Probabilistic Unsupervised Learning with Heterogeneous Noisy Data

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Samarth Pardhi

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.
April 2024

Supervisor: **Dr. Leelavati Narlikar**

Certificate

This is to certify that this dissertation entitled *Probabilistic Unsupervised Learning with Heterogeneous Noisy Data* towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by **Samarth Pardhi** at Indian Institute of Science Education and Research under the supervision of **Dr. Leelavati Narlikar**, Associate Professor, Department of Data Science, during the academic year 2023-2024.

Dr. Leelavati Narlikar

Committee:

- Dr. Leelavati Narlikar
- Dr. Moumanti Poddar

Declaration

I hereby declare that the matter embodied in the report entitled *Probabilistic Unsupervised Learning with Heterogeneous Noisy Data* are the results of the work carried out by me at the Department of Mathematics, Indian Institute of Science Education and Research, Pune, under the supervision of **Dr. Leelavati Narlikar** and the same has not been submitted elsewhere for any other degree. Wherever others contribute, every effort is made to indicate this clearly, with due reference to the literature and acknowledgement of collaborative research and discussions.

Samarth Pardhi

20191078

Acknowledgments

I am immensely grateful to my supervisor, Dr. Leelavati Narlikar, for her unwavering support and guidance throughout this journey. Her simplicity, humor, and light-hearted approach fostered an environment easy to work, allowing me to perform at my best. I extend my heartfelt gratitude to my parents and sister for their constant encouragement and invaluable support. Their love has always been a source of comfort and strength. Energy never dies, it just changes the forms. My sincere thanks to my friends, whose hangouts and invaluable support always made the challenging moments more bearable. I am indebted to my labmates for their inspiration and shared knowledge, as well as to all my fellow high spirited sports people, who provided a much-needed respite from the rigors of research. Lastly, I am grateful to all the brilliant minds I met during this journey, whose knowledge, experiences, and warm smiles enriched my days and made this experience truly rewarding.

Abstract

Mixture models are widely used in situations where the interest is in the probability densities. In such cases, the focus is usually to estimate parameters and mixing probabilities with successful clustering. But in real-life applications, analysis becomes substantially challenging for two reasons; the first one is when we introduce heterogeneous features of continuous, categorical, and even ordinal type, and the second one is too many features make the algorithm computationally expensive and not all features contribute for the inference. Feature selection will be implemented for the same. Goal of this project is to summarize the existing methods and develop model-based approaches that are robust and scalable. These approaches will enable model-based clustering simultaneously selecting relevant features within heterogeneous data. Idea is to use Bayesian framework using the Gibbs sampling techniques. These are very popular techniques in mixture models. We investigate Gaussian and Categorical features in model-based clustering, assuming the number of cluster is finite and does not grow with sample size; such frameworks are called finite mixture model. These proposed methods are compared with maximum likelihood approach, which uses the Expected-Maximisation algorithm.

Contents

1 Introduction

In high-dimensional datasets, the clustering of observations often occurs primarily within a subset of features (or variables, will be used interchangeably). Several studies ([Fowlkes et al., 1988]; [Milligan, 1989]; [Brusco et al., 2001]) have pointed out that including unnecessary features can complicate or obscure the identification of clusters. To address this issue, various methods have been proposed, such as weighting variables differently or selecting the most discriminatory ones. Research has shown that feature selection techniques tend to outperform variable weighting methods in terms of identifying the true cluster structure [Gnanadesikan, R. et al, 1995]. Moreover, besides improving cluster prediction accuracy, feature selection also reduces the resources needed for storing and processing data, making it a more cost-effective approach.

In real-world datasets, we often encounter diverse data types that naturally group together. It's important to consider these groupings when clustering populations. Such tabular datasets, containing different types of features (categorical, ordinal, numeric), are common in fields like data science and biomedical research. The Bayesian paradigm presents a unified approach to tackle feature selection and heterogeneous data clustering issues concurrently. Recent advancements in MCMC techniques [Gilks et al., 1996] have spurred significant research in both domains.

We introduce a method that addresses the simultaneous selection of discriminating features and grouping of samples into K clusters, where K is unknown. Our approach utilizes a collapsed-Gibbs sampler while sampling, which integrates out all parameters except the cluster assignment parameter.

1.1 Notation

To insure clarity and coherence throughout the document, scalars will be represented using non-bold font, with potential subscripts (e.g., γ, β_i). Vectors will be denoted by bold lowercase letters, possibly with subscripts (e.g., $\mathbf{x}$, $\mathbf{y}_j$). Vectors are formulated as column vectors, rather than row vectors. Matrices will be represented by bold uppercase letters, with potential subscripts (e.g., $\mathbf{A}$, $\mathbf{B}_k$). The transpose of a matrix $\mathbf{X}$ will be denoted as $\mathbf{X}^\top$, and the inverse will be denoted as $\mathbf{X}^{-1}$. The $(p \times p)$ identity matrix will be represented as $\mathbf{I}_p$.

Specific notations for certain quantities will be used as detailed in the tables below.

1.1.1 General notations

$f(\mathbf{x}) \propto g(\mathbf{x})$	f is proportional to g, which means $\exists c$ given $f(\mathbf{x}) = cg(\mathbf{x}) \; \forall \; \mathbf{x}$
$f(\mathbf{x} \mid \theta)$	Likelihood
$f(\theta)$	Prior probability
$f(\theta \mid \mathbf{x})$	Posterior probability
$\mathbf{d}S_M$	$(M-1)$-dimensional probability simplex in $\mathbb{R}^K$

$$\mathbb{1}\{x_0 = x\} = \begin{cases} 1, & \text{if } x_0 = x \\ 0, & \text{if } x_0 \neq x \end{cases}$$

1.1.2 Data

N	Number of data samples.
l	Number of Gaussian features.
m	Number of Categorical features.
$D = (l + m)$	Total number of features.
J	Total number of categories for the categorical data.
$g_i \in \mathbb{R}^l$	Data point for Gaussian feature.
$c_i \in \{1, 2, \ldots, J\}$	Data point for categorical feature.
$\mathbf{x}_i = \{g_1, \ldots, g_l, c_1, \ldots, c_m\} \in \mathbb{R}^D$	The i^{th} data vector.
$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$	Data, set of data vectors.
$\mathbf{X}_k$	Set of data vectors from mixture component k.
$\mathbf{X}_{k[-i]}$	Set of data vectors from the cluster k excluding data vector
N_k	Number of data vectors from the cluster k.
$N_{k[-i]}$	Number of data vectors from the cluster k, without considering i^{th} data vector.

