beyond the choice of a distribution. Historically, mixture models have adhered to the principle of combining components of the same underlying distribution. Despite the expanding array of distribution choices, this foundational concept remains the core of multivariate finite mixture model theory. In their study, Doğru & Arslan (2016) employed univariate two-component mixtures within the context of mixture regression models. Specifically, their analysis incorporated a combination of normal and t distributions, as well as skew t and skew normal distributions. However, within unsupervised learning scenarios, the utilisation of multivariate mixtures comprising both identical and non-identical distributions remains undone. Addressing this, we present an innovative and versatile framework that overcomes traditional boundaries, enabling the mixture of diverse distributions in all possible permutations. Hence, we create a single framework that includes traditional mixture models as specific examples, prompting a fresh perspective on the extent of flexibility and utility offered by mixture modeling. Moreover, our framework tackles the difficult task of estimating parameters in mixture set up involving complex multivariate distributions. We do so by using classification EM or hard EM (Celeux & Govaert, 1992), which exploits known maximum likelihood estimates (MLEs) of the parameters of the mixing densities to efficiently model various distribution mixtures, avoiding the complications of parametric inference. We also provide proof of convergence for hard EM involving non-identical distributions. Secondly, the suggested framework is incredibly useful in addressing real-world challenges related to estimating model parameters and recognising patterns. These challenges are common in various applications. Our flexible framework enhances the precision and strength of model-based analyses.

To validate the usefulness of our suggested model, we thoroughly assess its performance using both simulated and real-world data sets. These empirical investigations provide evidence of the framework's effectiveness in capturing the complexities of intricate data patterns, addressing tasks like clustering, parameter estimation and fitting distributions to multivariate data. We explore an appropriate criterion for selecting the optimum model from various mixtures. Additionally, we offer methods to validate the chosen model through goodness-of-fit evaluations.

The remainder of the paper is organised as follows. Section 2.1 discusses briefly the multivariate distributions used in our proposed mixture models. Section 2.2 details how to formulate

mixture models using these different mixture densities. Section 3 presents the data sets used in our applications, and the results of our proposed model are discussed. Section 4 states the limitations of our study, and finally, Section 5 summarises the primary findings and examines potential avenues for further exploration.

2 Methods

In this section, we begin by giving a quick overview of the types of multivariate distributions we use in our proposed mixture models. After that, we explain how to set up mixture models using these different mixture densities.

2.1 Multivariate Distributions

Below are concise descriptions of the multivariate distributions considered in our study. We provide a summary of each distribution and highlight any special forms, where relevant.

2.1.1 Generalised hyperbolic distribution

Before delving into the generalised hyperbolic distribution, we first discuss the generalised inverse Gaussian (GIG) Distribution. It was first introduced by Good (1953). Later many other authors (see Barndorff-Nielsen & Halgreen, 1977; Blæsild, 1978; Halgreen, 1979; Jorgensen, 2012) discussed its statistical properties which laid down the foundation for the application of the GIG distribution. If $W \sim GIG(\psi, \chi, \lambda)$, the probability density function can be written in the form

$$f(w|\psi, \chi, \lambda) = \frac{(\psi/\chi)^{\lambda/2} w^{\lambda-1}}{2K_\lambda\left(\sqrt{\psi\chi}\right)} \exp\left[-\frac{\psi w + \chi/w}{2}\right], \tag{1}$$

for $w > 0$, where $\psi, \chi \in \boldsymbol{R}^+$ and K_λ is the modified Bessel function of the third kind with index λ. Gamma distribution and inverse Gaussian distribution are special forms of the GIG distribution. When $\chi = 0$ and $\lambda > 0$, the GIG density reduces to a gamma density. On the other hand, when $\lambda = -1/2$, the GIG density can be seen as a density of an inverse Gaussian distribution.

The generalised hyperbolic distribution has been discussed vividly by McNeil *et al.* (2015). If $\boldsymbol{X}$ follows a generalised hyperbolic distribution, then its probability density function is given by

$$\begin{aligned} f(\boldsymbol{x}|\boldsymbol{\vartheta}) = &\left[\frac{\chi + \delta(\boldsymbol{x}, \boldsymbol{\mu}|\boldsymbol{\Sigma})}{\psi + \boldsymbol{\gamma}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\gamma}}\right]^{\frac{\lambda - p/2}{2}} \\ &\times \frac{[\psi/\chi]^{\lambda/2} K_{\lambda - p/2}\left(\sqrt{\left[\psi + \boldsymbol{\gamma}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\gamma}\right]\left[\chi + \delta(\boldsymbol{x}, \boldsymbol{\mu}|\boldsymbol{\Sigma})\right]}\right)}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2} K_\lambda\left(\sqrt{\chi\psi}\right)\exp[(\boldsymbol{\mu} - \boldsymbol{x})'\boldsymbol{\Sigma}^{-1}\boldsymbol{\gamma}]}, \end{aligned} \tag{2}$$

where $\delta(\boldsymbol{x}, \boldsymbol{\mu}|\boldsymbol{\Sigma}) = (\boldsymbol{x} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})$ is the squared Mahalanobis distance between $\boldsymbol{x}$ and $\boldsymbol{\mu}$ and $\boldsymbol{\vartheta} = (\lambda, \chi, \psi, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma})$ denotes the parameter space.

A $p \times 1$ generalised hyperbolic random vector $\boldsymbol{X}$ can be represented as a variance-mean mixture, consisting of a generalised inverse Gaussian (GIG) random variable W and a multivariate Gaussian random vector $\boldsymbol{Z}$. A random vector $\boldsymbol{X}$ follows a multivariate generalised hyperbolic (MGH) distribution, if

$$\boldsymbol{X} = \boldsymbol{\mu} + W\boldsymbol{\gamma} + \sqrt{W}\boldsymbol{Z}, \tag{3}$$

where

(i) $\boldsymbol{Z} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}_{p \times p})$,
(ii) $\boldsymbol{\mu}, \boldsymbol{\gamma} \in \mathbb{R}^p$,
(iii) $W \geq 0$ is a scalar-valued random variable, which is independent of $\boldsymbol{Z}$ and follows $GIG(\lambda, \chi, \psi)$.

It is important to note that there are other different definitions available that lead to different parameterisations. We now discuss several limiting cases of the MGH distribution.

- **Multivariate hyperbolic distribution:** A random vector $\boldsymbol{X}$ is said to have a multivariate hyperbolic (MVH) density if it follows an MGH distribution with $\lambda = \frac{p+1}{2}$. It is to be noted that if $\lambda = 1$, its univariate margins follow one-dimensional hyperbolic distributions.
- **Normal inverse Gaussian distribution:** When a random vector $\boldsymbol{X}$ follows an MGH distribution with $\lambda = -1/2$, it is said to have a multivariate normal inverse Gaussian (MNIG) density.
- **Variance-Gamma distribution:** The variance-Gamma distribution (Barndorff-Nielsen, 1978) is also known as generalised Laplace distribution or the Bessel function distribution. A random vector $\boldsymbol{X}$ is said to have a multivariate variance-Gamma density if it follows an MGH distribution with $\lambda > 0$ and $\chi \to 0$.
- **Multivariate Student-t distribution:** The multivariate t-distribution (MVT) is also a special case of a MGH distribution. When $\psi = 0$, $\lambda < 0$ and $\boldsymbol{\gamma} = 0$, by setting the degree of freedom $\nu = -2\lambda^2$, a MGH distribution can be seen as a multivariate student t distribution.
- **Multivariate skew t distribution:** It is also possible to derive a multivariate skew t Distribution (MST) using a MGH distribution. A random vector $\boldsymbol{X}$ is said to have a multivariate skew t density if it follows a MGH distribution with $\psi = 0$.

2.1.2 *Multivariate normal distribution*

For a $p \times 1$ continuous random vector $\boldsymbol{X}$, the density of a p variate multivariate normal distribution (MVN) is given by

$$f(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right], \tag{4}$$

where $\boldsymbol{\mu}$ is a $p \times 1$ vector, $\boldsymbol{\Sigma}$ is a $p \times p$ symmetric, positive definite matrix and the support of $\boldsymbol{X}$ is $\mathrm{I}\mathrm{R}^p$.

The maximum likelihood estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are given by $\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x_i}$, $\hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_i (\boldsymbol{x_i} - \hat{\boldsymbol{\mu}})(\boldsymbol{x_i} - \hat{\boldsymbol{\mu}})^T$.

2.1.3 *Multivariate skew normal distribution*

The multivariate skew normal distribution (MSN) was first formulated by Azzalini & Capitanio (1999). Let us first consider the following stochastic representation.

Suppose, $\begin{pmatrix} \boldsymbol{Y} \\ Y_0 \end{pmatrix} \sim N_{p+1}(\boldsymbol{0}, \boldsymbol{\Omega}^*)$, $\boldsymbol{\Omega}^* = \begin{pmatrix} \boldsymbol{\Omega} & \delta_0 \\ \delta_0{}^T & 1 \end{pmatrix}$, where $\delta_0 \in \mathbb{R}^p$ and $\boldsymbol{\Omega}$ is a $p \times p$ symmetric positive definite matrix. Then, $\boldsymbol{U} = sgn(Y_0)\boldsymbol{Y}$ has a density

$$f(\boldsymbol{u}) = 2\Phi(\boldsymbol{\alpha}^T \boldsymbol{u})\phi_p(\boldsymbol{u}; \boldsymbol{0}, \boldsymbol{\Omega}), \ \boldsymbol{u} \in \mathbb{R}^p, \tag{5}$$

where $\boldsymbol{\alpha} = \boldsymbol{\Omega}^{-1}\frac{\delta_0}{(1 - \delta_0{}^T\boldsymbol{\Omega}^{-1}\delta_0)^{1/2}}$, $\Phi(\cdot)$ is the cumulative distribution function of the univariate standard normal distribution and $\phi_p(\boldsymbol{u}; \boldsymbol{0}, \boldsymbol{\Omega})$ is the p-variate normal density function with mean vector $\boldsymbol{0}$ and covariance matrix $\boldsymbol{\Omega}$. Now, $\boldsymbol{X} = \boldsymbol{U} + \boldsymbol{\mu}$, is said to have a p dimensional multivariate skew normal distribution with location $\boldsymbol{\mu}$, which is expressed as $SN_p(\boldsymbol{\mu}, \boldsymbol{\Omega}, \boldsymbol{\alpha})$.

The methodology for deriving maximum likelihood estimates for the parameters governing the multivariate generalised hyperbolic distribution and the multivariate skew normal distribution is outlined in the appendix section. Additionally, pertinent properties of the generalised inverse Gaussian distribution are also presented for comprehensive understanding.

2.2 Mixtures of Non-Identical Distributions

In this section, we introduce the mixture model with mixture densities from different distributions.

Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N$ denote a random sample of size N, where $\boldsymbol{X}_i$ is a p dimensional random vector with probability density function $f(\boldsymbol{x}_i)$ on $\mathbb{R}^p$. We can write $\boldsymbol{X} = (\boldsymbol{X}_1^T, \ldots, \boldsymbol{X}_N^T)^T$, where the superscript T denotes vector transpose and N denotes the total number of observations. An observed random sample is denoted by $\boldsymbol{x} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_N^T)^T$, where $\boldsymbol{x}_i$ is the observed value of the random vector $\boldsymbol{X}_i$.

The density of a mixture model with k components for one observation $\boldsymbol{x}_i$ is given by the mixture density

$$p(\boldsymbol{x}_i) = \sum_{j=1}^{k} \pi_j f_j(\boldsymbol{x}_i|\boldsymbol{\alpha}_j), \tag{6}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$ contains the corresponding mixture proportions with $\sum_{i=1}^{k}\pi_i = 1$ and $0 \leq \pi_i \leq 1$. $f_j(\boldsymbol{x}_i|\boldsymbol{\alpha}_j)$ is the density component of mixture j and $\boldsymbol{\alpha}_j$, $j = 1, 2, \ldots, k$, are vectors of component specific parameters for each density. Then $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)$ denotes the vector of all parameters (except $\boldsymbol{\pi}$) of the model. The log-likelihood of the model for a sample of size N is then given by

$$\log p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N|\boldsymbol{\alpha}, \boldsymbol{\pi}) = \sum_{i=1}^{N} \log\left[\sum_{j=1}^{k} \pi_j f_j(\boldsymbol{x}_i|\boldsymbol{\alpha}_j)\right]. \tag{7}$$

The parameters can be estimated using a hard EM algorithm with some modifications. For that purpose, let us introduce latent variables Z_i, which are categorical variables taking on values $1, \ldots, k$ with probabilities $\pi_1, \ldots, \pi_k$ such that $Pr(\boldsymbol{X}_i|Z_i = j) = f_j(\boldsymbol{x}_i)$, $j = 1, \ldots, k$.

Further probabilities γ_{ij} are introduced (conditional on the observed data $\boldsymbol{X} = \boldsymbol{x}$ and the parameters $\boldsymbol{\alpha}$):

$$\gamma_{ij}(\boldsymbol{x}_i) = Pr(Z_i = j|\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{\alpha}) = \frac{\pi_j f_j(\boldsymbol{x}_i|\boldsymbol{\alpha}_j)}{\sum_{j=1}^{k}\pi_j f_j(\boldsymbol{x}_i|\boldsymbol{\alpha}_j)}. \tag{8}$$

Equation (8) is the probability of cluster membership j for a data point $\boldsymbol{x}_i$. In the case of a hard EM, the following objective function is optimised.

$$\hat{\boldsymbol{\Theta}} = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \max_{z_1, \ldots, z_N} p_{\boldsymbol{\Theta}}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N, z_1, z_2, \ldots, z_N), \tag{9}$$

where $\boldsymbol{\Theta}$ denotes all parameters $(\boldsymbol{\pi}, \boldsymbol{\alpha})$. Hard EM maximises the classification likelihood. It applies a delta function approximation to the posterior probabilities $Pr(Z_i = j|\boldsymbol{X}=\boldsymbol{x}, \boldsymbol{\alpha})$, where Z_i, $i = 1, \ldots, N$ are the latent variables representing class labels. For iteration t the approximation changes the E step as follows:

$$Pr(Z_i^t = j|\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{\alpha}^t) \approx \boldsymbol{I}(j = z_i^{t*}), \tag{10}$$

where $z_i^{t*} = \underset{j}{\text{argmax}}\, \gamma_{ij}^t = \underset{j}{\text{argmax}} \log Pr(\boldsymbol{x_i}|Z_i^t = j, \boldsymbol{\alpha^t}) + \log Pr(Z_i^t = j|\boldsymbol{\alpha}^t)$ (Murphy, 2012).

In other words, hard EM includes a classification step between E and M step, where data points are assigned to their respective clusters based on the maximum posterior probability z_i^{t*}.

At the M step, we obtain the estimates of π and $\boldsymbol{\alpha}$. For a hard EM, π_j is estimated by $\frac{N_j}{N}$, where $N_j =$ number of data points in cluster j. $\boldsymbol{\alpha}_j$'s are estimated by ML estimation method considering only the assigned observations of cluster j.

In addition to the general setting of a hard EM, we include an extra step at the M step of the algorithm as a modification proposed by Pal & Heumann (2022). Instead of obtaining the MLE right away at the M step, we can do a quality check of the model. For cluster j, $j = 1, \ldots, k$ we denote

$$\boldsymbol{\alpha}_j^{new} = \begin{cases} \boldsymbol{\alpha}_j^{init} & \text{if cluster } j \text{ is empty} \\ \boldsymbol{\alpha}_j^{MLE} & \text{otherwise} \end{cases} \tag{11}$$

where $\boldsymbol{\alpha}_j^{init}$ is the initial value of the parameter $\boldsymbol{\alpha}_j$. Hard EM is well known for its greedy convergence; as a result, often, the algorithm converges with one or more clusters being empty. Hence, we would like to force the algorithm to re-iterate if one or more clusters are found to be empty at each M step. This adjustment becomes particularly crucial when we do not estimate the number of clusters and consider that the number of clusters (k) to be pre-specified. After this step, the log-likelihood is obtained using the updated parameter values. And the entire process is continued till convergence. The algorithm is displayed below.

Algorithm 1: EM Algorithm for Mixtures of Non-Identical Distributions

Initialise the model parameters, $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$. Evaluate the initial value of the log-likelihood from Equation (7);
while *log likelihood difference* $\geq \epsilon$ **do**
  Evaluate γ_{ij} from Equation (8), using the parameter values and data;
  $\pi_j^{new} = \frac{N_j}{N}$, where $N_j =$ number of data points in cluster j;
  for *i in 1 to N* **do**
    cluster z_i=$\underset{j}{\text{argmax}}\, \gamma_{ij}$;
    Assign data point x_i to cluster z_i;
  end
  for *j in 1 to k* **do**
    if *cluster j is empty* **then**
      Use initial values of $\boldsymbol{\alpha}_j$ as an update;
    else
      $\boldsymbol{\alpha}_j^{new} = \boldsymbol{\alpha}_j^{MLE}$;
    end
  end
  Re-evaluate log-likelihood using the new values of the parameters.
end

For our experiments, we used 0.0001 as the value of ϵ in Algorithm 1. The algorithm is written in Python programming language (Van Rossum & Drake, 2009) for further data analysis.

Initialisation of the EM algorithm requires some starting values of the parameters. As we are using different mixture densities, different techniques need to be used for different combinations of densities. It is well known that the EM algorithm is quite sensitive to the choice of the starting value (Melnykov & Melnykov, 2012). In this regard, hard EM is not any different. However, hard EM can offer reliable performance with the right initialisation methods. Various methods of selecting starting values are found in the literature, including random initialisation (Hipp & Bauer, 2006), iteratively constrained EM (Lubke & Muthén, 2007), KMeans clustering (Steinley & Brusco, 2011) and sum scores (Bartholomew *et al.*, 2011). However, KMeans initialisation and iteratively constrained EM with random initialisation are more favored for improved performance (Shireman *et al.*, 2017). There are certain robust versions of the EM algorithms that also take into account the number of clusters as well (Yang *et al.*, 2012). For our study, we propose to perform KMeans clustering first, and then obtain the maximum likelihood estimates of the parameters for each cluster by the techniques mentioned in the appendix section. For example, if we want to fit a mixture of a multivariate skew normal and a multivariate generalised hyperbolic distribution, at first, we perform a KMeans clustering to divide the data points into two clusters. And then we obtain the MLE of a multivariate skew normal and a multivariate generalised hyperbolic distribution using the data points available in those two respective clusters. These values of the estimated parameters serve as the initial guess of the parameters to be used in our proposed EM algorithm. The initial values of $\boldsymbol{\pi}$ are computed using the ratio of the number of cluster members obtained by the KMeans algorithm and the number of total observations.

We can formulate mixture models with different mixture densities described in the Section 2.1. Distributions that are not considered in our study can also be used if the maximum likelihood estimates of the distribution parameters can be obtained. In this way, any specific combination of mixtures becomes a special case of our proposed model. Although we can fit the mixture model for any combination of densities, for practical purposes, it can be of interest to know which combination of mixture densities yields the optimum result. We can fit all combinations of mixture densities and then choose the best option by some criterion like Akaike information criterion (AIC) (Akaike, 1974) or Bayesian information criterion (BIC) (Schwarz, 1978). However, it is enough in most cases if we use different combinations of multivariate generalised hyperbolic and multivariate skew normal distributions, as these distributions can be used to model a wide variety of data types (e.g. symmetric and asymmetric).

2.2.1 Convergence

Our proposed model uses a hard EM algorithm to estimate the parameters of the mixture model. Although hard EM uses an approximation to estimate the MLE, it maximises the classification likelihood by obtaining the MAP estimate, that is, the mode of the distribution of $Pr(Z_i = j|X, \alpha)$. Celeux & Govaert (1992) have shown that for a mixture of identical distributions at each iteration the classification likelihood increases and if ML estimates of the mixture densities are well defined, it converges to a stationary point. It can be easily extended for a mixture of non-identical distributions.

Let, $\boldsymbol{P} = (P_1, P_2, \ldots, P_k)$ be the k partitions or clusters. Then a classification maximum likelihood (CML) criterion can be defined as

$$C(\boldsymbol{P}, \boldsymbol{\pi}, \boldsymbol{\alpha}) = \sum_{j=1}^{k} \sum_{\boldsymbol{x_i} \in P_j} \log\left[\pi_j f_j(\boldsymbol{x_i}, \boldsymbol{\alpha_j})\right]. \tag{12}$$

Proposition: Any sequence $(\boldsymbol{P}^t, \boldsymbol{\pi}^t, \boldsymbol{\alpha}^t)$ for iteration t increases the CML criterion C and the sequence $C(\boldsymbol{P}^t, \boldsymbol{\pi}^t, \boldsymbol{\alpha}^t)$ converges to a stationary point. Furthermore, if the ML estimates of the parameters are well-defined, the sequence $(\boldsymbol{P}^t, \boldsymbol{\pi}^t, \boldsymbol{\alpha}^t)$ converges to a stationary position.

Proof *At first we show that CML criterion increases at each iteration step. Since,* $(\boldsymbol{\pi}^t, \boldsymbol{\alpha}_j{}^t)$ *maximises* $\sum_{\boldsymbol{x_i}} \in P_j^t \log\left[\pi_j f_j(\boldsymbol{x_i}, \boldsymbol{\alpha_j})\right]$, *we can directly write from Equation (12)*

$$C(\boldsymbol{P}^t, \boldsymbol{\pi}^{t+1}, \boldsymbol{\alpha}^{t+1}) \geq C(\boldsymbol{P}^t, \boldsymbol{\pi}^t, \boldsymbol{\alpha}^t)$$

Now, $\boldsymbol{x_i} \in P_j^{t+1}$ is equivalent to $\gamma_{ij}^{t+1} \geq \gamma_{ij'}^{t+1} \forall j' \neq j$, which implies

$$\pi_j^{t+1} f_j(\boldsymbol{x_i}, \alpha_j^{t+1}) \geq \pi_{j'}^{t+1} f_{j'}(\boldsymbol{x_i}, \alpha_{j'}^{t+1})$$

Thus, we can write

$$C(\boldsymbol{P}^{t+1}, \boldsymbol{\pi}^{t+1}, \boldsymbol{\alpha}^{t+1}) \geq C(\boldsymbol{P}^t, \boldsymbol{\pi}^{t+1}, \boldsymbol{\alpha}^{t+1})$$

Since there is a finite number of partitions of the sample into k clusters, the increasing sequence $C(\boldsymbol{P}^t, \boldsymbol{\pi}^t, \boldsymbol{\alpha}^t)$ takes a finite number of values, and as a result, it converges to a stationary value. If the ML estimates of $\boldsymbol{\pi}^t$ and $\boldsymbol{\alpha}^t$ are well defined, for a t large enough we can deduce, $\boldsymbol{\pi}^t = \boldsymbol{\pi}^{t+1}$ and $\boldsymbol{\alpha}^t = \boldsymbol{\alpha}^{t+1}$. That directly leads to $\boldsymbol{P}^t = \boldsymbol{P}^{t+1}$. Thus, for a t large enough, we can write

$$C(\boldsymbol{P}^t, \boldsymbol{\pi}^t, \boldsymbol{\alpha}^t) = C(\boldsymbol{P}^t, \boldsymbol{\pi}^{t+1}, \boldsymbol{\alpha}^{t+1}) = C(\boldsymbol{P}^{t+1}, \boldsymbol{\pi}^{t+1}, \boldsymbol{\alpha}^{t+1})$$

As mentioned above, the algorithm requires well defined ML estimates of $\boldsymbol{\alpha}_j$'s. The ML estimates of a multivariate skew normal and a multivariate generalised hyperbolic distributions are obtained using the usual EM algorithm, which again enjoys good convergence properties (Dempster *et al.*, 1977). The EM algorithm has a reliable global convergence under fairly general conditions. Unless the initial values of the parameters are not too bad, it converges nearly always to a local maximum (McLachlan & Krishnan, 2007). □

2.2.2 *Computation of standard error*

A common critique of the EM algorithm is its failure to automatically furnish an estimate of the maximum likelihood estimate's (MLE) covariance matrix, a feature provided by certain other methods like Newton-type techniques. Typically, the asymptotic covariance matrix of the ML estimates is estimated using the inverse of the observed information matrix. While the direct assessment of the observed information matrix after MLE computation seems straightforward, analytically evaluating second-order derivatives of the incomplete-data log-likelihood can prove challenging and laborious, especially contingent upon the underlying distribution. In practice, many researchers employ an approximation (Basford *et al.*, 1997) tailored for independent data to derive the Fisher information matrix. Let $l(\boldsymbol{\alpha}|\boldsymbol{x_i})$ be the complete data log-likelihood function when $\boldsymbol{x} = \boldsymbol{x_i}$. Then an approximated empirical information matrix is given by

$$I(\hat{\boldsymbol{\alpha}}, \boldsymbol{x_i}) = \sum_{i=1}^{N} s(\boldsymbol{x_i}, \hat{\boldsymbol{\alpha}}) s^T(\boldsymbol{x_i}, \hat{\boldsymbol{\alpha}}), \tag{13}$$

where $s(\boldsymbol{x_i}, \hat{\boldsymbol{\alpha}}) = \left.\frac{\partial l(\boldsymbol{\alpha}|\boldsymbol{x_i})}{\partial \boldsymbol{\alpha}}\right|_{\boldsymbol{\alpha}=\hat{\boldsymbol{\alpha}}}$ are the individual scores for $i = 1, \ldots, N$. An advantage of employing classification EM algorithm lies in its capability to directly utilise the maximum

likelihood estimates (MLE) of the parameters of the mixture component densities, along with their respective information matrices. This eliminates the need for additional computations. After model fitting, for each cluster j, the information matrix of parameter α_j can be calculated using established formulas only considering the data points assigned to cluster j. Many researchers (Cabral *et al.*, 2012; Lin, 2009) have provided formulas for the observed information matrices for the parameters in a mixture of identical distributions. These can be readily used by setting $\gamma_{ij} = 1$ and $k = 1$. Thus, the information matrix of α_j is given by $I(\hat{\boldsymbol{\alpha}_j}, \boldsymbol{x_i} \in P_j)$. Then the standard errors are obtained using the square root of the diagonal elements of the matrix $I^{-1}(\hat{\boldsymbol{\alpha}_j}, \boldsymbol{x_i} \in P_j)$. It is to be noted that this relies on an approximate estimate of the observed Fisher information matrix. It is data driven and is contingent upon the number of elements assigned to each cluster. Consequently, it serves as a means of acquiring conditional uncertainty.

The description above outlines the general approach for computing standard errors in the context of mixtures of non-identical distributions. However, in practice, obtaining the score vectors can be computationally demanding, particularly for certain distributions like the multivariate generalised hyperbolic distribution where they are not readily available. Addressing this challenge would necessitate further research to develop a more comprehensive practical solution.

2.2.3 *Identifiability*

For finite mixture models, two types of identifiability problems might occur, such as generic problem and supposedly trivial problem (see Frühwirth-Schnatter, 2006). The trivial identifiability problem refers to issues that arise as a result of empty mixture proportions π_j, mixture densities with the same parameters, and invariability of the likelihood of permutations of the mixture components (also known as label switching). By limiting the viable parameter space, these issues can be avoided. Let us constrain the whole parameter space $\boldsymbol{\Omega}$ to $\boldsymbol{\Omega}^* \subset \boldsymbol{\Omega}$ such that:

- $\pi_j > 0 \forall j = 1, \ldots, k$,
- $\boldsymbol{\alpha_j} \neq \boldsymbol{\alpha_l} \forall j \neq l; j, l \in 1, \ldots, k$,
- $\pi_j < \pi_l \forall 1 < j < l < k$.

These constraints can provide solutions for trivial identifiability issues. For the generic identifiability problem, we must consider a specific combination of mixture densities. Our proposed model considers multivariate generalised hyperbolic distributions, including five limiting cases (see Section 2.1.1), multivariate normal and multivariate skew normal distributions. Although multivariate normal and multivariate skew normal distributions are identifiable, the multivariate generalised hyperbolic distribution is not identifiable when we use the parametrisation $(\lambda, \chi, \psi, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma})$ (Browne & McNicholas, 2015). It can be easily seen that $MGH(\lambda, \chi, \psi, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma})$ and $MGH(\lambda, \chi/r, r\psi, \boldsymbol{\mu}, \boldsymbol{r\Sigma}, \boldsymbol{r\gamma})$ have the same density for any $r > 0$. This problem is solved by introducing a suitable constraint, for example, ensuring that the determinant of the dispersion matrix $\boldsymbol{\Sigma}$ is 1. However, this restriction can reduce the flexibility of model fitting. Instead, we can simply ensure that the expected value of the generalised inverse Gaussian distributed mixing variable W is 1. Thus, the parametrisation $(\lambda, \overline{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma})$ (see appendix section) is seen as a better option to deal with the identifiability issue, which makes use of the mixture properties of the MGH distribution mentioned in the appendix section. However, we must also note that this parametrisation is not valid when $\overline{\alpha}^{=0}$ and $\lambda \in [-1, 0]$ which corresponds to a Student-t distribution with non-existing variance.

2.2.4 Model diagnostics

The Kolmogorov–Smirnov (KS) goodness of fit test is a tool used to evaluate how well a model fits multivariate data. In the case of univariate data, the test compares the empirical cumulative distribution function (ECDF) of the data with the model's cumulative distribution function (CDF). The standard univariate Kolmogorov–Smirnov test statistic is given by

$$D = \sup_{x} \left| F_N(x) - \hat{F}(x) \right|,$$

where $F_N(x)$ represents the empirical cumulative distribution function (ECDF) and $\hat{F}(x)$ is the cumulative distribution function (CDF) of the fitted model (Massey Jr, 1951). However, for multivariate data, constructing the ECDF is not problematic in principle, but challenges arise when calculating the multivariate KS statistic and its null distribution. The empirical cumulative distribution function (ECDF) for multivariate data can be defined as

$$F_N(x_1, x_2, \ldots, x_p) = P(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_p \leq x_p).$$

In other words, it represents the proportion of observations that satisfy all p inequalities.

An innovative alternative approach generally used by researchers (see O'Hagan *et al.*, 2016) involves log density values to assess model fit. This method uses the model's log density values, effectively reducing the problem's complexity to univariate density values and eliminating the need for complicated numerical integration techniques.

To elaborate, for hypothesis testing, the KS statistic compares an empirical distribution function to a fully specified hypothesised distribution function. In simpler terms, the KS statistic represents the maximum absolute difference between the two cumulative distributions. In our study, we follow an alternative approach where we assume that the original data (whether real or simulated) stems from a fully specified distribution. Post model fitting, we calculate the ECDF by ordering data points based on their log density values. We consider these ECDFs as the representative of the reference or hypothesised distribution. Subsequently, employing the estimated parameter values, we generate multiple data sets and their corresponding ECDFs. Our objective is to assess whether the underlying distribution of these simulated data sets, generated by the estimated parameter values, aligns with the hypothesised distribution. To achieve this, we compare the ECDFs derived from the original data with those of the simulated data sets. A well-fitted model should exhibit ECDF patterns of the hypothesised distribution, closely matching those incorporating estimated parameter values. Thus we can visually compare the ECDFs and gauge the model's goodness of fit. This visual diagnostic helps determine how effectively the model captures the distribution of the underlying data in a multivariate setting. This alternative approach forms the basis of our simulation study and real data analysis, providing valuable insights into model performance and goodness of fit. Please note that we do not derive a test statistic and compare it with a critical value at a predetermined significance level. Instead, our approach draws inspiration from the Kolmogorov–Smirnov statistic, serving as a graphical diagnostic tool to evaluate the goodness of fit of the model.

3 Results

We apply our proposed model to both simulated and real data sets. For our simulation study, our main objective is to find out if our proposed model provides a good fit in terms of parameter estimation and goodness of fit evaluation. On the other hand, for our real data sets, we try to find out which combination of mixtures provides the best fit and whether that chosen mixture exhibits a good fit. For all the goodness of fit evaluations we use 100 simulations using the estimated parameter values. Details of the data sets and the results are described below.

3.1 Simulated Data Sets

Mixture models serve as invaluable tools for unsupervised learning, particularly in scenarios where true class labels are unknown. However, in simulation studies, we often have access to the true class labels, providing a unique opportunity to evaluate clustering performance. To conduct thorough assessments, we employ metrics such as adjusted Rand index (ARI) (Hubert & Arabie, 1985), normalised mutual information (NMI) (Ana & Jain, 2003; Kreer, 1957) and homogeneity score (HMS) (Rosenberg & Hirschberg, 2007). These metrics offer comprehensive insights into the efficacy of clustering algorithms, enabling robust evaluations of their performance. Higher scores of these metrics indicate a better performance. In our simulation study, whenever we evaluate the ARI, NMI, HMS or execution time, we conduct 20 repetitions of the experiments and obtain the mean and standard deviation. We use nine different data sets for our simulation study. A brief description of the data sets are given below along with the true data generating process (DGP) in Table 1.

For our first experiment, we draw 3 000 random samples with different proportions from a multivariate skew normal distribution, a multivariate generalised hyperbolic distribution and a multivariate normal inverse Gaussian distribution. A mixture model with the same mixing distributions (MSN, MGH and MNIG) has been fitted. We also fit four other mixture models with identical mixing distributions to compare the clustering patterns. The resulting data set with true clusters and predicted clusters is shown below.

Figure 1 shows how different mixture models find pattern in the data we simulated. We see that mixture of skew normal, multivariate generalised hyperbolic and multivariate normal inverse Gaussian distribution resembles the most with the true cluster patterns. Although other mixture models such as mixture of multivariate generalised hyperbolic and mixture of Gaussian also perform well, the difference is seen in clustering the outlier data points. We see that in this case, mixture of non identical distributions (MSN, MGH and MNIG) provides a better result. For our next experiment we simulate 3 000 random samples with different proportions from multivariate skew normal, multivariate generalised hyperbolic and multivariate skew normal distribution. Like before we compare the clustering patterns of different mixture models. The result is shown below.