1.1.3 Model parameters

K	Number of mixture components.
$\pi_k = f(z_i = k)$	Prior probability that the data vector $\mathbf{x}_i$ will be belonged to the cluster k.
$\pi = (\pi_1, \pi_2, \ldots, \pi_K)$	Prior assignment probability for all K clusters.
$z_i \in \{1, 2, \ldots K\}$	Latent state indicating to which cluster the observation x_i belongs to.
$\mathbf{z} = (z_1, z_2, \ldots z_N)$	Latent states vector.
μ_k	Mean vector of a Gaussian density for mixture component k.
$\mathbf{\Sigma}_k$	Multivariate Gaussian's Covariance matrix for the mixture component k.
σ_k^2	Variance of a Gaussian density for the mixture component k.
$p_j = f(c_i = j)$	Probability that i^{th} categorical data point belongs to the category $j \in 1, 2, \ldots, J$.
$\mathbf{p} = (p_1, p_2, \ldots, p_J)$	Probability vector for all such J categories.
$\phi_{kj} \in \{0, 1\}$	The feature j's relevance for the k-th cluster.

1.1.4 Hyperparameters

α_k	Dirichlet prior parameter on the mixing weight π.
$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_K)$	Probability vector for all mixture components K.
m_0	Prior mean of univariate Gaussian μ.
κ_0	How strong we are about the above prior, m_0.
S_0	Proportional to prior mean for σ^2.
ν_0	How strong we are about the above prior, S_0.
$\beta_0 = (m_0, \kappa_0, S_0, \nu_0)$	Prior parameters for the Normal-inverse-Chi-squared prior on mean μ and variance σ^2 parameters of the Gaussian data.
$\gamma_{0,j}$	Prior Dirichlet parameter for the category $j \in \{1, 2, \ldots J\}$ in categorical data.
$\gamma_0 = (\gamma_{0,1}, \ldots, \gamma_{0,J})$	Prior vector for all categories J.
$\mathcal{H} = \{\alpha, \beta, \gamma\}$	Set of all hyperparameters.

2 Probabilistic Inference Methods

2.1 Bayesian Statistics

In contemporary statistics, Bayesian methods have gained growing significance and widespread application. Although Thomas Bayes conceived this idea, he passed away before officially presenting it. Fortunately, his associate Richard Price continued his research and released it in

1764. In this section, we discuss basic ideas related to Bayesian theory:

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ represent the observations of N data points, assumed to be independent and identically distributed (i.i.d.) with a probability parameterized by $\mathbf{\Theta}$. It's worth noting that the parameters $\mathbf{\Theta}$ may encompass hidden variables, such as latent variables in a mixture model indicating the cluster to which a data point belongs to.

The methodology of Bayesian analysis involves to assume a prior probability distribution for $\mathbf{\Theta}$ with hyperparameters β, denoted as $f(\mathbf{\Theta} \mid \beta)$. This distribution captures how believable each possible value of $\mathbf{\Theta}$ is before looking at the data. To understand $\mathbf{\Theta}$ better, we focus on its distribution after seeing the data, known as the posterior distribution. Mathematically, this is shown using Bayes' theorem:

$$f(\mathbf{\Theta}|\mathbf{X}, \beta) = \frac{f(\mathbf{X}|\mathbf{\Theta})f(\mathbf{\Theta}|\beta)}{f(\mathbf{X}|\beta)} = \frac{f(\mathbf{X}|\mathbf{\Theta})f(\mathbf{\Theta}|\beta)}{\int_{\mathbf{\Theta}} f(\mathbf{X}, \mathbf{\Theta}|\beta)d\mathbf{\Theta}} = \frac{f(\mathbf{X}|\mathbf{\Theta})f(\mathbf{\Theta}|\beta)}{\int_{\mathbf{\Theta}} f(\mathbf{X}|\mathbf{\Theta})f(\mathbf{\Theta}|\beta)f\mathbf{\Theta}} \propto f(\mathbf{X}|\mathbf{\Theta})f(\mathbf{\Theta}|\beta) \tag{1}$$

Here, $\mathbf{X}$ represents the observed data set. In essence, the posterior probability is proportional to the product of the likelihood and the prior probability.

In a broader context, the Bayesian approach involves assuming a prior distribution for any unknowns (in this case, $\mathbf{\Theta}$) and applying probability rules to address specific questions of interest. For example, finding the parameter by selecting the highest posterior probability of β results in the maximum a posteriori (MAP) estimator.

In many real-world scenarios, computing the posterior distribution analytically proves to be challenging. As a result, numerical approaches serve as viable alternatives. Markov Chain Monte Carlo (MCMC) stands out as a highly effective category of numerical Bayesian methods. MCMC techniques simulate a Markov chain with a stationary distribution that converges to the desired posterior distribution of the parameters. The data generated from these simulations are then used for computing Bayesian estimates and performing uncertainty analyses.

2.2 Monte Carlo Markov Chain

Determining the posterior through Bayesian estimation can be challenging at times. While we often manage to identify the functional form of *Likelihood* multiplied by *Prior*. In certain situations, performing inference in high-dimensional spaces can be computationally impractical. In such instances, we opt for approximation methods, often employing sampling approaches. Markov chain Monte Carlo (MCMC) algorithms, also called samplers, are numerical approximation algorithms. The historical development of MCMC is intriguing, with its roots traced back to physicists at Los Alamos during World War II, first published by Metropolis et al. (1953) in

a chemistry journal, later extended by Hastings (1970), and independently rediscovered in the context of Ising models by Geman and Geman (1984) [Geman and Geman 1984]. The fundamental principle underlying MCMC is the construction of a Markov chain on the state space $\mathbf{X}$ with a stationary distribution equal to the target density $f(^*\mathbf{X})$ of interest, which could be either a prior or a posterior distribution. This random walk on the state space ensures that the time spent in each state x is proportionate to the target density $f(^*\mathbf{X})$. By drawing correlated samples $(x_0, x_1, x_2, \ldots)$ from this chain, Monte Carlo integration can be performed with respect to $f(^*\mathbf{X})$.

Intuitively, this method resembles a stochastic hill-climbing technique for inference, working across the entire dataset. The primary focus of this inference strategy is to allocate computational resources towards sampling points from the regions of high probability in the true target posterior distribution $f(\mathbf{\Theta} \mid \mathbf{X})$ ([Andrieu et al., 2003]; [Bonawitz 2008]; [Geyer 2011]).

The stochastic walk is characterized by a Markov chain property, where the selection of the state at time $t+1$ depends only on the preceding state at time t. This memoryless property offers two primary benefits: MCMC techniques can operate for an indefinite number of iterations without requiring additional memory resources, and the stochastic walk can be characterized by the transition kernel $f(\mathbf{\Theta_{t+1}}|\mathbf{\Theta_t})$. The transition kernel is crucial, as its properties determine the convergence of the MCMC algorithm.

Convergence is ensured if the transition kernel meets two criteria: it has an invariant (or stationary) distribution, and it is ergodic. Invariant distribution ensures that the distribution remains unchanged under the transition kernel, ergodicity ensures that every state is accessible from any other state, and the stochastic process avoids cyclic trapping. A variety of MCMC algorithms exist, such as Metropolis-Hastings (MH), Gibbs sampling, Hamiltonian Monte Carlo, slice sampling, Adaptive rejection sampling, and more. Notably, all Markov Chain Monte Carlo (MCMC) algorithms are instances of the Metropolis-Hastings (MH) algorithm.

The Metropolis-Hastings (MH) algorithm, a generalization of the Metropolis-within-Gibbs (MWG) algorithm, is a widely applicable MCMC method. In MH, an arbitrary proposal kernel $f(\Theta'|\Theta_t)$ is converted into a transition kernel with the desired invariant distribution $f(^*\mathbf{X})$. The MH acceptance probability determines whether a proposed state Θ' is accepted or rejected. The key is to choose the proposal kernel carefully, ensuring both convergence to the target distribution and computational efficiency. The MH algorithm, with its flexibility and generality, plays a central role in Bayesian inference, aiming to maximize the unnormalized joint posterior distribution and collect samples for subsequent inference queries.

In summary, MCMC methods have evolved from their origins in atomic bomb research to becoming indispensable tools in Bayesian statistics and machine learning. These algorithms of-

fer a versatile and robust approach for exploring complex parameter spaces and approximating intricate posterior distributions, making them a cornerstone in modern statistical methodologies.

2.2.1 Gibbs Sampler

Gibbs sampling, a widely-used technique in Markov Chain Monte Carlo (MCMC) methods, traces its origins back to the work of [Turchin 1971]. However, it gained widespread popularity through the contributions of [Geman and Geman 1984], who applied it to the domain of image restoration. The algorithm derives its name from the renowned physicist J. W. Gibbs, drawing an analogy between the sampling process and principles of statistical physics.

The Gibbs sampling method finds its application in situations where the joint distribution of variables is either unknown or poses challenges for direct sampling. When we know the conditional distribution of each variable and they can be sampled from with ease, Gibbs sampling emerges as a valuable tool. The essence of this algorithm lies in its ability to sequentially sample from the conditional distribution of each parameter or variable, given the current values of the remaining variables.

Consider a scenario where we have data $\mathbf{X}$ and parameters, $\mathbf{\Theta} = \{\theta_1, \theta_2, ..., \theta_p\}$ which parameterise the probability distribution $f(\Theta \mid \mathbf{X})$. The Gibbs sampling process involves iteratively drawing samples from the distribution:

$$\theta_i^{(t)} \sim f(\theta_i | \theta_{-i}^{(t-1)}, \mathbf{X}, \mathbf{\Theta}) \tag{2}$$

Here, $\theta_{-i}^{(t-1)}$ represents $\mathbf{\Theta}$'s all current values at the $(t-1)$-th iteration, excluding θ_i. With sufficiently long sampling, these drawn values converge to random samples from the distribution $f(\theta_i | \theta_{-i}, \mathbf{X})$, where:

$$f(\theta_i | \theta_{-i}, \mathbf{X}) \propto \frac{f(\theta_1, \theta_2, ..., \theta_p, \mathbf{X})}{f(\theta_{-i}, \mathbf{X})}$$

This proportionality arises from the fact that the conditional distribution is proportional to the joint distribution. Omitting constant terms relative to the conditioned parameters offers computational advantages.

In a simplified illustration, if we have a joint probability distribution $f(\theta_1, \theta_2 | \mathbf{X})$, the Gibbs sampler alternates between drawing samples from $f(\theta_1 | \theta_2, X)$ and $f(\theta_2 | \theta_1, \mathbf{X})$ in an iterative manner. This process defines a sequence of realizations of the random variables θ_1 and θ_2, converging to the joint distribution $f(\theta_1, \theta_2)$.

2.2.2 Collapsed Gibbs sampling

In some cases, it is possible to perform analytical integration over certain unknown quantities while sampling the remaining ones. This technique is referred to as collapsed Gibbs sampling and is often preferred due to its efficacy in lower-dimensional spaces.

To elaborate, consider a scenario where we have a parameter set $\mathbf{\Theta}$ along with an additional latent variable $\mathbf{z}$. In this context, we first sample $\mathbf{z}$ and then integrate out $\mathbf{\Theta}$. Consequently, the $\mathbf{\Theta}$ parameters are excluded from the Markov chain. As a result, we can generate conditionally independent samples $\theta^* \sim f(\theta|\mathbf{z}^*, \mathbf{X})$, which exhibit considerably reduced variance compared to samples drawn from the joint state space [Liu et. al. 1994]. This technique is commonly known as Rao-Blackwellization, named after the theorem:

Rao-Blackwell Theorem

Let $\mathbf{\Theta}$ and $\mathbf{z}$ be random variables, dependent on each other and $f(\mathbf{\Theta}, \mathbf{z})$ represent some scalar function. Then, the following inequality holds:

$$\mathrm{Var}_{\mathbf{\Theta},\mathbf{z}}[f(z, \mathbf{\Theta})] \geq \mathrm{Var}_{\mathbf{z}}[\mathrm{E}_{\mathbf{\Theta}}[f(\mathbf{z}, \mathbf{\Theta}) \mid \mathbf{z}]] \tag{3}$$

This theorem assures that the variance of the estimate obtained by integrating out $\mathbf{\Theta}$ analytically will consistently be lower (or at least not higher) than the variance of a direct Monte Carlo estimate. We will see practical implementations of this in our model.