From Figure 2, we can see how different mixture models work on simulated data set 2. Here, we see that the cluster pattern shown by mixture of multivariate skew normal, multivariate generalised hyperbolic and multivariate skew normal distribution comes closest to the original one. Other mixture models also performed well, however those struggled to detect the correct clusters for the data points which are further away from the centre. Nevertheless to be certain if a mixture model performs well, it is necessary to perform a goodness of fit evaluation. We

Table 1. *Description of the simulated data sets.*

Data set	*Sample size (N)*	*Number of clusters (k)*	*Dimension (p)*	*True DGP*
Simulated data set 1	3 000	3	2	MSN, MGH and MNIG
Simulated data set 2	3 000	3	2	MSN, MGH and MSN
Simulated data set 3	2 200	2	2	MSN and MGH
Simulated data set 4	2 200	2	2	MSN and MVN
Simulated data set 5	3 000	3	7	MSN, MSN and MVT
Simulated data set 6	3 000	3	7	MSN, MNIG and MNIG
Simulated data set 7	300	3	3	MSN, MGH and MGH
Simulated data set 8	300	3	3	MSN, MGH and MNIG
Simulated data set 9	400	2, 3, 4, 5, 6	2, 4, 6, 8, 10, 12	See Table 9

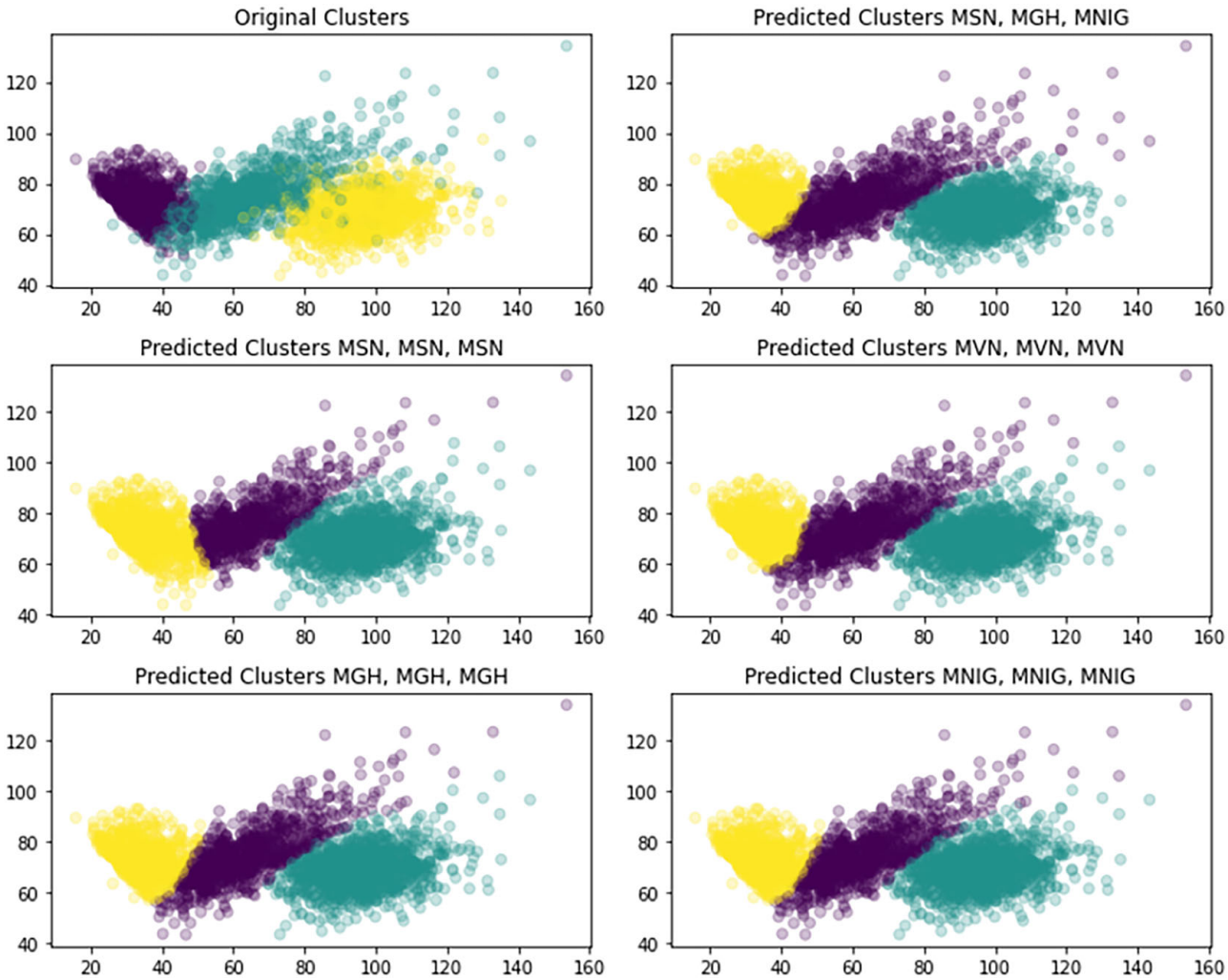

FIGURE 1. *True and predicted clusters by different mixture models for simulated data set 1 (one among 20 repeated experiments), true DGP: MSN, MGH and MNIG.*

provide the goodness of fit evaluations for mixture of MSN, MGH and MNIG and mixture of MSN, MGH and MSN for simulated data set 1 and 2 respectively.

Figures 3 and 4 show the goodness of fit evaluation of the ground truth models. We see that the ECDFs on the original data and 100 simulated data sets drawn using the estimated parameter values are completely superimposed on each other; which implies a very good fit. It is consistent with the visual resemblance of the true cluster patterns shown for the respective data sets. Moreover, as demonstrated in Table 2, the mixture models comprising MSN, MGH and MNIG distributions, as well as those combining MSN, MGH and MSN, exhibit superior performance compared to mixtures of identical distributions across simulated data sets 1 and 2. This superiority is evident across various clustering metrics, including adjusted rand index (ARI), normalised mutual information (NMI) and homogeneity score (HMS), further reinforcing the robustness of the goodness-of-fit results.

It is also of our interest to compare the estimated parameter values with the true values. For that purpose, we conduct two more experiments (simulated data sets 3 and 4) with simpler structures. For case 1, we draw random samples from a multivariate skew normal distribution and a multivariate generalised hyperbolic distribution. For case 2, we draw random samples from a multivariate skew normal distribution and a multivariate normal distribution. The detailed parameter values are given below.

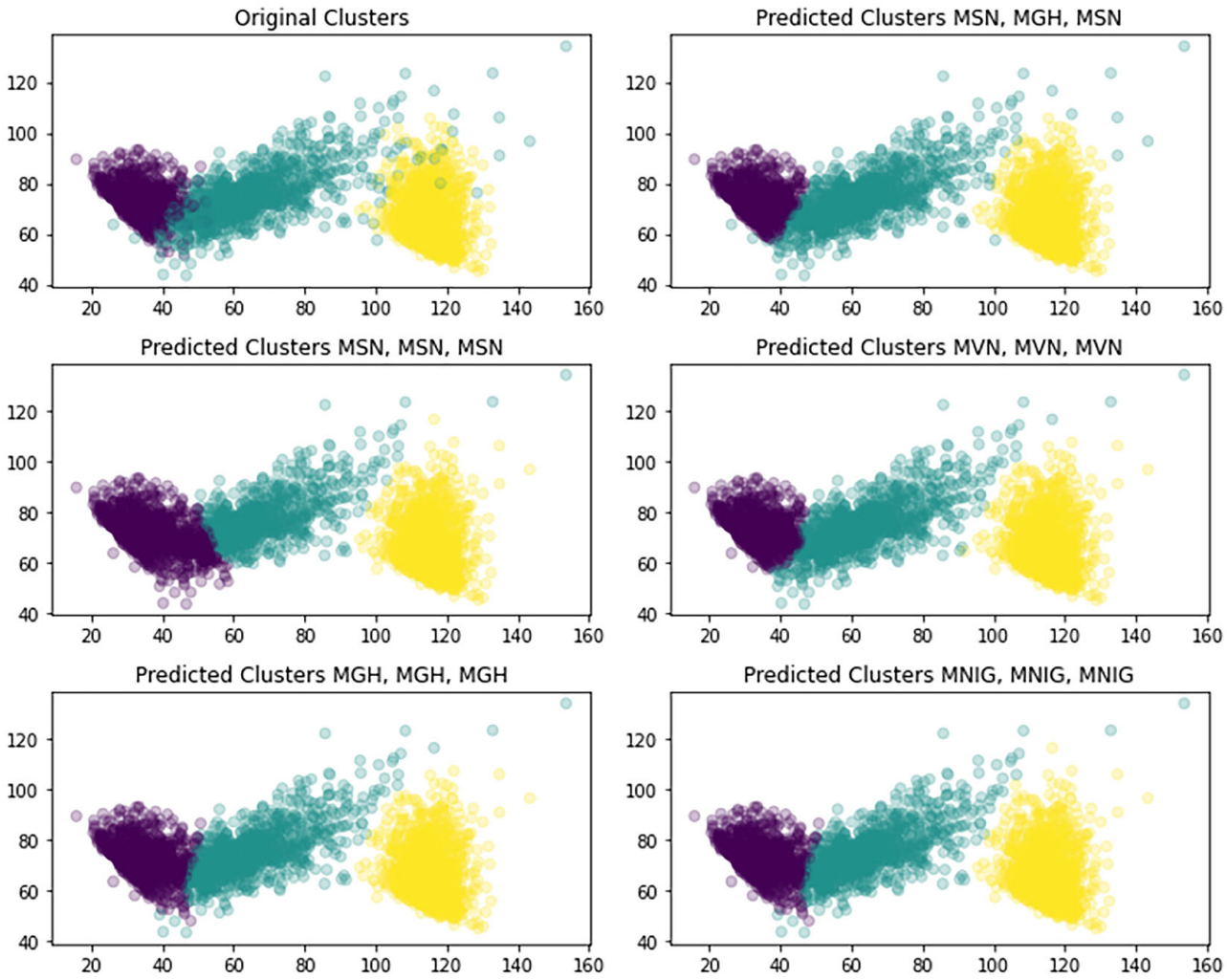

FIGURE 2. *True and predicted clusters by different mixture models for simulated data set 2 (one among 20 repeated experiments), true DGP: MSN, MGH and MSN.*

Tables 3 and 4 show the true and estimated values of the parameters from a mixture of a multivariate skew normal and a multivariate generalised hyperbolic distribution and a mixture of a multivariate skew normal and a multivariate normal distribution, respectively. We see that the estimated values are very close to the true values implying an evident good fit. However to visualise the performance goodness of fit evaluations for the above cases are conducted. The results are shown below.

From Figures 5 and 6, we see that, both for case 1 and 2, the model exhibits good fit. Additionally, we evaluate the clustering performance of the fitted models on simulated data sets 3 and 4. Figure 7 illustrates scatter plots of these data sets with the predicted clusters. Analysis presented in Table 5 reveals exemplary performance of the fitted models, as evidenced by high scores in terms of ARI, NMI and HMS metrics. The minimal overlap observed between clusters further bolsters the efficacy of the clustering algorithm.

We systematically investigate the bias and mean squared error (MSE) of our model parameters using simulated data set 3, where the sample size ranges from 220 to 2 200. Figures 8 and 9 depict the bias-MSE plots specifically for the location and skewness parameters. Notably, our analysis reveals consistently minimal bias and MSE across the range of sample sizes, indicating robust estimation performance. Furthermore, with increasing sample size, we observe a consistent trend of decreasing MSE across all situations. This phenomenon underscores the advantage

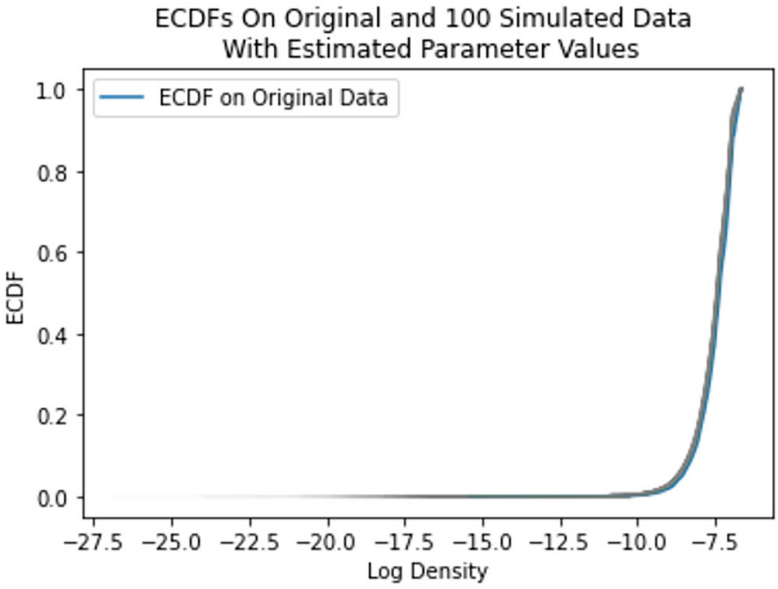

FIGURE 3. *Goodness of fit evaluation for simulated data set 1.*

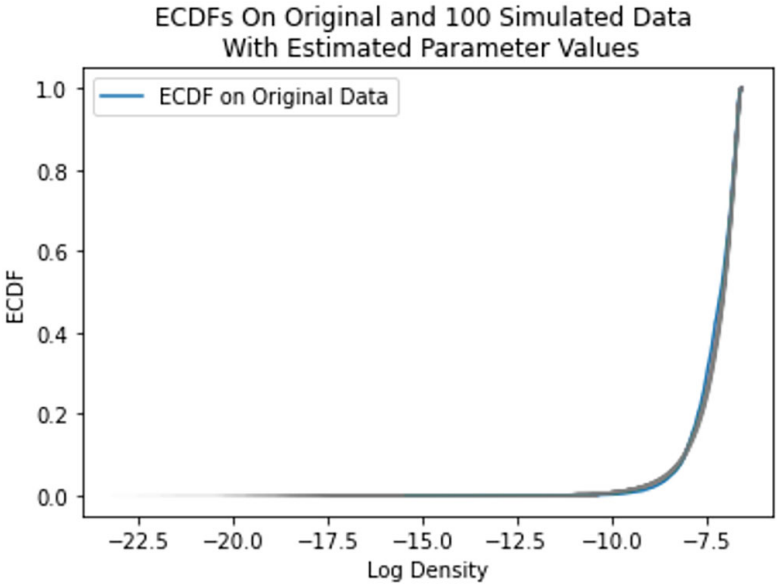

FIGURE 4. *Goodness of fit evaluation for simulated data set 2.*

of larger sample sizes in enhancing estimation precision. The asymptotic properties of the expectation-maximisation (EM) algorithm suggest that, under certain regularity conditions, parameter estimates converge to their true values as sample size increases. Our findings align with this theoretical framework, providing empirical validation of increasingly accurate parameter estimation with larger sample sizes.

Furthermore, we undertake two more simulation experiments where we fit all possible combinations of mixture models and verify the chosen model using BIC values. At first, we draw 3 000 random samples in different proportions from multivariate skew normal distribution, multivariate skew normal distribution and multivariate t distribution. For the later, we draw 3000 random samples in different proportions from multivariate skew normal distribution, multivariate normal inverse Gaussian distribution and multivariate normal inverse gaussian distribution. The top 10 performing mixture models based on BIC values are given below.

Table 2. *Clustering performance on simulated data sets 1 and 2.*

Data set	*Model*	*ARI*	*NMI*	*HMS*
Simulated data set 1	MSN, MGH and MNIG (true DGP)	***0.79 ± 0.04***	***0.75 ± 0.03***	***0.75 ± 0.03***
	MSN, MSN and MSN	0.76 ± 0.04	0.72 ± 0.03	0.72 ± 0.03
	MVN, MVN and MVN	0.77 ± 0.04	0.73 ± 0.03	0.73 ± 0.03
	MGH, MGH and MGH	0.78 ± 0.04	0.74 ± 0.03	0.74 ± 0.03
	MNIG, MNIG and MNIG	0.78 ± 0.04	0.74 ± 0.03	0.73 ± 0.03
Simulated data set 2	MSN, MGH and MSN (true DGP)	***0.82 ± 0.04***	***0.80 ± 0.03***	***0.80 ± 0.03***
	MSN, MSN and MSN	0.80 ± 0.02	0.78 ± 0.01	0.78 ± 0.01
	MVN, MVN and MVN	0.81 ± 0.02	0.78 ± 0.01	0.78 ± 0.01
	MGH, MGH and MGH	0.81 ± 0.01	0.79 ± 0.02	0.79 ± 0.02
	MNIG, MNIG and MNIG	0.81 ± 0.02	0.79 ± 0.01	0.79 ± 0.02

Table 3. *True and estimated parameter values for the mixture of a multivariate skew normal and a multivariate generalised hyperbolic distribution for simulated data set 3.*

Distribution	*Parameter*	*True value*	*Estimated value*
	$\boldsymbol{\pi}$	(0.5454, 0.4546)	(0.5454, 0.4546)
MSN	$\boldsymbol{\mu}$	(4, 7, 9)	(3.9889, 6.8482, 9.0319)
	$\boldsymbol{\Omega}$	$\begin{pmatrix} 2.3 & 0.5 & 0.1 \\ 0.5 & 3.8 & 0.3 \\ 0.1 & 0.3 & 2.5 \end{pmatrix}$	$\begin{pmatrix} 2.2136 & 0.4899 & 0.1765 \\ 0.4899 & 3.8455 & 0.5023 \\ 0.1765 & 0.5023 & 2.2842 \end{pmatrix}$
	$\boldsymbol{\delta}$	(2, 1, 7)	(2.1779, 1.3875, 6.8092)
MGH	λ	−3	−4.1121
	χ	2	6.2243
	ψ	0.5	0.0001
	$\boldsymbol{\mu}$	(10, 14, 18)	(10.1093, 14.0981, 18.0926)
	$\boldsymbol{\Sigma}$	$\begin{pmatrix} 1.0 & 0.2 & -0.1 \\ 0.2 & 0.8 & 0.4 \\ -0.1 & 0.4 & 1.2 \end{pmatrix}$	$\begin{pmatrix} 0.5709 & 0.1275 & -0.0621 \\ 0.1275 & 0.4879 & 0.2643 \\ -0.0621 & 0.2643 & 0.7811 \end{pmatrix}$
	$\boldsymbol{\gamma}$	(0.25, 0.5, 0.75)	(0.0101, 0.1885, 0.3858)

Table 4. *True and estimated parameter values for the mixture of a multivariate skew normal and a multivariate normal distribution for simulated data set 4.*

Distribution	*Parameter*	*True value*	*Estimated value*
	$\boldsymbol{\pi}$	(0.5454, 0.4546)	(0.5456, 0.4544)
MSN	$\boldsymbol{\mu}$	(4, 2, 8)	(4.0774, 2.0663, 7.9239)
	$\boldsymbol{\Omega}$	$\begin{pmatrix} 2.0 & 1.5 & 0.3 \\ 1.5 & 2.0 & 1.0 \\ 0.3 & 1.0 & 7.0 \end{pmatrix}$	$\begin{pmatrix} 1.6646 & 1.2363 & 0.2913 \\ 1.2363 & 1.7686 & 0.9626 \\ 0.2913 & 0.9626 & 6.9902 \end{pmatrix}$
	$\boldsymbol{\delta}$	(10, 13, 6)	(5.4748, 7.2886, 3.5454)
MVN	$\boldsymbol{\mu}$	(10, 12, 15)	(9.9290, 11.9344, 14.9817)
	$\boldsymbol{\Sigma}$	$\begin{pmatrix} 2.0 & 1.5 & 0.3 \\ 1.5 & 2.0 & 1.0 \\ 0.3 & 1.0 & 7.0 \end{pmatrix}$	$\begin{pmatrix} 2.1027 & 1.5855 & 0.2212 \\ 1.5855 & 4.0346 & 0.8855 \\ 0.2212 & 0.8855 & 6.5112 \end{pmatrix}$

From Table 6, we see that the correct mixture models are chosen by the BIC values. We also see many mixtures of non identical distributions among the top performers. Furthermore, we perform goodness of fit evaluation to verify if the models provide a good fit.

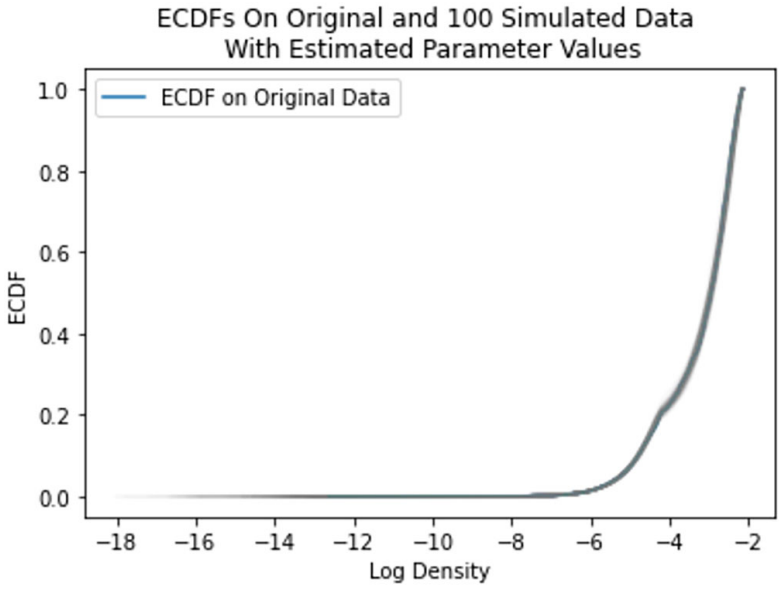

FIGURE 5. *Goodness of fit evaluation for data set 3.*

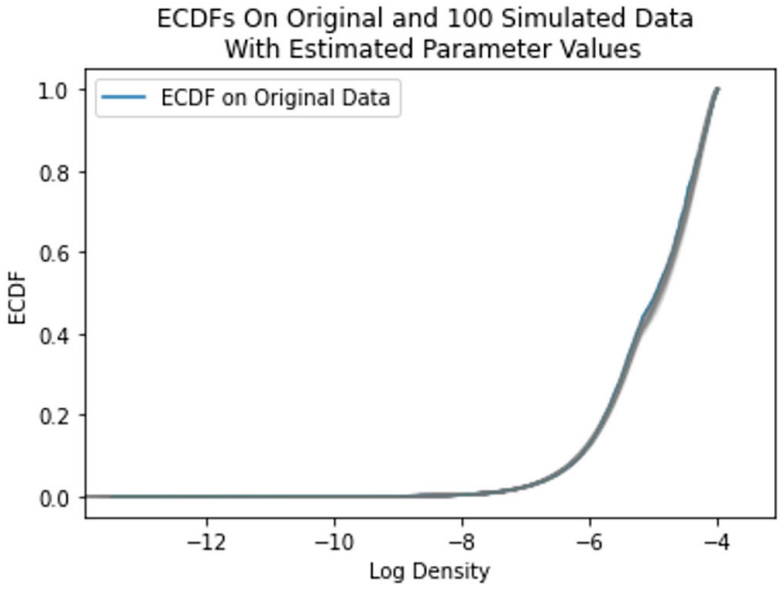

FIGURE 6. *Goodness of fit evaluation for data set 4.*

From Figures 10 and 11, we see that the chosen mixtures exhibits very good fit. First the choice of optimum mixture by BIC values and then verifying the goodness of fit provide a comprehensive strategy for fitting mixtures of non-identical or identical distributions to multivariate data. The BIC score consistently excels in model selection, demonstrating a propensity to identify the true model with high probability when it is among the considered options, especially with large sample sizes. However, a critical inquiry arises concerning the performance of BIC when the true mixture densities are absent among the fitted models. To explore this, we exhaustively fit all possible combinations of mixture densities, deliberately excluding the true ones. Table 7 showcases the top 10 performing mixture models alongside their corresponding BIC scores. Interestingly, Figures 12 and 13 reveal the goodness-of-fit outcomes of the

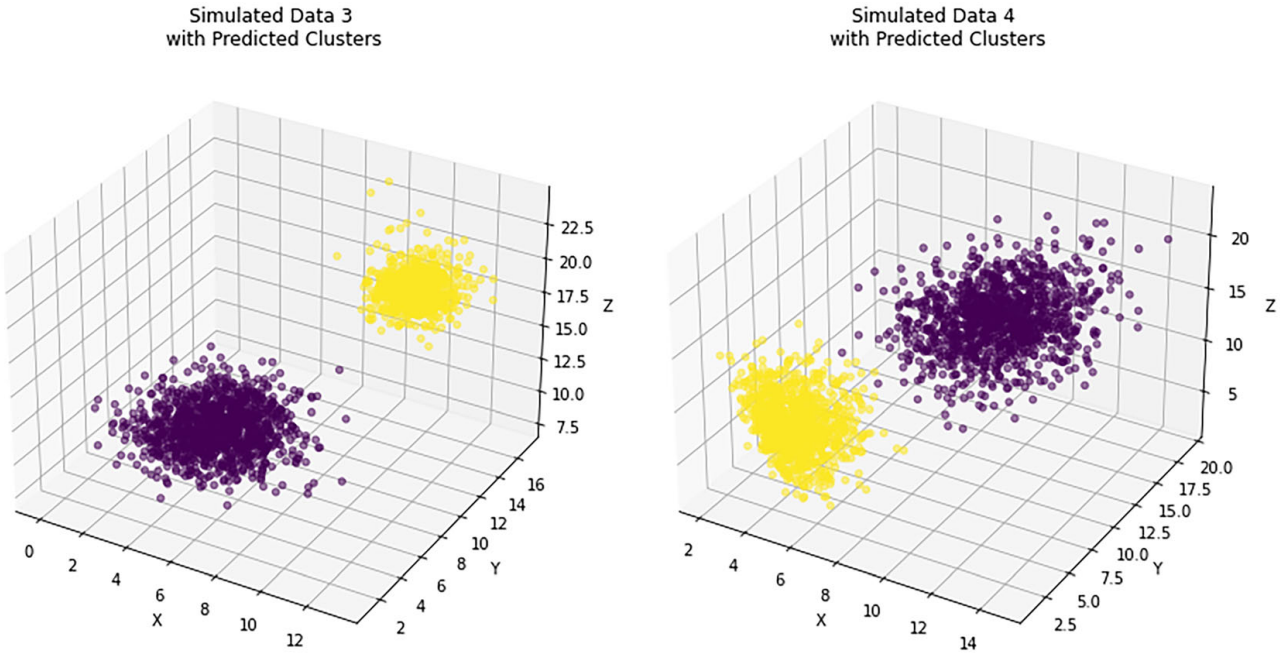

FIGURE 7. *Simulated data sets 3 and 4 with predicted clusters (one among 20 repeated experiments).*

Table 5. *Clustering performance on simulated data sets 3 and 4.*

Metric	*Simulated data set 3*	*Simulated data set 4*
ARI	0.99 ± 0.01	0.99 ± 0.01
NMI	0.99 ± 0.01	0.97 ± 0.01
HMS	0.99 ± 0.01	0.97 ± 0.01

highest-ranking models when true densities are absent, demonstrating a notably poor fit despite its low BIC scores. This highlights a crucial caveat: while BIC effectively selects the best model when true densities are present, in the absence of true densities the model chosen as optimal by BIC may yield unsatisfactory results.

Utilising the 'mixsmsn' R package (Prates *et al.*, 2013), we leverage the implementation of finite mixtures of scale mixtures of skew normal (SMSN) distributions for comparative analysis against our proposed model. To facilitate this comparison, we generate two additional data sets, namely, simulated data sets 7 and 8. The former comprises 300 random samples drawn from MSN, MGH and MGH distributions with varying proportions, while the latter consists of 300 random samples, drawn in different proportions from MSN, MGH and MNIG distributions. Figures 14 and 15 visually present these data sets, showcasing both true and predicted clusters by different models. Table 8 complements this analysis, providing metrics including ARI, NMI, HMS, BIC and execution time in seconds. Mean values along with standard deviations over 20 repetitions are provided. Additionally, we include in parentheses the percentage of times the model was selected as the best model based on the corresponding metric. Here, 'mixsmsn:MSN' refers to the mixture of multivariate skew normal, 'mixsmsn:MST' denotes the mixture of multivariate skew t, 'mixsmsn:MVN' represents the mixture of multivariate normal, and 'mixsmsn:MVT' stands for the mixture of multivariate t distributions, all available within the 'mixsmsn' package.

Our findings reveal the superior performance of our proposed model across various metrics in both data sets, demonstrating higher ARI, NMI, HMS and lower BIC scores compared with the

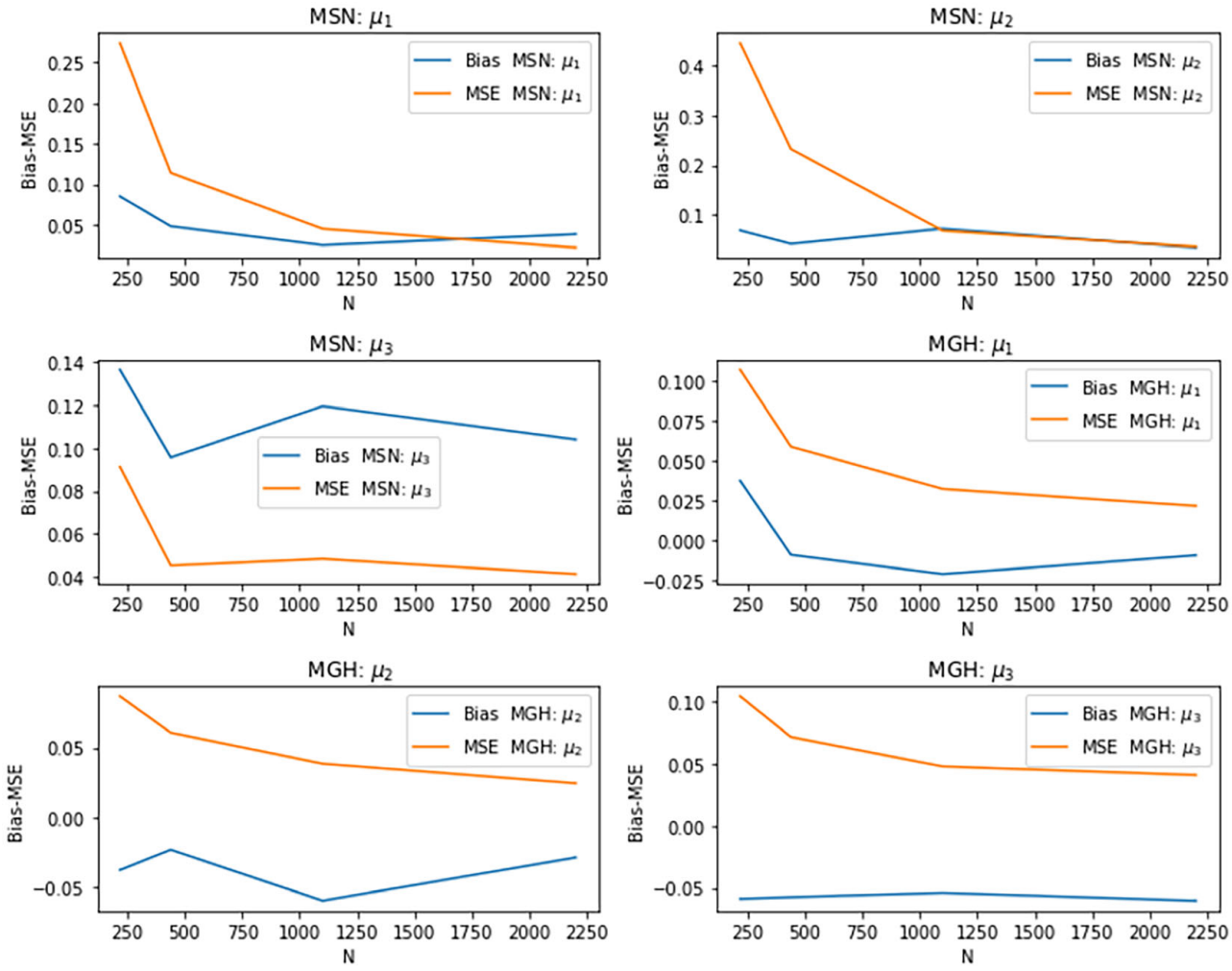

FIGURE 8. *Bias-MSE plot of location parameters on simulated data set 3 with increasing sample size over 100 repetitions.*

alternative models. Out of the 20 repetitions, mixtures of non-identical distributions are consistently chosen as the best model more frequently than any other models across all metrics except for execution time. Notably, the simplicity of 'mixsmsn:MVN' translates into faster execution times, as anticipated. However, it is evident that mixtures of identical distributions struggle to capture the true underlying patterns, particularly in data sets with significant long tails and cluster overlap. While 'mixsmsn:MSN' and 'mixsmsn:MST' exhibit commendable performance in simulated data set 8 due to their ability to model skewed data, the slight increase in execution time for mixtures of non-identical distributions is justified by their capacity to accommodate complex distributions. Ultimately, the notable superiority of non-identical distribution mixtures underscores their appeal in accurately modeling multivariate data with intricate cluster structures.

Our assessment of model performance extends to scenarios where both the number of clusters and the data dimensionality vary. To achieve this, we undertake a comprehensive simulation study (simulated data set 9), encompassing an expanding range of clusters (k) from 2 to 6 and dimensions (p) from 2 to 12. Initially, from the myriad of possible mixture combinations, we randomly select a mixture model. Subsequently, parameter values are drawn randomly from the permissible ranges. Finally, utilising these parameters and mixing densities, we generate random samples of size 400 (N) with varying proportions. Table 9 presents the randomly chosen mixture models corresponding to each k.

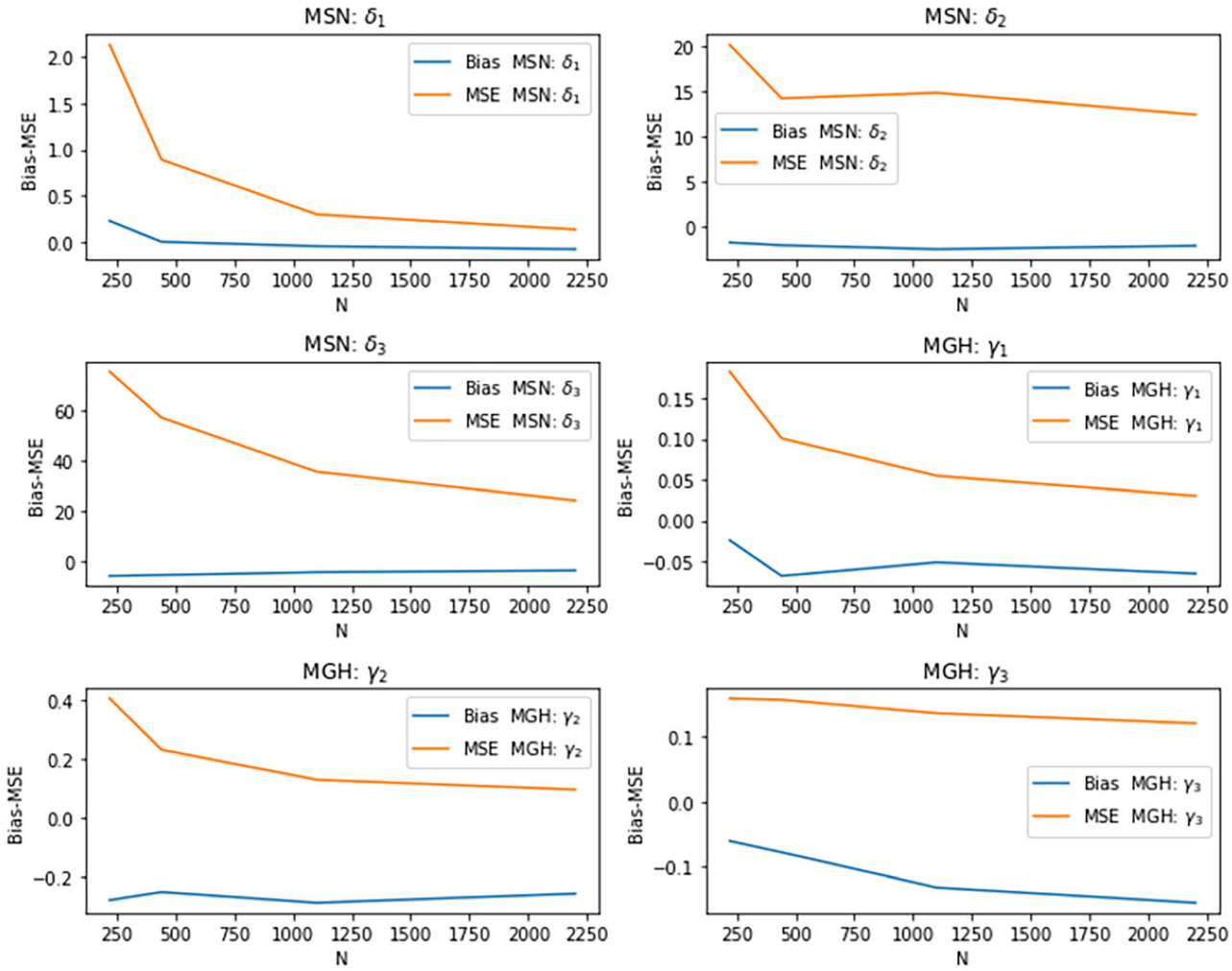

FIGURE 9. *Bias-MSE plot of skewness parameters on simulated data set 3 with increasing sample size over 100 repetitions.*

Our model evaluation is based on multiple criteria, including ARI, NMI, HMS and execution time in seconds. Figure 16 depict the mean ARI, NMI and HMS scores across increasing numbers of clusters and dimensions, while Figure 17 illustrates the corresponding execution times. The results unveil consistently high ARI, NMI and HMS scores alongside rapid execution times. Notably, no discernible pattern emerges with increasing dimensions and numbers of clusters, underscoring the robustness of our proposed model across diverse scenarios. Moreover, the clustering performance, often influenced by cluster overlap, remains exemplary.