3 Mixture Models

Mixture distribution arise when the measurements are taken under different conditions,

The basic idea of a mixture model is to express the overall distribution of the data as a weighted sum of individual component distributions. Mathematically, given a set of observations $\mathbf{X}$, a mixture model can be defined as:

$$f(\mathbf{X}) = \sum_{k=1}^{K} \pi_k f(\mathbf{X} \mid \theta_k)$$

Here, π_k represents the mixing proportion of the i-th component, $f(\mathbf{X} \mid \theta_k)$ is the probability density function of the i-th component, and θ_i denotes the parameters associated with that component.

Model-Based Clustering

Model-based clustering is a powerful approach that assumes the observed data is generated by a mixture of several underlying probability distributions. The goal is to identify the cluster assignments of the data points and estimate the parameters of the component distributions. This method offers flexibility in capturing different clustering structures, accommodating clusters of varying shapes and sizes.

When contrasting with conventional deterministic clustering techniques such as k-means, model-based clustering offers several benefits. Deterministic approaches hinge on distance metrics between objects and centroids to return coherent clusters. Nevertheless, they presuppose uniform structure across clusters and struggle with clusters of varying shapes, sizes, and orientations. Conversely, model-based clustering exhibits enhanced capability in accommodating overlapping groups by factoring in cluster membership probabilities within these areas.

In essence, model-based clustering is a more flexible and robust approach that can capture complex cluster structures and overlap, making it a powerful tool for clustering analysis.

3.1 Conjugate Priors for Mixture Models

In Bayesian statistics, conjugate priors are a cornerstone, simplifying posterior computations and facilitating analytical solutions. These priors, belonging to the same distribution family as the posterior, streamline the update of beliefs with new data. This chapter delves into the overarching concept of conjugate priors, their analysis, and their application in mixture models.

The **Marginal Likelihood**, also known as evidence or the marginal likelihood of the data, is a pivotal element in Bayesian analysis. It acts as a normalization constant, ensuring that the posterior distribution integrates to one. Mathematically, the marginal likelihood is calculated through the integral:

$$f(\mathbf{X}) = \int f(\mathbf{X} \mid \theta) f(\theta)\, d\theta \tag{4}$$

Here, $\mathbf{X}$ represents the observed data, $f(\mathbf{X} \mid \theta)$ is the likelihood function, and $f(\theta)$ is the prior distribution. This measure provides insight into the model's fit to the data across various parameter settings.

Another essential quantity is the **Posterior Predictive**. This tool assesses model adequacy by generating new data points based on estimated parameters. The posterior predictive distribution is calculated as:

$$f(\tilde{\mathbf{X}} \mid \mathbf{X}) = \int f(\tilde{\mathbf{X}} \mid \theta) f(\theta \mid \mathbf{X})\, d\theta \tag{5}$$

Here, $\tilde{\mathbf{X}}$ denotes new data, $f(\tilde{\mathbf{X}} \mid \theta)$ is the predictive likelihood, and $f(\theta \mid \mathbf{X})$ is the posterior distribution. Comparing observed data with samples from the posterior predictive distribution aids in model validation.

Understanding the synergy of conjugate priors, marginal likelihood, and posterior predictive checks is pivotal in the context of mixture models, enhancing the rigor of Bayesian analysis.

3.1.1 Preliminaries

Dirichlet Distribution

The Dirichlet distribution, denoted as $\text{Dir}(\alpha_1, \ldots, \alpha_K)$, is characterized by positive scalars $\alpha_i > 0$ for $i = 1, \ldots, K$, where $K \geq 2$. The Dirichlet distribution is defined on the support of the $(K-1)$-dimensional simplex S_K, encompassing all K-dimensional vectors that constitute valid probability distributions. The probability density of $x = (x_1, \ldots, x_K)$ when $x \in S_K$ is given by:

$$f(x_1, \ldots, x_K; \alpha_1, \ldots, \alpha_K) = \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} x_i^{\alpha_i - 1} \tag{6}$$

Several pedagogical considerations should be noted. Firstly, the probability density function $f(x)$ is technically defined only in $(K-1)$-dimensional space and not in K dimensions. This is due to the fact that S_K has zero measure in K dimensions. Secondly, the support is, in reality, the open simplex, implying that all $x_i > 0$ (which also implies none of the x_i equals 1).

The Dirichlet distribution serves as a generalization of the Beta distribution, which acts as the conjugate prior for binary events like coin flipping.

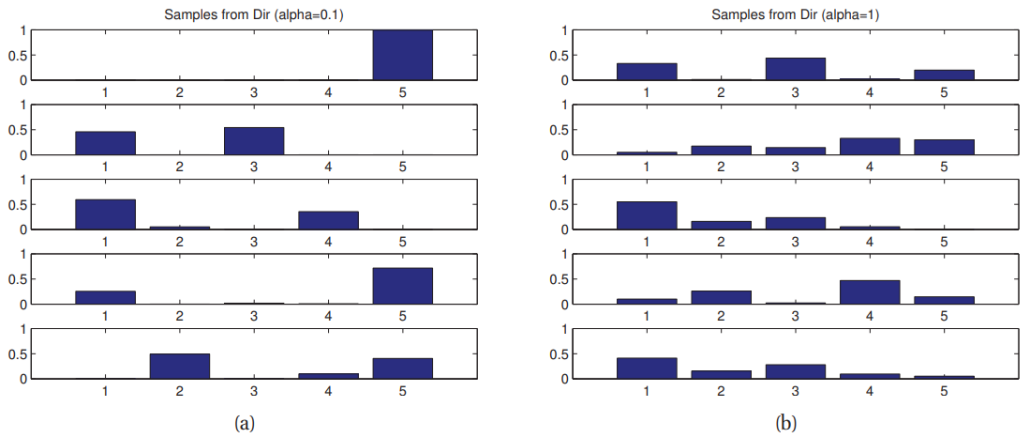


Figure 1: Samples drawn from a 5-dimensional symmetric Dirichlet distribution with varying parameter values. (a) $\alpha = (0.1, 0.1, \ldots, 0.1)$. This yields a highly sparse distribution. (b) $\alpha = (1, 1, \ldots, 1)$. This results in a more uniform distribution. Figure is taken from [Murphy, 2007]

Often we use a symmetric Dirichlet prior, specially in Bayesian setting where we take it as uninformative prior of the form $\alpha_k = \alpha/K$, the mean becomes $1/K$, and the variance becomes $\frac{K-1}{k^2(\alpha+1)}$. So increasing alpha decreases the variance, increases the precision of the distribution. Refer to Figure 1 for some sampled probability vectors.