By generating random samples from a wide spectrum of mixture densities in various combinations, covering a broad range of data scenarios, the exceptional performance of our proposed model underscores its practical applicability and flexibility in fitting finite mixture models to multivariate data.

3.2 Real Data Sets

We consider four real data sets, often used for testing mixture models. We try all possible combinations of different distributions considered in Section 2, to fit mixture models. The best mixture model is chosen based on BIC scores. Furthermore, for the chosen mixture models, a goodness of fit evaluation is performed. A brief description of the data sets is given below and in Table 10.

Table 6. *Top 10 best performing mixture models for simulated data sets 5 (true DGP: MSN, MSN and MVT) and 6 (true DGP: MSN, MNIG and MNIG).*

Simulated data set 5		*Simulated data set 6*	
Mixture model	*BIC*	*Mixture model*	*BIC*
MSN, MSN and MVT	**−37 332.9446**	**MSN, MNIG and MNIG**	**−16 207.3807**
MSN, MSN and MNIG	−37 308.6252	MSN, MGH and MNIG	−16 201.3589
MSN, MSN and MGH	−37 303.1705	MSN, MGH and MGH	−16 193.9610
MVN, MNIG and MVT	−37 066.4260	MSN, MNIG and MVT	−16 181.4745
MSN, MNIG and MVT	−37 062.7835	MSN MGH and MVT	−16 175.4527
MNIG, MNIG, and MVT	−37 057.7873	MSN, MVT and MVT	−16 152.4272
MSN, MGH and MVT	−37 055.8779	MSN, MGH and MVN	−16 149.0625
MGH, MNIG and MVT	−37 050.2629	MSN, MSN and MNIG	−16 148.4191
MGH, MGH and MVT	−37 043.3570	MSN, MSN and MGH	−16 141.0230
MVN, MVN and MVT	−37 042.8184	MSN, MSN and MVT	−16 122.5057

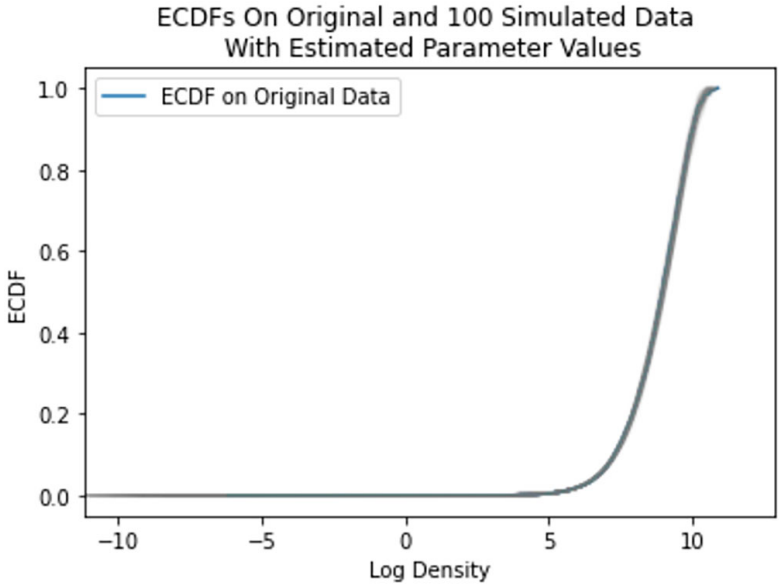

FIGURE 10. *Goodness of fit evaluation for simulated data 5.*

- **Wine Data:** These data comprise the outcomes of a chemical assessment conducted on wines grown in the same region in Italy but originating from three distinct cultivars. The evaluation measured the quantities of 13 elements present in each of the three wine types. These 13 elements are specifically: alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavanoids, non-flavanoid phenol, proanthocyanins, color intensity, Hue, OD280/OD315 of diluted wines and proline. This data set has been utilised by numerous researchers (Basu, 2004; Fischer & Poland, 2005; Zhong & Fukushima, 2007) and is accessible for download from the UCI Machine Learning Repository (Aeberhard & Forina, 1991). The primary goal of the analysis is to differentiate between the various wine types based on their constituent elements. We employ this data set for fitting different mixture models, where we have 13 dimensions, 3 clusters, and a total of 178 data points.
- **Seeds data:** The data set comprises kernels from three wheat varieties: Kama, Rosa and Canadian, with 70 kernels per variety. Utilising a non-destructive soft X-ray technique, the data set provides visual insights into the internal kernel structures, a cost-effective alternative to more advanced imaging methods. The X-ray images were captured on 13 × 18 cm X-ray

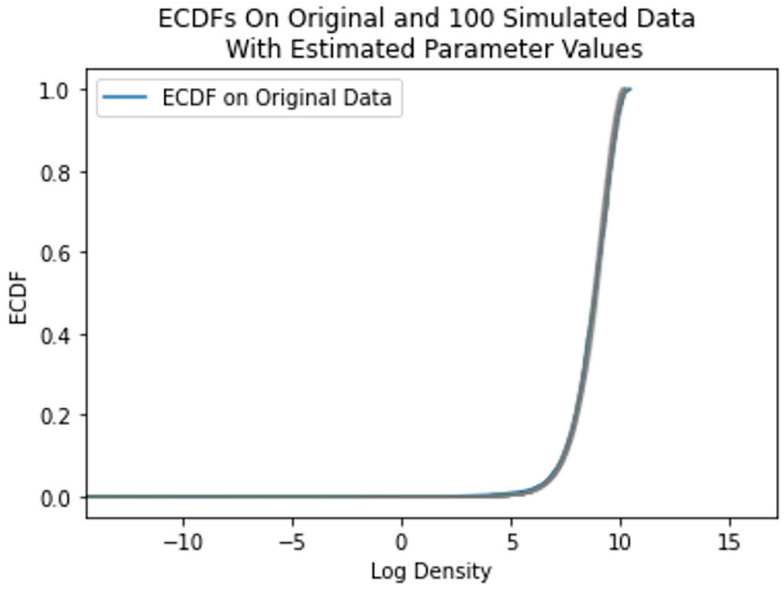

FIGURE 11. *Goodness of fit evaluation for simulated data 6.*

Table 7. *Top 10 best performing mixture models for simulated data sets 5 and 6 when true mixture densities are not present.*

Simulated data 5		*Simulated data 6*	
Mixture model	*BIC*	*Mixture model*	*BIC*
MNIG, MNIG and MNIG	**−36 898.6569**	**MVH, MVH and MVH**	**−16 044.7550**
MGH, MVN and MNIG	−36 894.7674	MGH, MVH and MVH	−16 036.7451
MGH, MNIG and MNIG	−36 889.9839	MGH, MVH and MVH	−16 032.5343
MGH, MVH and MNIG	−36 889.9011	MGH, MGH and MGH	−16 025.2476
MGH, MVH and MVH	−36 889.6472	MVH, MVH and MVT	−16 018.9459
MGH, MGH and MNIG	−36 882.7258	MGH, MVH and MVT	−16 010.9358
MGH, MGH and MVH	−36 882.4719	MGH, MGH and MVT	−16 006.7248
MGH, MGH and MGH	−36 875.8112	MVN, MVT and MVT	−16 004.9640
MGH, MVN and MVN	−36 870.0378	MVH, MVH and MVN	−15 992.7282
MVH, MVN and MNIG	−36 865.4834	MVH, MVT and MVT	−15 991.7078

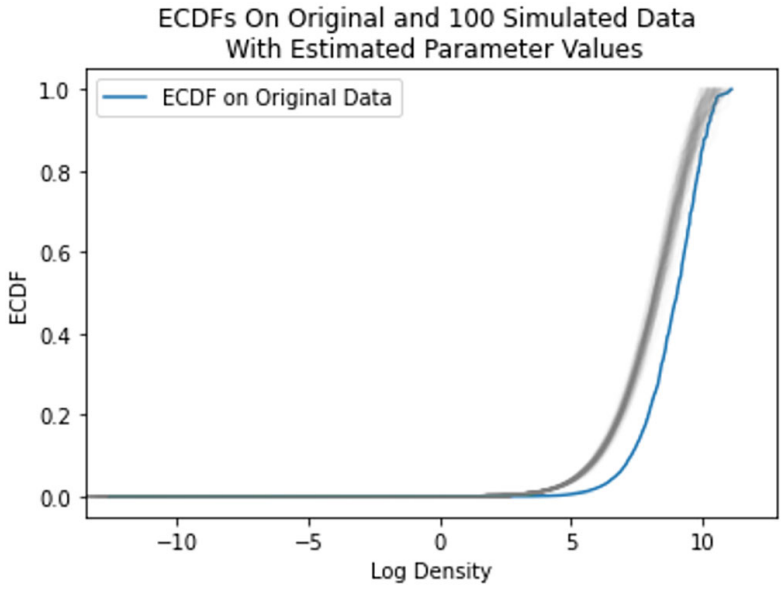

FIGURE 12. *Goodness of fit evaluation for simulated data 5 for model MNIG, MNIG and MNIG.*

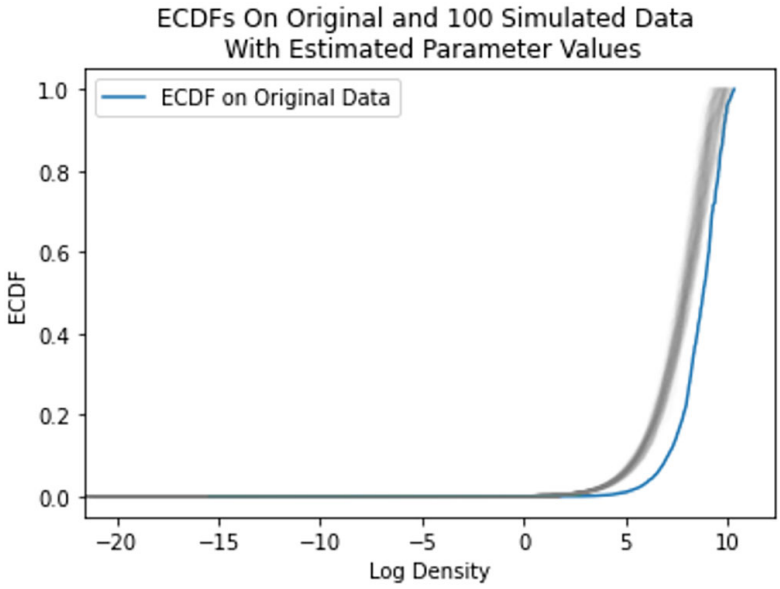

FIGURE 13. *Goodness of fit evaluation for simulated data 6 for model MVH, MVH and MVH.*

KODAK plates. The study sourced wheat grains from experimental fields at the Institute of Agrophysics of the Polish Academy of Sciences in Lublin. The data set's key features include seven continuous geometric parameters of the wheat kernels, facilitating analysis and distinction among the wheat varieties: area, perimeter, compactness, length of the kernel, width of the kernel, asymmetry coefficient and length of kernel groove. These parameters enable precise characterisation of the wheat kernels and differentiation across the three wheat varieties, as stated by many researchers previously (see Fillbrunn & Berthold, 2015). We use this data set for fitting mixture models, where we have seven dimensions, three clusters and a total of 210 data points. This data set can also be accessed via UCI Machine Learning Repository (Charytanowicz & Lukasik, 2012).

- **Rice data:** The rice data set chosen for this study focuses on two specific rice species, Osmancik and Cammeo, both of which are certified varieties cultivated in Turkey. Osmancik, which has been widely planted since 1997, is characterised by its broad, elongated, glossy and muted appearance. On the other hand, Cammeo, introduced in 2014, shares similar features, being broad, elongated, glossy and muted in appearance. In total, 3 810 images of rice grains were captured for these two species, subsequently processed and subjected to feature analysis. For each rice grain, seven morphological characteristics were extracted as part of the study. This data set is available for download from UCI Machine Learning Repository (mis, 2019). Previously other researchers (Cinar & Koklu, 2019) have used this data set for classification tasks. We use this data set for fitting mixture models with different identical and non-identical mixture components. Here, we have 7 dimensions, 2 clusters and 3 810 data points.
- **Olive Data:** The Olive Oil data set (Forina *et al.*, 1983) comprises seven independent variables, representing the levels of fatty acids present in the oils, and three categories denoting different regions in Italy. The primary objective of this analysis is to establish a robust criterion for accurately distinguishing oils originating from the three distinct geographical regions. This problem holds practical significance due to variations in the perceived quality of oils from specific regions, leading to instances of fraudulent claims regarding their origin by unscrupulous suppliers. Furthermore, the composition of these oils is a subject of independent research, given their high nutritional value and the varying benefits associated with different

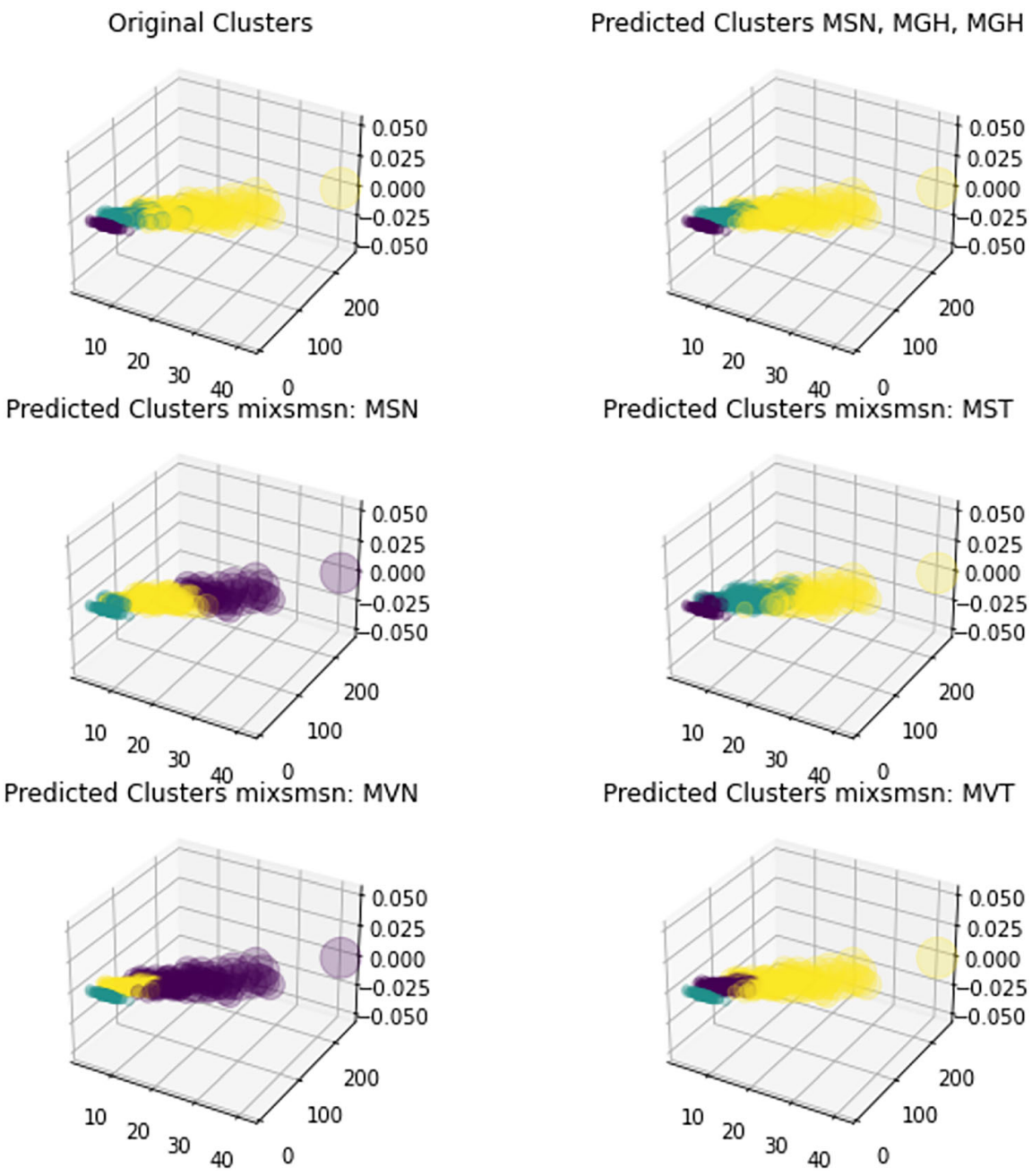

FIGURE 14. *True and predicted clusters by different mixture models for simulated data set 7 (one among 20 repeated experiments).*

constituent fatty acids. Additionally, it is crucial to recognise that fatty acid concentrations are influenced by regional climates, which carries implications for informed decisions regarding the cultivation of specific olive varieties in particular geographic areas. This data set can be accessed via a book originally written for data analysis using R and ggobi (Cook *et al.*, 2007). We have used this data set for fitting mixture models. Here, we have 3 clusters, 7 dimensions and 572 data points.