Inverse Wishart

This distribution is the generalization of the inverse Gamma. Consider a dd a positive definite (covariance) matrix $\mathbf{X}$ and a degree of freedom parameter $\nu > d-1$ and positive semi-definite matrix $\mathbf{S}$.

$$IW_\nu(\mathbf{X} \mid \mathbf{S}^{-1}) = \frac{1}{Z}|\mathbf{X}|^{-(d+\nu+1)/2} \exp\left(-\frac{1}{2} Tr(\mathbf{S}\mathbf{X}^{-1}) \right) \tag{7}$$

where,

$$Z = \frac{|\mathbf{S}|^{\nu/2}}{2^{\nu d/2}\pi^{d(d-1)/4}\prod_{i=1}^{d}\Gamma(\frac{\nu+1-i}{2})} \tag{8}$$

Inverse-Chi-squared

This inverse-chi-squared distribution is the reparameterization of the inverse Gamma.

$$\mathcal{X}^{-2}(x \mid \nu, \sigma^2) = \frac{1}{\Gamma(\nu/2)}\left(\frac{\nu\sigma^2}{2}\right)^{\nu/2} x^{-\frac{\nu}{2}-1}\exp[-\frac{\nu\sigma^2}{2x}], x > 0 \tag{9}$$

The parameter $\nu > 0$ is termed the degree of freedom, while $\sigma^2 > 0$ represents the scale. In Section 29, we will demonstrate how varying ν results in the effectiveness of the prior.

3.1.2 Multivariate Gaussian with a Conjugate Prior

Likelihood

The likelihood of random vectors $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ originates from a multivariate Gaussian distribution with mean μ and covariance $\mathbf{\Sigma}$ is described as follows:

$$\begin{aligned} f(\mathbf{X} \mid \mu, \mathbf{\Sigma}) &= \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n \mid \mu, \mathbf{\Sigma}) \\ &= (2\pi)^{-ND/2} \mid \mathbf{\Sigma} \mid^{N/2} \exp\left(-\frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n - \mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}_n - \mu) \right) \end{aligned} \tag{10}$$

We can show that,

$$\sum_{n=1}^{N}(\mathbf{x}_n-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}_n-\mu)=tr(\Sigma^{-1}\mathbf{S}_{\bar{\mathbf{x}}})+N(\bar{\mathbf{x}}-\mu)^T\mathbf{\Sigma}^{-1}(\bar{\mathbf{x}}-\mu). \tag{11}$$

The likelihood can be written as,

$$f(\mathbf{X}\mid\mu,\mathbf{\Sigma})=(2\pi)^{-ND/2}\mid\mathbf{\Sigma}\mid^{N/2}\exp\left(-\frac{N}{2}(\mu-\bar{\mathbf{x}})^T\mathbf{\Sigma}^{-1}(\mu-\bar{\mathbf{x}})-\frac{1}{2}\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{S}_{\bar{\mathbf{x}}})\right) \tag{12}$$

where

$$\mathbf{S}_{\bar{\mathbf{x}}}=\sum_{n=1}^{N}(\mathbf{x}_n-\bar{\mathbf{x}})(\mathbf{x}_n-\bar{\mathbf{x}})^T$$

$$\bar{\mathbf{x}}=\frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

Prior on parameters

The Gaussian-Inverse-Wishart (GIW) prior provides complete conjugacy for the mean μ and covariance matrix $\mathbf{\Sigma}$ of a multivariate Gaussian distribution.

$$\begin{aligned}\mathrm{GIW}(\mu,\mathbf{\Sigma})&=\mathcal{N}(\mu\mid\mathbf{m}_0,\frac{1}{\kappa_0}\mathbf{\Sigma})\cdot\mathrm{IW}(\mathbf{\Sigma}\mid\mathbf{S}_0,\nu_0)\\&=\frac{1}{Z_{\mathrm{GIW}}(D,\kappa_0,\nu_0,\mathbf{S}_0)}|\mathbf{\Sigma}|^{-1/2}\exp\left(\frac{\kappa_0}{2}(\mu-\mathbf{m}_0)^T\mathbf{\Sigma}^{-1}(\mu-\mathbf{m}_0)\right)|\mathbf{\Sigma}|^{-\frac{\nu_0+D+1}{2}}\\&\quad\cdot\quad\exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{S}_0)\right)\end{aligned} \tag{13}$$

with

$$Z_{\mathrm{GIW}}(D,k_0,\nu_0,\mathbf{S}_0)=2^{\frac{\nu_0+1}{2}}\pi^{D(D+1)/4}\kappa_0^{-D/2}|\mathbf{S}_0|^{-\nu_0/2}\prod_{i=1}^{D}\Gamma\left(\frac{\nu_0+1-i}{2}\right) \tag{14}$$

Thus, the full conjugate is given as,

$$f(\mu,\mathbf{\Sigma})=\mathrm{GIW}(\mu,\mathbf{\Sigma}\mid\mathbf{m}_0,\kappa_0,\nu_0,\mathbf{S}_0) \tag{15}$$

Referring to the hyperparameter notation in (1.1.4), the intuition behind the parameters κ_0 and ν_0 remains the same as in the univariate case. However, $\mathbf{m}_0$ and $\mathbf{S}_0$ are expressed in vector form for this multivariate Gaussian distribution.