The top 10 best-performing mixture models based on BIC scores are given below for all the data sets.

From Table 11, we see the best-performing model is a mixture of multivariate skew normal, multivariate skew normal and multivariate normal inverse Gaussian for wine data. On the other hand for the seeds data, the best model is a mixture of multivariate skew normal, multivariate

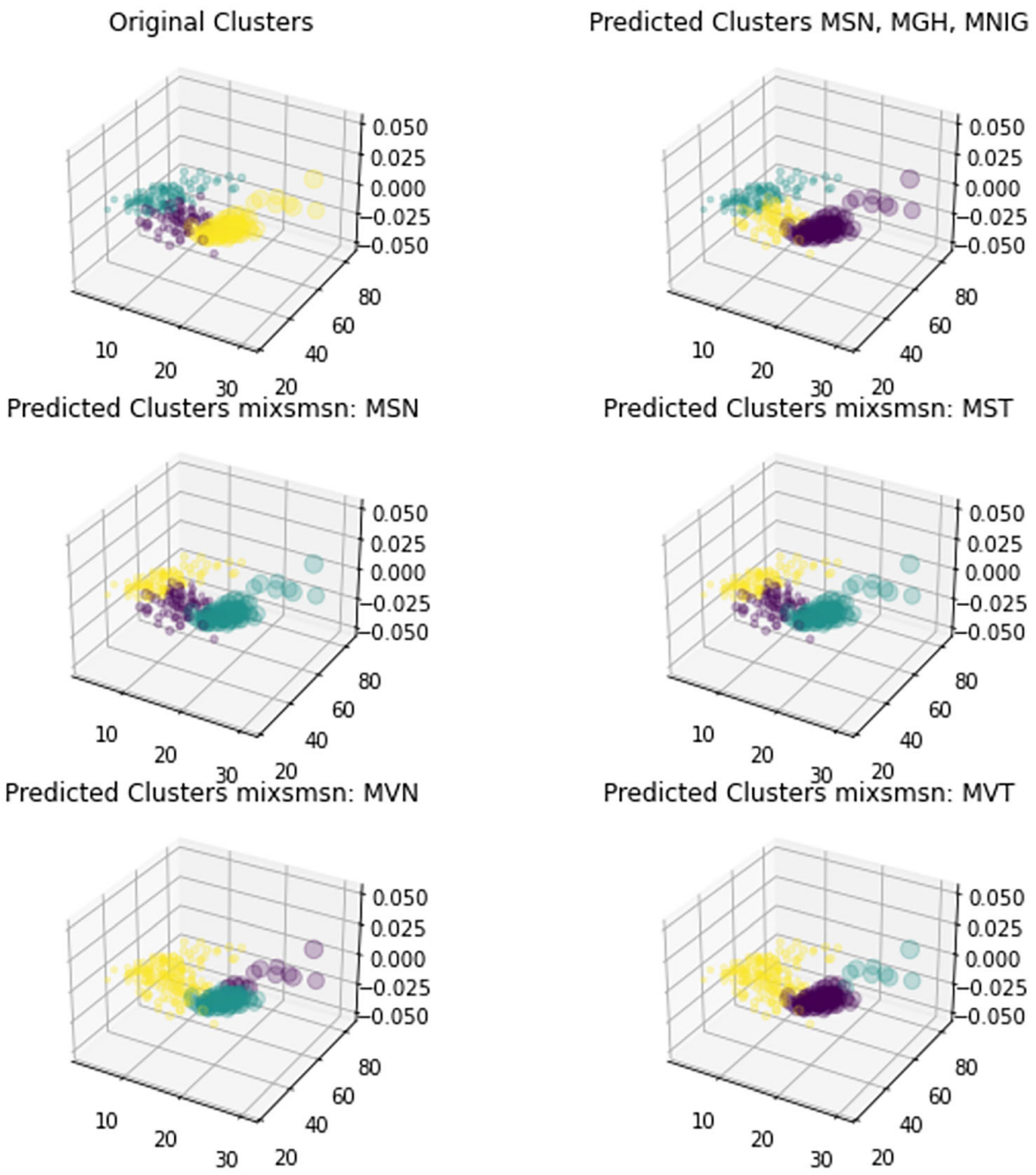

FIGURE 15. *True and predicted clusters by different mixture models for simulated data set 8 (one among 20 repeated experiments).*

skew normal and multivariate t. It is seen in Table 12 that for rice data mixture of multivariate generalised hyperbolic and multivariate normal inverse gaussian distribution is chosen as the best mixture model. Furthermore, for Olive data, mixture of multivariate skew normal, multivariate generalised hyperbolic and multivariate t distribution is found to be optimum. We also find many mixtures with non-identical mixture components in the tables. It shows that mixture models with different densities give an wide range of options for model selection. Fitting data with this variety of models provides more flexibility and can yield better results.

Figures 18, 19, 20 and 21 illustrate that ECDFs on original data and 100 simulated data with estimated parameter values are superimposed on each other, which implies a very good fit. It further supports the choice of optimum mixture models using BIC scores.

Table 8. *Clustering performance on simulated data sets 7 (true DGP: MSN, MGH and MGH) and 8 (true DGP: MSN, MGH and MNIG).*

Data set	*Model*	*ARI*	*NMI*	*HMS*	*Time*	*BIC*
Simulated data set 7	MSN, MGH and MGH	$\mathbf{0.79 \pm 0.16 (0.85)}$	$\mathbf{0.81 \pm 0.12 (0.8)}$	$\mathbf{0.81 \pm 0.13 (0.8)}$	$8.28 \pm 4.75 (0.0)$	$\mathbf{5594.23 \pm 113.71 (1.0)}$
	mixsmsn:MSN	$0.41 \pm 0.03 (0.05)$	$0.52 \pm 0.03 (0.1)$	$0.49 \pm 0.03 (0.05)$	$1.57 \pm 0.28 (0.0)$	$6182.56 \pm 42.62 (0.0)$
	mixsmsn:MST	$0.41 \pm 0.04 (0.0)$	$0.52 \pm 0.04 (0.0)$	$0.48 \pm 0.03 (0.0)$	$5.83 \pm 1.05 (0.0)$	$6191.56 \pm 45.55 (0.0)$
	mixsmsn:MVN	$0.51 \pm 0.19 (0.05)$	$0.60 \pm 0.14 (0.1)$	$0.58 \pm 0.15 (0.15)$	$\mathbf{0.3 \pm 0.11 (1.0)}$	$5915.1 \pm 165.78 (0.0)$
	mixsmsn:MVT	$0.49 \pm 0.17 (0.05)$	$0.59 \pm 0.13 (0.0)$	$0.57 \pm 0.14 (0.0)$	$1.84 \pm 0.58 (0.0)$	$5940.35 \pm 158.79 (0.0)$
Simulated data set 8	MSN, MGH and MNIG	$\mathbf{0.78 \pm 0.21 (0.5)}$	$\mathbf{0.84 \pm 0.13 (0.55)}$	$\mathbf{0.80 \pm 0.16 (0.55)}$	$4.39 \pm 2.74 (0.0)$	$\mathbf{5517.61 \pm 61.98 (0.95)}$
	mixsmsn:MSN	$0.72 \pm 0.17 (0.1)$	$0.77 \pm 0.12 (0.1)$	$0.75 \pm 0.14 (0.1)$	$0.64 \pm 0.19 (0.0)$	$5680.07 \pm 71.47 (0.0)$
	mixsmsn:MST	$0.76 \pm 0.18 (0.2)$	$0.80 \pm 0.12 (0.15)$	$0.77 \pm 0.15 (0.15)$	$1.74 \pm 0.52 (0.0)$	$5660.48 \pm 79.22 (0.05)$
	mixsmsn:MVN	$0.69 \pm 0.16 (0.15)$	$0.77 \pm 0.09 (0.15)$	$0.73 \pm 0.12 (0.15)$	$\mathbf{0.08 \pm 0.04 (1.0)}$	$5673.49 \pm 66.57 (0.0)$
	mixsmsn:MVT	$0.65 \pm 0.13 (0.05)$	$0.73 \pm 0.06 (0.05)$	$0.69 \pm 0.09 (0.05)$	$0.45 \pm 0.21 (0.0)$	$5666.06 \pm 54.24 (0.0)$

Table 9. *Fitted models with increasing number of clusters.*

Number of clusters	*Fitted models*
2	MSN and MVT
3	MSN, MVT and MVN
4	MSN, MGH, MNIG and MVT
5	MVT, MVN, MVH, MNIG and MVSN
6	MVN, MNIG, MVT, MVH, MSN and MSN

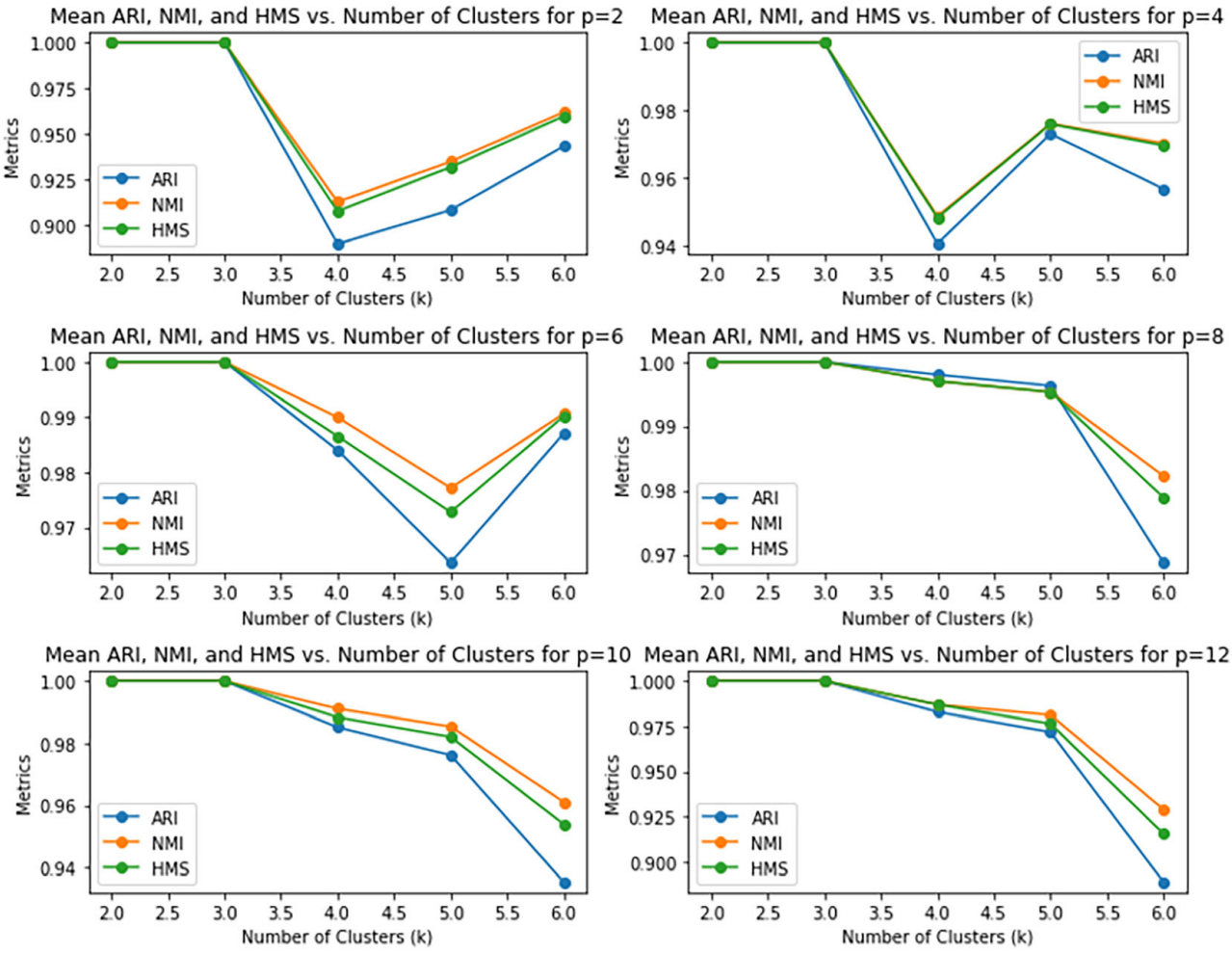

FIGURE 16. *Mean ARI, NMI and HMS for data set 9 when number of clusters and dimensions both changes.*

4 Limitations

While our proposed framework offers numerous advantages, it is not without limitations. Initially, our approach relied on a predetermined number of clusters, yet practical scenarios often demand estimation of this parameter. One effective strategy involves employing algorithms such as KMeans or Gaussian mixture models (GMM) across varying cluster numbers and selecting the model with the lowest Bayesian information criterion (BIC) value.

The estimation of parameters in the M step poses a notable challenge due to its lack of analytical tractability. Absence of closed-form solutions necessitates multiple iterations of EM or MCECM algorithms at each M step, resulting in computationally intensive and

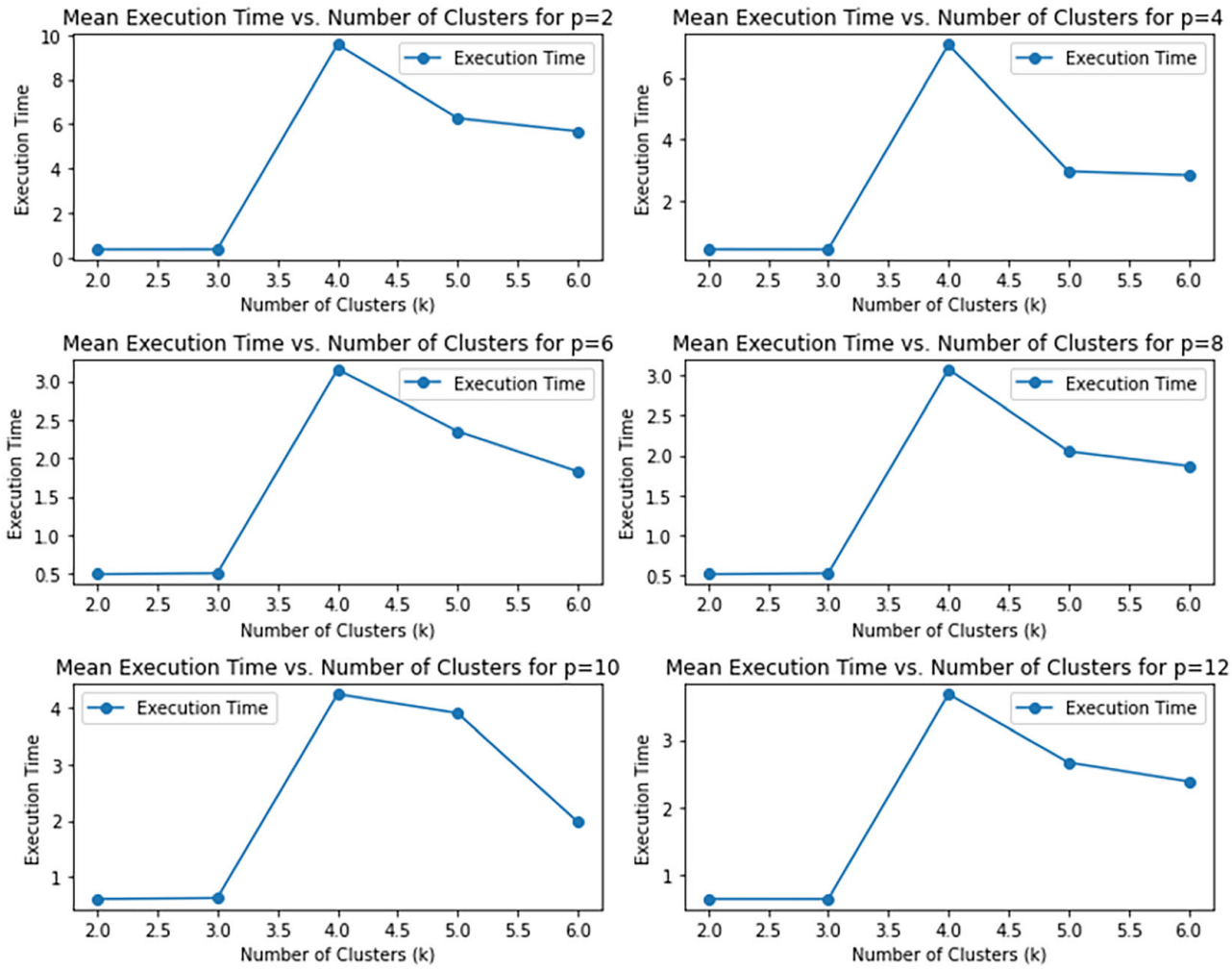

FIGURE 17. *Mean execution time for data set 9 when number of clusters and dimensions both changes.*

Table 10. *Description of the real data sets.*

Data set	*Sample size (N)*	*Number of clusters (k)*	*Dimension (p)*
Wine data	178	3	13
Seeds data	210	3	7
Rice data	3 810	2	7
Olive data	572	3	7

Table 11. *Top 10 best performing mixture models for wine and seeds data.*

Wine data		*Seeds data*	
Mixture model	*BIC*	*Mixture model*	*BIC*
MSN, MSN and MNIG	**5 931.4065**	**MSN, MSN and MVT**	**−2 523.2987**
MSN, MSN and MGH	5 938.2819	MSN, MVN and MVT	−2 511.5568
MSN, MSN and MSN	5 942.9507	MSN, MSN and MSN	−2 510.2317
MSN, MSN and MVN	5 960.8441	MSN, MNIG and MVT	−2 507.2438
MSN, MSN and MNIG	5 965.6589	MSN MSN and MNIG	−2 505.4540
MSN, MSN and MVT	5 969.2698	MSN, MVT and MVT	−2 502.1123
MSN, MVN and MVT	5 970.1388	MVN, MVT and MVT	−2 500.8528
MGH, MGH and MGH	5 971.6882	MSN, MSN and MGH	−2 498.0489
MSN, MVN and MVN	5 976.5649	MVN, MNIG and MVT	−2 496.5398
MSN, MNIG and MNIG	5 979.9558	MSN, MVN and MVN	−2 496.3614

Table 12. *Top 10 best performing mixture models for rice and olive data.*

Rice data		*Olive data*	
Mixture model	*BIC*	*Mixture model*	*BIC*
MGH and MNIG	**107 964.0822**	**MSN, MGH and MVT**	**38 285.1227**
MGH and MGH	107 975.2304	MSN, MSN and MVT	38 288.9038
MNIG and MNIG	107 975.3681	MSN, MNIG and MNIG	38 326.9283
MVN and MNIG	118 066.8532	MNIG, MNIG and MNIG	38 329.3443
MGH and MVN	118 086.6512	MGH, MNIG and MNIG	38 339.9092
MSN and MSN	144 581.9866	MGH, MNIG and MVT	38 346.1153
MSN and MVN	147 000.1895	MSN, MNIG and MVT	38 349.2046
MSN and MVT	148 417.9704	MNIG, MNIG and MVT	38 354.3180
MVN and MVN	149 446.4086	MSN, MGH and MGH	38 360.4696
MVN and MVT	152 399.1557	MSN, MSN and MSN	38 362.3696

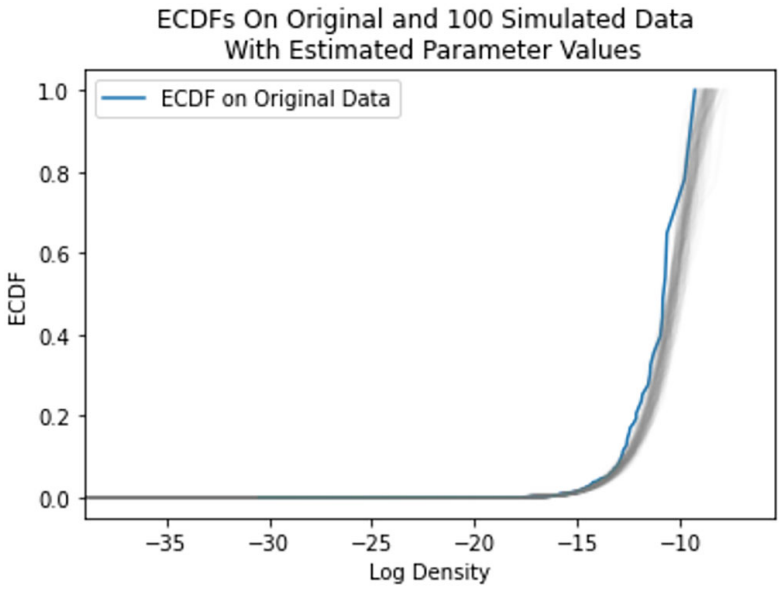

FIGURE 18. *Goodness of fit evaluation for wine data.*

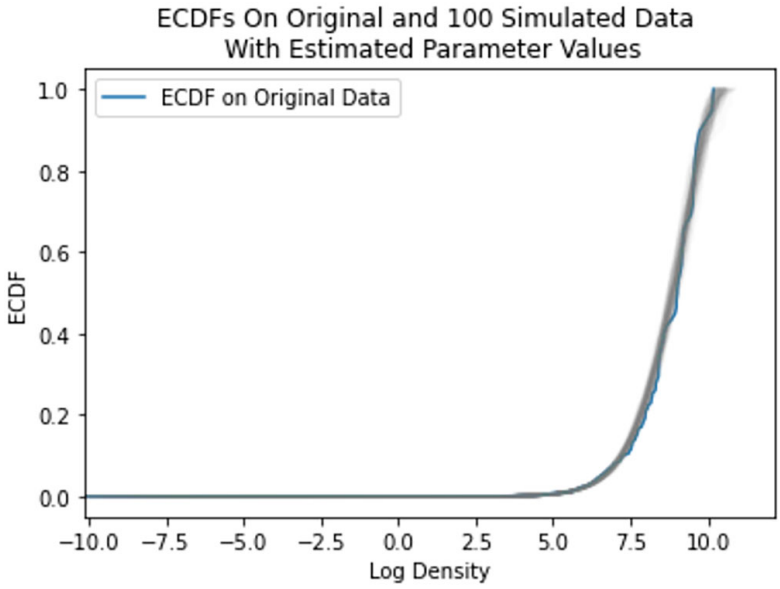

FIGURE 19. *Goodness of fit evaluation for seeds data.*

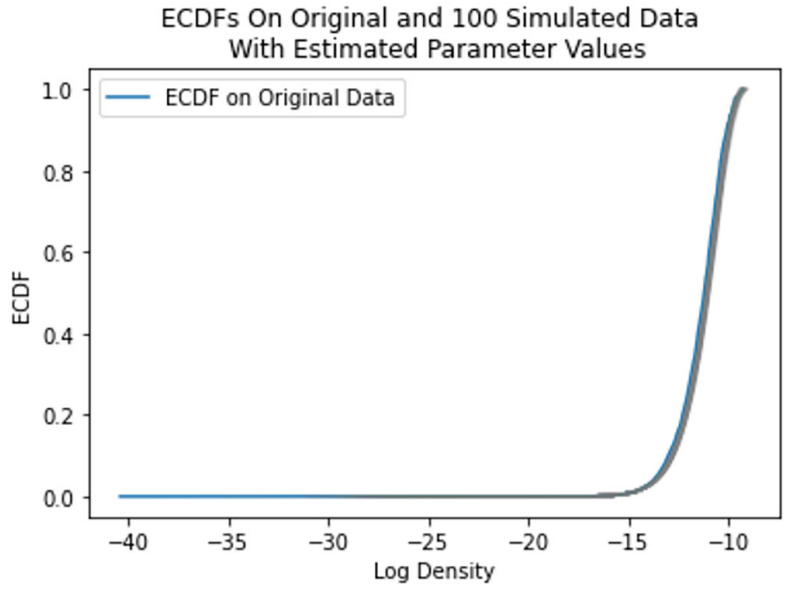

FIGURE 20. *Goodness of fit evaluation for rice data.*

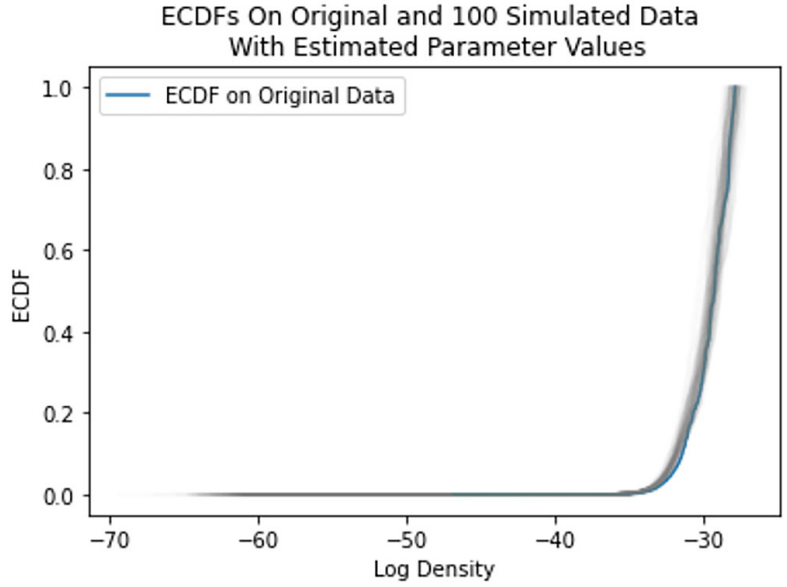

FIGURE 21. *Goodness of fit evaluation for Olive data.*

time-consuming processes. Moreover, as the number of clusters increases, the myriad of potential combinations of mixing distributions expands. This calls for exhaustive exploration of numerous mixture models to identify the optimal fit for the data set.

In addition to these challenges, clustering algorithms commonly encounter certain limitations. Computational errors may arise when no data points are assigned to a cluster or when only one data point is assigned to a cluster. Furthermore, poor ML estimates may result when clusters contain very few data points. Hence, enhancement in estimation quality is observed with increasing sample size. Furthermore, it is worth noting that our current framework has not been tailored for high-dimensional data sets. As we endeavor to extend our framework to accommodate such data, we must address the escalating complexity inherent in high-dimensional spaces. The number of parameters for the covariance matrix increases

exponentially with the dimensionality of the data (known as the curse of dimensionality), posing significant challenges. To overcome this issue, further research may be required to develop penalisation techniques or other strategies capable of mitigating the adverse effects of high-dimensional data on our framework's performance.

In our study, we have outlined a general method for computing standard errors within the framework of mixtures of non-identical distributions. However, obtaining the score vectors can pose computational challenges, especially for distributions like the multivariate generalised hyperbolic distribution where they are not readily accessible. In our simulation study and real data applications, we have not calculated the standard errors. We need to carefully assess the suitability of the simple score approximation approach, which requires further research.

5 Conclusion

In this study, we presented a flexible method for fitting mixture models that surpasses the traditional limits of using identical underlying distributions for each mixture component. Our proposed methodology provided an innovative framework to model mixtures with any combination of different distributions, thereby significantly enhancing the flexibility and applicability of mixture models. By employing the expectation-maximisation (EM) algorithm, we demonstrated how this framework could seamlessly incorporate diverse distributions into the mixture model construction process. We have also provided proofs of convergence for the hard EM algorithm involving mixtures of non-identical distributions. The model inherits promising convergence properties, ensuring its effectiveness in real-world applications.

An essential aspect of our study involved discussing the issues of identifiability and model diagnostics related to these flexible mixture models. We explored the utility of goodness-of-fit evaluations to validate the chosen combinations of mixture models, revealing a compelling level of agreement between our model and the observed data. This validation process reinforced the robustness and credibility of our chosen mixture distributions.

Through an extensive simulation study and applications on real data, we noticed a noteworthy pattern. The mixtures of non-identical distributions consistently outperformed mixtures of the identical distribution, as reflected in the Bayesian information criterion (BIC) scores. This observation underscored the significance of our framework in providing a broader spectrum of options for modeling multivariate data with intricate cluster structures. It accommodates mixtures of both identical and non-identical distributions which makes any conventional mixture model a special case of our framework. The best mixture model can be chosen using BIC values from an wide range of mixture models such that it provides the optimum fit. Mixtures of identical distribution can also provide good fit in some cases, which is covered by our framework. Furthermore, we successfully showcased the efficacy of our model in parameter estimation through simulated data experiments. The estimated parameter values demonstrated remarkable proximity to the true values, illustrating the model's precision and reliability.

In conclusion, our proposed approach has shown great potential for analyzing multivariate data with inherent cluster patterns. By offering a flexible framework to construct mixture models with varying distributions, our model aids researchers and practitioners with a powerful tool to unravel complex data patterns and structures.

Acknowledgements

The authors would like to extend their heartfelt gratitude to the editor-in-chief, the editor and two anonymous reviewers for their insightful comments and invaluable suggestions, which have greatly enhanced the quality of this study.

References

mis 2019. Rice (Cammeo and Osmancik). UCI Machine Learning Repository, https://doi.org/10.24432/C5MW4Z

Abe, T., Fujisawa, H., Kawashima, T. & Ley, C. (2021). EM algorithm using overparameterization for the multivariate skew-normal distribution. *Econ. Stat.*, **19**, 151–168.

Aeberhard, S. & Forina, M. 1991. Wine. UCI Machine Learning Repository, https://doi.org/10.24432/C5PC7J

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Trans. Autom. Control*, **19**(6), 716–723.

Ana, L.N.F. & Jain, A.K. (2003). Robust data clustering. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings*. Vol. **2**, pp. II–II, IEEE.

Azzalini, A. & Capitanio, A. (1999). Statistical applications of the multivariate skew normal distribution. *J. Royal Stat. Soc.: Ser. B (Stat. Methodol.)*, **61**(3), 579–602.

Balaban, M., Moshiri, N., Mai, U., Jia, X. & Mirarab, S. (2019). Treecluster: Clustering biological sequences using phylogenetic trees. *PloS One*, **14**(8), e0221068.

Barndorff-Nielsen, O. (1978). Hyperbolic distributions and distributions on hyperbolae. *Scandinavian J. Stat.*, 151–157.

Barndorff-Nielsen, O. & Halgreen, C. (1977). Infinite divisibility of the hyperbolic and generalized inverse Gaussian distributions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, **38**(4), 309–311.

Bartholomew, D.J., Knott, M. & Moustaki, I. (2011). *Latent variable models and factor analysis: A unified approach*. John Wiley & Sons.

Basford, K.E., Greenway, D.R., McLachlan, G.J. & Peel, D. (1997). Standard errors of fitted component means of normal mixtures. *Comput. Stat.*, **12**(1), 1–18.

Basu, S. (2004). Semi-supervised clustering with limited background knowledge. In *Aaai*, pp. 979–980.

Blæsild, P. (1978). *The shape of the generalized inverse Gaussian and hyperbolic distributions*. Department of Theoretical Statistics, Inst., Univ.

Breymann, W. & Lüthi, D. 2013. ghyp: A package on generalized hyperbolic distributions. Manual for R Package ghyp.

Browne, R.P. & McNicholas, P.D. (2015). A mixture of generalized hyperbolic distributions. *Canad. J. Stat.*, **43**(2), 176–198.

Cabral, C.R.B., Lachos, V.H. & Prates, M.O. (2012). Multivariate mixture modeling using skew-normal independent distributions. *Comput. Stat. Data Anal.*, **56**(1), 126–142.

Celeux, G. & Govaert, G. (1992). A classification em algorithm for clustering and two stochastic versions. *Comput. Stat. Data Anal.*, **14**(3), 315–332.

Charytanowicz, N.J.K.P.K.P. & Lukasik, S. 2012. seeds. UCI Machine Learning Repository, https://doi.org/10.24432/C5H30K

Cinar, I. & Koklu, M. (2019). Classification of rice varieties using artificial intelligence methods. *Int. J. Intell. Syst. Appl. Eng.*, **7**(3), 188–194.

Cook, D., Swayne, D.F. & Buja, A. (2007). *Interactive and dynamic graphics for data analysis: with r and ggobi*, Vol. **1**. Springer.

Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the em algorithm. *J. Royal Stat. Soc.: Ser. B (Methodol.)*, **39**(1), 1–22.

Doğru, F.Z. & Arslan, O. (2016). Robust mixture regression using mixture of different distributions. In *Recent Advances in Robust Statistics: Theory and Applications*, pp. 57–79, Springer.

Fillbrunn, A. & Berthold, M.R. (2015). Diversity-driven widening of hierarchical agglomerative clustering. In *Advances in Intelligent Data Analysis XIV: 14th International Symposium, IDA 2015, Saint Etienne. France, October 22-24, 2015. Proceedings 14*, pp. 84–94, Springer.

Fischer, I. & Poland, J. (2005). Amplifying the block matrix structure for spectral clustering. In *Proceedings of the 14th Annual Machine Learning Conference of Belgium and the Netherlands*, pp. 21–28, Citeseer.

Forina, M., Armanino, C., Lanteri, S. & Tiscornia, E. (1983). Classification of olive oils from their fatty acid composition. In *Food Research and Data Analysis: Proceedings from the IUFOST Symposium, September 20-23, 1982, Oslo, Norway/edited by H. Martens and H. Russwurm, Jr*, London: Applied Science Publishers, 1983.

Frühwirth-Schnatter, S. (2006). *Finite mixture and markov switching models*. Springer.

Good, I.J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, **40** (3-4), 237–264.

Grimmer, J., Roberts, M.E. & Stewart, B.M. (2021). Machine learning for social science: An agnostic approach. *Ann. Rev. Polit. Sci.*, **24**, 395–419.

Halgreen, C. (1979). Self-decomposability of the generalized inverse Gaussian and hyperbolic distributions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, **47**(1), 13–17.