Full Joint

$$
\begin{aligned}
f(\mathbf{X}, \mu, \mathbf{\Sigma}) &= f(\mathbf{X} \mid \mu, \mathbf{\Sigma}) f(\mu, \mathbf{\Sigma}) \\
&= \frac{(2\pi)^{-ND/2}}{Z_{\text{GIW}}(D, \kappa_0, \nu_0, \mathbf{S}_0)} |\mathbf{\Sigma}|^{-\frac{\nu_0+N+D+2}{2}} \exp\Bigg(-\frac{N}{2}(\mu - \bar{\mathbf{x}})^T \mathbf{\Sigma}^{-1}(\mu - \bar{\mathbf{x}}) \\
&\qquad\qquad -\frac{\kappa_0}{2}(\mu - \bar{\mathbf{x}})^T \mathbf{\Sigma}^{-1}(\mu - \bar{\mathbf{x}}) \Bigg) \\
&= \frac{(2\pi)^{-ND/2}}{Z_{\text{GIW}}(D, \kappa_0, \nu_0, \mathbf{S}_0)} |\mathbf{\Sigma}|^{-\frac{\nu_0+N+D+2}{2}} \exp\Bigg\{ -\frac{\kappa_0 + N}{2}\left(\mu - \frac{\kappa_0 \mathbf{m}_0 + N\bar{\mathbf{x}}}{\kappa_N}\right)^T \\
&\qquad \mathbf{\Sigma}^{-1}\left(\mu - \frac{\kappa_0 \mathbf{m}_0 + N\bar{\mathbf{x}}}{\kappa_N}\right) - \frac{1}{2}\text{tr}\left[\mathbf{\Sigma}^{-1}\left(\mathbf{S}_0 + \mathbf{S}_{\bar{\mathbf{x}}} + \frac{\kappa_0 N}{\kappa_0 + N}(\bar{\mathbf{x}} - \mathbf{m}_0)(\bar{\mathbf{x}} - \mathbf{m}_0)^T\right)\right]\Bigg\}
\end{aligned}
\tag{16}
$$

Posterior of Parameters

The posterior of the parameters μ and $\mathbf{\Sigma}$ is given by:

$$
f(\mu, \mathbf{\Sigma}) \propto f(\mathbf{X} \mid \mu, \mathbf{\Sigma}) f(\mu, \mathbf{\Sigma}) \tag{17}
$$

The right-hand side of the above equation is the full joint distribution given in equation (16). By comparing equations (13) and (16), it follows that the posterior is a Gaussian-Inverse-Wishart (GIW) density with updated parameters:

$$
f(\mu, \mathbf{\Sigma} \mid \mathbf{X}) = \text{GIW}(\mu, \mathbf{\Sigma} \mid \mathbf{m}_N, \kappa_N, \mathbf{S}_N, \nu_N) \tag{18}
$$

$$
\begin{aligned}
\mathbf{m}_N &= \frac{\kappa_0 \mathbf{m}_0 + N\bar{\mathbf{x}}}{\kappa_N} \\
\kappa_N &= \kappa_0 + N \\
\nu_N &= \nu_0 + N \\
\mathbf{S}_N &= \mathbf{S}_0 + \mathbf{S}_{\bar{\mathbf{x}}} + \frac{\kappa_0 N}{\kappa_0 + N}(\bar{\mathbf{x}} - \mathbf{m}_0)(\bar{\mathbf{x}} - \mathbf{m}_0)^T \\
&= \mathbf{S}_0 + \mathbf{S} + \kappa_0 \mathbf{m}_0 \mathbf{m}_0^T - \kappa_N \mathbf{m}_N \mathbf{m}_N^T
\end{aligned}
\tag{19}
$$

where we define, $\mathbf{S} = \sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^T$.

Marginal Likelihood of Data

From the full joint distribution given in equation (16), we can derive the marginal likelihood of the data in the following way:

$$\begin{aligned} f(\mathbf{X}) &= \int_{\mu}\int_{\mathbf{\Sigma}} f(\mathbf{X},\mu,\mathbf{\Sigma})d\mu\mathbf{\Sigma} = \int_{\mu}\int_{\mathbf{\Sigma}} f(\mathbf{X}\mid\mu,\mathbf{\Sigma})f(\mu,\mathbf{\Sigma})d\mu d\mathbf{\Sigma} \\ &= \frac{(2\pi)^{-ND/2}}{Z_{\text{GIW}}(D,\kappa_0,\mathbf{S}_0,\nu_0)}\int_{\mu}\int_{\mathbf{\Sigma}} |\mathbf{\Sigma}|^{-\frac{\nu_0+N+D+2}{2}} \\ &\qquad \times \quad \exp\left(-\frac{\kappa_N}{2}(\mu-\mathbf{m}_N)\mathbf{\Sigma}^{-1}(\mu-\mathbf{m}_N) - \frac{1}{2}\text{tr}(\mathbf{\Sigma^{-1}S_N})\right)d\mu d\mathbf{\Sigma} \\ &= (2\pi)^{-ND/2}\frac{Z_{GIW}(D,\kappa_N,\mathbf{S}_N,\nu_N)}{Z_{\text{GIW}}(D,\kappa_0,\mathbf{S}_0,\nu_0)} \\ &= \pi^{-ND/2}\frac{\kappa_0^{D/2}|\mathbf{S}_0|^{\nu_0/2}}{\kappa_N^{D/2}|\mathbf{S}_N|^{\nu_N/2}}\prod_{i=1}^{D}\frac{\left(\frac{\nu_N+1-i}{2}\right)}{\left(\frac{\nu_0+1-i}{2}\right)} \end{aligned} \tag{20}$$

Posterior Predictive

Let's consider that we have a new data vector $\mathbf{x}^*$. In this case, the posterior predictive distribution for this vector can be expressed as:

$$f(\mathbf{x}^*\mid\mathbf{X}) = \frac{f(\mathbf{x}^*,\mathbf{X})}{f(\mathbf{X})} \tag{21}$$

The denominator is defined by equation (20). Similarly, the numerator can be derived by considering the marginal likelihood of the augmented set $\{\mathbf{X},\mathbf{x}^*\}$. Utilizing the notation $\mathbf{S}_{N*}$ to represent the calculation in equation (19) on this new augmented set, equation (21) can be evaluated as follows:

$$\begin{aligned} f(\mathbf{x}^*\mid\mathbf{X}) &= \frac{(2\pi)^{-(N+1)D/2}}{(2\pi)^{-ND/2}}\frac{Z_{\text{GIW}}(D,\kappa_N+1,\mathbf{S}_{N*},\nu_N+1)}{Z_{\text{GIW}}(D,\kappa_0,\mathbf{S}_0,\nu_0)}\frac{Z_{\text{GIW}}(D,\kappa_0,\nu_0,\mathbf{S}_0)}{Z_{\text{GIW}}(D,\kappa_N,\nu_N,\mathbf{S}_N)} \\ &= (2\pi)^{-D/2}\frac{Z_{\text{GIW}}(D,\kappa_N+1,\nu_N+1,\mathbf{S}_{N*})}{Z_{\text{GIW}}(D,\kappa_N,\nu_N,\mathbf{S}_N)} \\ &= \pi^{-D/2}\frac{(\kappa_N+1)^{D/2}|\mathbf{S}_{N*}|^{-(\nu_N+1)/2}\prod_{i=1}^{D}\Gamma\left(\frac{\nu_N+2-i}{2}\right)}{\kappa_N^{D/2}|\mathbf{S}_N|^{-\nu_N/2}\prod_{i=1}^{D}\Gamma\left(\frac{\nu_N+1-i}{2}\right)} \end{aligned} \tag{22}$$

where we used the form in equation (20) and then substituted equation (13).

An alternative representation shows that the posterior predictive distribution follows a mul-

tivariate Student's t-distribution [Murphy, 2007]:

$$f(\mathbf{x}^* \mid \mathbf{X}) = \mathcal{T}\left(\mathbf{x}^* \mid \mathbf{m}_N, \frac{\kappa_N + 1}{\kappa_N(\nu_N - D + 1)}\mathbf{S}_N, \nu_N - D + 1\right) \tag{23}$$

While in practice, both equations (22) and (23) are important for computing the posterior predictive.

3.1.3 Univariate Gaussian with a Conjugate Prior

Likelihood

Given the random vectors $\mathbf{x} = \{x_1, x_2, \ldots x_N\}$ generated by a multivariate Gaussian distribution with mean μ and variance σ^2, the likelihood is given by:

$$\begin{aligned} f(\mathbf{x} \mid \mu, \sigma^2) &= \prod_{n=1}^{N} \mathcal{N}(x_n \mid \mu, \sigma^2) \\ &= (2\pi\sigma^2)^{-N/2} \exp\left\{ -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n - \mu)\right\} \\ &= \frac{1}{(2\pi)^{N/2}\sigma^N} \exp\left(-\frac{1}{2\sigma^2}[Ns^2 + N(\bar{x} - \mu)^2]\right) \end{aligned} \tag{24}$$

$$f(\mathbf{x} \mid \mu, \sigma^2) = \left(\frac{1}{\sigma^2}\right)^{N/2} \exp\left(-\frac{N}{2\sigma^2}(\bar{x} - \mu)^2\right) \exp\left(-\frac{Ns^2}{2\sigma^2}\right) \tag{25}$$

where

$$\begin{aligned} s^2 &= \frac{1}{n}\sum_{n=1}^{N}(x_n - \bar{x}) \\ \bar{x} &= \frac{1}{N}\sum_{i=1}^{N} x_n \end{aligned}$$

The equivalence of equations (24) and (25) can be explained as,

$$\begin{aligned} \sum_n (x_n - \mu)^2 &= \sum_n [(x_n - \bar{x}) - (\mu - \bar{x})]^2 \\ &= \sum_n (x_i - \bar{x})^2 + \sum_n (\bar{x} - \mu)^2 - 2\sum_n (x_i - \bar{x})(\mu - \bar{x}) \\ &= Ns^2 + N(\bar{x} - \mu)^2 \end{aligned} \tag{26}$$

Prior on parameters

For the mean μ and variance σ^2 of an univariate Gaussian, the normal-inverse-chi-squared prior is fully conjugate:

$$\begin{aligned}\mathcal{NIX}^2(m_0, \kappa_0, \nu_0, S_0) &= \mathcal{N}(\mu \mid m_0, \sigma^2/\kappa_0) \cdot \mathcal{X}^{-2}(\sigma^2 \mid \nu_0, S_0) \\ &= \frac{1}{Z_p(m_0, \kappa_0, \nu_0, S_0)} \sigma^{-1}(\sigma^2)^{-(\nu_0/2+1)} \exp\left(-\frac{1}{2\sigma^2}[S_0 + \kappa_0(\mu_0 - \mu)] \right)\end{aligned} \tag{27}$$

with

$$Z_p(m_0, \kappa_0, \nu_0, s_0^2) = \sqrt{\frac{2\pi}{\kappa_0}} \Gamma(\nu_0/2) \left(\frac{2}{S_0}\right)^{\nu_0/2} \tag{28}$$

In the $\mathcal{NIX}^2(m_0, \kappa_0, \nu_0, S_0)$ distribution, μ_0 is the prior mean and κ_0 is how strongly we believe this; σ_0^2 is the prior variance and ν_0 is how strongly we believe this. Please refer to the Figure 2.