Hipp, J.R. & Bauer, D.J. (2006). Local solutions in the estimation of growth mixture models. *Psychol. Methods*, **11**(1), 36.
Hubert, L. & Arabie, P. (1985). Comparing partitions. *J. Classif.*, **2**, 193–218.
Jorgensen, B. (2012). *Statistical properties of the generalized inverse Gaussian distribution*, Vol. **9**. Springer Science & Business Media.
Kim, W., Kanezaki, A. & Tanaka, M. (2020). Unsupervised learning of image segmentation based on differentiable feature clustering. *IEEE Trans. Image Process.*, **29**, 8055–8068.
Kreer, J. (1957). A question of terminology. *IRE Trans. Inform. Theory*, **3**(3), 208–208.
Lee, S. & McLachlan, G.J. (2014). Finite mixtures of multivariate skew t-distributions: some recent and new results. *Stat. Comput.*, **24**, 181–202.
Li, T., Kou, G., Peng, Y. & Philip, S.Y. (2021). An integrated cluster detection, optimization, and interpretation approach for financial data. *IEEE Trans. Cybern.*, **52**(12), 13848–13861.
Lin, T.I. (2009). Maximum likelihood estimation for multivariate skew normal mixture models. *J. Multiv. Anal.*, **100** (2), 257–265.
Lubke, G. & Muthén, B.O. (2007). Performance of factor mixture models as a function of model size, covariate effects, and class-specific parameters. *Struct. Equ. Model.: A Multidiscip. J.*, **14**(1), 26–47.
MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. **1**, pp. 281–297, Oakland, CA, USA.
Massey Jr, F.J. (1951). The kolmogorov-smirnov test for goodness of fit. *J. Am. Stat. Assoc.*, **46**(253), 68–78.
McLachlan, G.J. & Krishnan, T. (2007). *The em algorithm and extensions*. John Wiley & Sons.
McLachlan, G.J., Lee, S.X. & Rathnayake, S.I. (2019). Finite mixture models. *Ann. Rev. Stat. Appl.*, **6**, 355–378.
McNeil, A.J., Frey, R. & Embrechts, P. (2015). *Quantitative risk management: concepts, techniques and tools-revised edition*. Princeton university press.
McNicholas, S.M., McNicholas, P.D. & Browne, R.P. 2013. Mixtures of variance-gamma distributions. arXiv preprint arXiv:1309.2695.
Melnykov, V. & Melnykov, I. (2012). Initializing the EM algorithm in Gaussian mixture models with an unknown number of components. *Comput. Stat. Data Anal.*, **56**(6), 1381–1395.
Meng, X.-L. & Rubin, D.B. (1993). Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika*, **80**(2), 267–278.
Murphy, K.P. (2012). *Machine learning: a probabilistic perspective*. MIT press.
O'Hagan, A., Murphy, T.B., Gormley, I.C., McNicholas, P.D. & Karlis, D. (2016). Clustering with the multivariate normal inverse Gaussian distribution. *Comput. Stat. Data Anal.*, **93**, 18–30.
Pal, S. & Heumann, C. (2022). Clustering compositional data using dirichlet mixture model. *Plos One*, **17**(5), e0268438.
Petegrosso, R., Li, Z. & Kuang, R. (2020). Machine learning and statistical methods for clustering single-cell rna-sequencing data. *Brief. Bioinform.*, **21**(4), 1209–1223.
Prates, M.O., Lachos, V.H. & Cabral, C.R.B. (2013). mixsmsn: Fitting finite mixture of scale mixture of skew-normal distributions. *J. Stat. Softw.*, **54**, 1–20.
Rosenberg, A. & Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL)*, pp. 410–420.
Schwarz, G. (1978). Estimating the dimension of a model. *The Ann. Stat.*, 461–464.
Shireman, E., Steinley, D. & Brusco, M.J. (2017). Examining the effect of initialization strategies on the performance of Gaussian mixture modeling. *Behav. Res. Methods*, **49**(1), 282–293.
Steinley, D. & Brusco, M.J. (2011). Evaluating mixture modeling for clustering: recommendations and cautions. *Psychol. Methods*, **16**(1), 63.
Van Rossum, G. & Drake, F.L. (2009). *Python 3 reference manual*. CreateSpace: Scotts Valley, CA.
Vrbik, I. & McNicholas, P.D. (2012). Analytic calculations for the em algorithm for multivariate skew-t mixture models. *Stat. Probab. Lett.*, **82**(6), 1169–1174.
Ward Jr, J.H. (1963). Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.*, **58**(301), 236–244.
Yang, M.-S., Lai, C.-Y. & Lin, C.-Y. (2012). A robust EM clustering algorithm for Gaussian mixture models. *Pattern Recogn.*, **45**(11), 3950–3961.
Zehra Doru, F., Murat Bulut, Y. & Arslan, O. (2021). Finite mixtures of multivariate skew laplace distributions. *REVSTAT-Stat. J.*, **19**(1), 35–46. https://revstat.ine.pt/index.php/REVSTAT/article/view/330
Zhong, P. & Fukushima, M. (2007). Regularized nonsmooth newton method for multi-class support vector machines. *Optim. Methods Softw.*, **22**(1), 225–236.

APPENDIX A

A.1 Properties of GIG Distribution

We now give some properties of the GIG distribution, which are required for parameter estimation in the later stage.

$$E[W] = \sqrt{\frac{\chi}{\psi}}\frac{K_{\lambda+1}\left(\sqrt{\psi\chi}\right)}{K_{\lambda}\left(\sqrt{\psi\chi}\right)}, \tag{A1}$$

$$E\left[\frac{1}{W}\right] = \sqrt{\frac{\psi}{\chi}}\frac{K_{\lambda+1}\left(\sqrt{\psi\chi}\right)}{K_{\lambda}\left(\sqrt{\psi\chi}\right)} - \frac{2\lambda}{\chi}, \tag{A2}$$

$$E[\log W] = \log\sqrt{\frac{\psi}{\chi}} + \frac{1}{K_{\lambda}\left(\sqrt{\psi\chi}\right)}\frac{\partial}{\partial\lambda}K_{\lambda}\left(\sqrt{\psi\chi}\right). \tag{A3}$$

A.2 Estimation of MGH Distribution Parameters

It is possible to show that the conditional distribution of $\boldsymbol{X}|W = w$ follows a multivariate normal distribution:

$$(\boldsymbol{X}|W = w) \sim N(\boldsymbol{\mu} + w\boldsymbol{\gamma},\, w\boldsymbol{\Sigma}). \tag{A4}$$

For all the limiting distributions of MGH distributions, it is enough if we estimate the parameters of a MGH distribution. In this regard, let us now parameterise $(\lambda, \chi, \psi, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma}) \rightarrow (\lambda, \overline{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma})$. The formulas to switch between these two parametrisations are given below.

- $(\lambda, \chi, \psi, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma}) \rightarrow (\lambda, \overline{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma})$:
  Set $K = \sqrt{\frac{\chi}{\psi}}\frac{K_{\lambda+1}\left(\sqrt{\chi\psi}\right)}{K_{\lambda}\left(\sqrt{\chi\psi}\right)}$

$$\overline{\alpha} = \sqrt{\chi\psi},\ \boldsymbol{\Sigma} \equiv K\boldsymbol{\Sigma},\ \boldsymbol{\gamma} \equiv K\boldsymbol{\gamma}. \tag{A5}$$

- $(\lambda, \overline{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma}) \rightarrow (\lambda, \chi, \psi, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma})$:

$$\psi = \overline{\alpha}\frac{K_{\lambda+1}(\overline{\alpha})}{K_{\lambda}(\overline{\alpha})} \text{ and } \chi = \overline{\alpha}\frac{K_{\lambda}(\overline{\alpha})}{K_{\lambda+1}(\overline{\alpha})}. \tag{A6}$$

Using the parametrisation $(\lambda, \overline{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma})$, the parameters can be estimated using a multicycle, expectation, conditional maximisation (MCECM) algorithm (Meng & Rubin, 1993). A ECM algorithm replaces the original M-step of EM with several computationally simpler conditional maximisation (CM) step. Each CM-step maximises the expected complete data log-likelihood found on the preceding E-step subject to the constraints on the parameter space. In other words, here, we perform one E-step before each CM-step. This technique was previously used by Breymann & Lüthi (2013) for estimating the parameters of a MGH distribution. The algorithm is explained below.

Let us assume an iid data $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and the parameter space $\boldsymbol{\Theta} = (\lambda, \overline{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma})$. The log-likelihood function is given by

$$\log L(\boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = \sum_{i=1}^{N} \log f_{\boldsymbol{X}}(\boldsymbol{x}_i, \boldsymbol{\Theta}). \tag{A7}$$

As it is not possible to maximise the log-likelihood function directly let us introduce a latent variable W which has a GIG distribution. Now using Equation (A4), the log-likelihood function can be written in an augmented form as follows:

$$\begin{aligned} \log L(\boldsymbol{\Theta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N, w_1, \ldots, w_N) = & \sum_{i=1}^{N} \log f_{X|W}(\boldsymbol{x}_i|w_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma}) \\ & + \sum_{i=1}^{N} \log f_W(w_i; \lambda, \overline{\alpha}). \end{aligned} \tag{A8}$$

Now these two summands of the log-likelihood function can be maximised easily. The first part is a log of normal density and for the second part, the log density of the GIG distribution is given as

$$\log f_W(w) = \frac{\lambda}{2}\log(\psi/\chi) - \log\left(2K_\lambda\sqrt{\chi\psi}\right) + (\lambda - 1)\log w - \frac{\chi}{2}\frac{1}{w} - \frac{\psi}{2}w. \tag{A9}$$

To estimate the parameters:

(i) Some starting values of the parameters are initialised. A reasonable choice is $\lambda = 1$, $\overline{\alpha} = 1$, and $\boldsymbol{\gamma} = (0, \ldots, 0)$. The values for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are chosen as the sample mean and sample covariance, respectively.
(ii) Compute the values of $\chi^{[t]}$ and $\psi^{[t]}$ in terms of $\overline{\alpha}^{[t]}$ for t-th iteration using Equation (A6).
(iii) Let $\eta_i^{[t]} = E[w_i|\boldsymbol{x}_i; \boldsymbol{\Theta}^{[t]}]$, $\delta_i^{[t]} = E[w_i^{-1}|\boldsymbol{x}_i; \boldsymbol{\Theta}^{[t]}]$ and $\xi_i^{[t]} = E[\log w_i|\boldsymbol{x}_i; \boldsymbol{\Theta}^{[t]}]$. These values can be calculated using Equations (A1), (A2), and (A3). Now, taking the averages,

$$\overline{\eta}^{[t]} = \sum_{i=1}^{N} \eta_i^{[t]} \text{ and } \overline{\delta}^{[t]} = \sum_{i=1}^{N} \delta_i^{[t]}. \tag{A10}$$

(iv) Get an update of $\boldsymbol{\gamma}^{[t+1]}$ by setting,

$$\boldsymbol{\gamma}^{[t+1]} = \frac{1}{N}\frac{\sum_{i=1}^{N}\delta_i^{[t]}(\overline{\boldsymbol{x}} - \boldsymbol{x}_i)}{\overline{\eta}^{[t]}\overline{\delta}^{[t]} + 1}. \tag{A11}$$

(v) The update of $\boldsymbol{\mu}^{[t+1]}$ and $\boldsymbol{\Sigma}^{[t+1]}$ then obtained as

$$\boldsymbol{\mu}^{[t+1]} = \frac{1}{N}\frac{\sum_{i=1}^{N}\delta_i^{[t]}\boldsymbol{x}_i - \boldsymbol{\gamma}^{[t+1]}}{\overline{\delta}^{[t]}}. \tag{A12}$$

$$\boldsymbol{\Sigma}^{[t+1]} = \frac{1}{n}\sum_{i=1}^{N}\delta_i^{[t]}(\boldsymbol{x}_i - \boldsymbol{\mu}^{[t+1]})(\boldsymbol{x}_i - \boldsymbol{\mu}^{[t+1]})' - \overline{\eta}^{[t]}\overline{\boldsymbol{\gamma}}^{[t+1]}\overline{\boldsymbol{\gamma}}^{[t+1]'}. \tag{A13}$$

(vi) Set $\boldsymbol{\Theta}^{[t,2]} = (\lambda^{[t]}, \overline{\alpha}^{[t]}, \boldsymbol{\mu}^{[t+1]}, \boldsymbol{\Sigma}^{[t+1]}, \boldsymbol{\gamma}^{[t+1]})$ such that λ and $\overline{\alpha}$ have the old values from t -th iteration and $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, and $\boldsymbol{\gamma}$ have the updated value from the $(t+1)$-th iteration. Using these values, calculate the values of $\eta_i^{[t,2]}$, $\delta_i^{[t,2]}$ and $\xi_i^{[t,2]}$.

(vii) The second summand of the log-likelihood function can be maximised with respect to λ, χ, and ψ by replacing w, $1/w$ and $\log w$ with the respective expected values using Equations (A1), (A2), and (A3) to obtain $\lambda^{[t+1]}$, $\chi^{[t+1]}$, $\psi^{[t+1]}$ and subsequently $\overline{\alpha}^{[t+1]}$.

(viii) After that, the same process is repeated from step (ii) until convergence to get the final estimation of the parameters.

A.3 Estimation of MSN Distribution Parameters

Estimating the parameters of the multivariate skew normal distribution is challenging. We follow a technique used by Abe *et al.* (2021), which incorporates an overparameter into the conventional stochastic representation and then obtains the EM algorithm in a closed form. The stochastic representation is given below. $\begin{pmatrix} \boldsymbol{Y} \\ Y_0 \end{pmatrix} \sim N_{p+1}(\boldsymbol{0}, \boldsymbol{\Sigma})$, $\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Omega} & \tau\boldsymbol{\Omega}^{1/2}\boldsymbol{\delta} \\ \tau\boldsymbol{\delta}^T\boldsymbol{\Omega}^{1/2} & \tau^2 \end{pmatrix}$, where $\tau \in \boldsymbol{R}$ and $\boldsymbol{\delta} \in \mathbb{R}^p$.

Let us now denote, $\boldsymbol{\lambda} = \dfrac{\boldsymbol{\delta}}{\sqrt{1 - \boldsymbol{\delta}^T\boldsymbol{\delta}}}$.

Then, it can be shown that $\boldsymbol{U} = sgn(Y_0)\boldsymbol{Y}$ has a multivariate skew normal density with location $\boldsymbol{0}$, given by

$$f(\boldsymbol{u}) = 2\Phi(\boldsymbol{\lambda}^{\boldsymbol{T}}\boldsymbol{\Omega}^{-1/2}\boldsymbol{u})\phi_p(\boldsymbol{u};\, \boldsymbol{0},\, \boldsymbol{\Omega}),\ \boldsymbol{u} \in \mathbb{R}^p. \tag{A14}$$

Then we say that $\boldsymbol{X} = \boldsymbol{U} + \boldsymbol{\mu}$, has a p dimensional multivariate skew normal distribution with location $\boldsymbol{\mu}$, which is expressed as $SN_p(\boldsymbol{\mu}, \boldsymbol{\Omega}, \boldsymbol{\lambda})$.

To estimate the parameters using an EM algorithm, let us introduce a latent variable $\boldsymbol{\xi}$ which consists of parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. For N independent random samples drawn from a multivariate skew normal distribution, the expected complete data log-likelihood function for the E step can be written as

$$\begin{aligned} Q(\boldsymbol{\xi};\, \boldsymbol{\xi}') &= \sum_{i=1}^{N} E[\log f(\boldsymbol{x}_i,\, y_{0i};\, \boldsymbol{\xi})|\boldsymbol{x}_i,\, \boldsymbol{\xi}'] \\ &= -N\frac{p+1}{2}\log 2\pi \ -\ N/2\log|\boldsymbol{\Sigma}| - \frac{1}{2}tr\left(\sum_{i=1}^{N} S(\boldsymbol{x}_i,\, \boldsymbol{\mu},\, \boldsymbol{\xi}')\boldsymbol{\Sigma}^{-1}\right), \end{aligned} \tag{A15}$$

where

$$S(\boldsymbol{x}_i,\, \boldsymbol{\mu},\, \xi') = \begin{pmatrix} (\boldsymbol{x_i} - \boldsymbol{\mu})(\boldsymbol{x_i} - \boldsymbol{\mu})^T & (\boldsymbol{x_i} - \boldsymbol{\mu})E[|y_{0i}||\boldsymbol{x}_i,\, \boldsymbol{\xi}'] \\ (\boldsymbol{x_i} - \boldsymbol{\mu})^T E[|y_{0i}||\boldsymbol{x}_i,\, \boldsymbol{\xi}'] & E\left[y_{0i}^2|\boldsymbol{x}_i;\, \boldsymbol{\xi}\right]. \end{pmatrix}.$$

Let us denote, $c_\lambda = 1/\sqrt{1 + \boldsymbol{\lambda}^T\boldsymbol{\lambda}}$, $\boldsymbol{\gamma} = \boldsymbol{\Omega}^{-1/2}\boldsymbol{\lambda}$, $v_i = \boldsymbol{\gamma}^T(\boldsymbol{x_i} - \boldsymbol{\mu})$, $\rho_1(v) = \dfrac{\phi(v)}{\Phi(v)} + v$ and $\rho_2(v) = 1 + v\rho_1(v)$.

It can be shown that $E[|Y_0||\boldsymbol{X}] = \tau c_\lambda \rho_1(\boldsymbol{\gamma}^T\boldsymbol{x})$ and $E\left[Y_0^2|\boldsymbol{X}\right] = \tau^2 c_\lambda^2 \rho_2(\boldsymbol{\gamma}^T\boldsymbol{x})$

Then for the t-th iteration in the M step, the updates of the parameters are given below.

$$\hat{\boldsymbol{\mu}}^{t+1} = \bar{\boldsymbol{x}} - c_{\lambda^t}\left(\hat{\boldsymbol{\Omega}}^t\right)^{1/2}\hat{\boldsymbol{\delta}}^t\frac{1}{N}\sum_{i=1}^{N}\rho_1(\hat{v_i}^t), \tag{A16}$$

$$\hat{\boldsymbol{\Omega}}^{t+1} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x_i} - \hat{\boldsymbol{\mu}}^{t+1})(\boldsymbol{x_i} - \hat{\boldsymbol{\mu}}^{t+1})^T, \tag{A17}$$

$$\hat{\boldsymbol{\delta}}^{t+1} = \left[\frac{1}{N}\sum_{i=1}^{N}\rho_2(\hat{v_i}^t)\right]^{-1/2} \times \left(\hat{\boldsymbol{\Omega}}^{t+1}\right)^{1/2} \times \left[\frac{1}{N}\sum_{i=1}^{N}\rho_1(\hat{v_i}^t)(\boldsymbol{x_i} - \hat{\boldsymbol{\mu}}^{t+1})\right]. \tag{A18}$$

[*Received November 2023; accepted July 2024*]