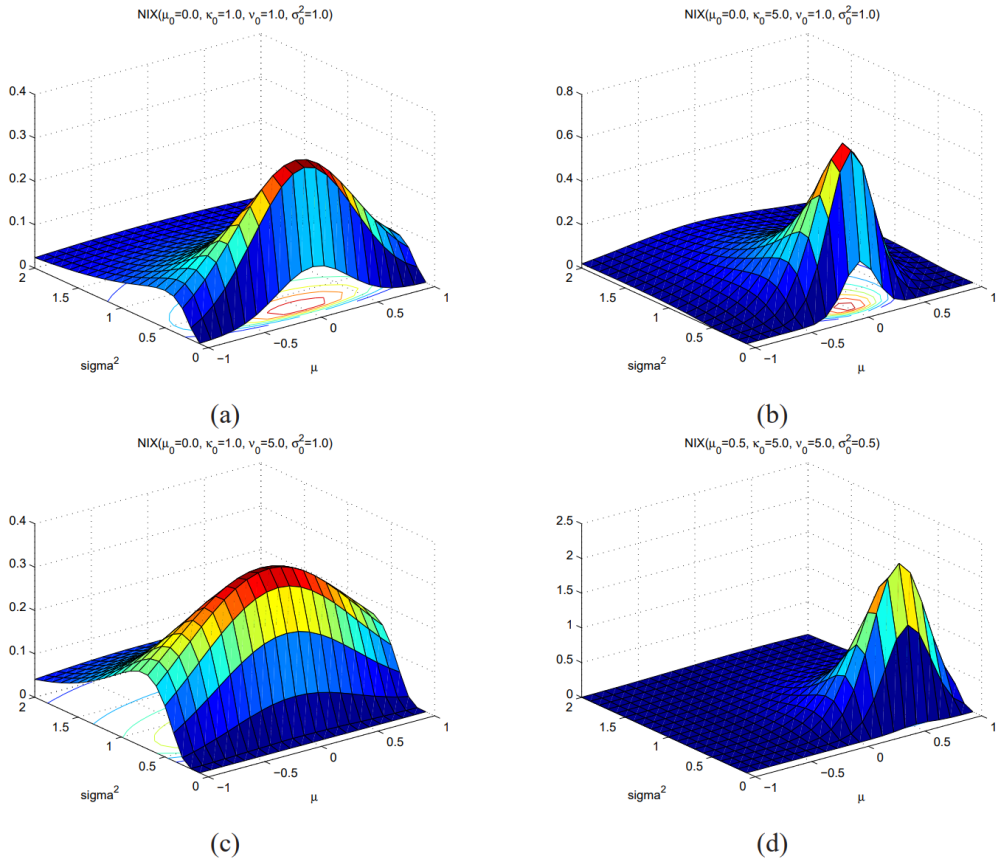


Figure 2: Visualisation of the probability distribution, $\mathcal{NIX}^2$. Notice, how the contour plot is changing the shape for different values of hyperparameters. Figure is taken from [Murphy, 2007]

Posterior on parameters

The posterior distribution of the parameters μ and σ can be expressed as follows:

$$\begin{aligned} f(\mu, \sigma^2 \mid \mathbf{x}) &\propto \mathcal{N}(\mu \mid m_0, \sigma^2/\kappa_0) \cdot \mathcal{X}^{-2}(\sigma^2 \mid \nu_0, S_0/\nu_0) f(\mathbf{x} \mid \mu, \sigma^2) \\ &= \left[\sigma^{-1}(\sigma^2)^{-(\nu_0/2+1)} \exp\left(-\frac{1}{2\sigma^2}[S_0 + \kappa_0(m_0 - \mu)^2]\right)\right] \\ &\qquad \times \left[(\sigma^2)^{-N/2} \exp\left(-\frac{1}{2\sigma^2}[Ns^2 + N(\bar{x} - \mu)^2]\right)\right] \\ &\propto \sigma^{-3}(\sigma^2)^{-(\nu_N/2)} \exp\left(-\frac{1}{2\sigma^2}[S_N + \kappa_N(m_N - \mu)^2]\right) \\ &= \mathcal{NIX}^2(m_N, \kappa_N, \nu_N, \sigma_N^2) \end{aligned} \tag{29}$$

Where matching the powers of σ^2 and rearranging the terms, we got

$$\begin{aligned} \kappa_N &= \kappa_0 + N \\ \nu_N &= \nu_0 + N \\ m_N &= \frac{\kappa_0 m_0 + N\bar{x}}{\kappa_N} \\ S_N &= S_0 + Ns^2 + \kappa_0 m_0^2 + N\bar{x}^2 - \kappa_N m_N^2 \\ &= S_0 + \kappa_0 m_0^2 + \sum_{n=1}^{N} x_i^2 - \kappa_N m_N^2 \end{aligned} \tag{30}$$

Marginal likelihood of data

Repeating the posterior calculations and while maintaining the track of the normalization constants gives the following outcome.

$$\begin{aligned} f(\mathbf{x}) &= \int_{\mu}\int_{\sigma^2} f(\mathbf{x} \mid \mu, \sigma^2) f(\mu, \sigma^2) d\mu d\sigma^2 \\ &= \frac{Z_p(m_N, \kappa_N, \nu_N, S_N)}{Z_p(m_0, \kappa_0, \nu_0, S_0)} \frac{1}{Z_i^N} \\ &= \frac{\sqrt{\kappa_0}}{\sqrt{\kappa_N}} \frac{\Gamma(\nu_N/2)}{\Gamma(\nu_0/2)} \left(\frac{S_0}{2}\right)^{\nu_0/2} \left(\frac{2}{S_N}\right)^{\nu_N/2} \frac{1}{(2\pi)^{(N/2)}} \\ &= \frac{\Gamma(\nu_N/2)}{\Gamma(\nu_0/2)} \sqrt{\frac{\kappa_0}{\kappa_N}} \frac{S_0^{\nu_0/2}}{S_N^{\nu_N/2}} \frac{1}{\pi^{N/2}} \end{aligned} \tag{31}$$

Posterior predictive

To get the posterior predictive, we marginalize out the parameters which simplifies to the ratio of joint likelihood to marginal likelihood.

$$\begin{aligned} f(x \mid \mathbf{x}) &= \int\int f(x \mid \mu, \sigma^2) f(\mu, \sigma^2 \mid \mathbf{x})\, d\mu\, d\sigma^2 \\ &= \frac{f(x, \mathbf{x})}{f(\mathbf{x})} \\ &= \frac{\Gamma((\nu_N + 1)/2)}{\Gamma(\nu_N/2)} \sqrt{\frac{\kappa_N}{\kappa_N + 1}} \frac{S_N^{\nu_N/2}}{(S_N + \frac{\kappa_N}{\kappa_N+1}(x - \mu_N)^2)^{(\nu_N+1)/2}} \frac{1}{\sqrt{\pi}} \\ &= t_{\nu_N}\left(m_N, \frac{(1 + \kappa_N)S_N}{\nu_N \kappa_N}\right) \end{aligned} \tag{32}$$

The probability distribution, t_{ν_N} denotes the Student's t distribution.

3.1.4 Categorical Distribution with a Conjugate Prior

Likelihood

The Likelihood of sample data $\mathbf{c} = \{c_1, c_2, \ldots, c_N\}$, $c_i \in \{1, 2, \ldots, J\}$ begin generated by a categorical distribution which we denote as $\text{Cat}(p_1, p_2, \ldots, p_J)$ assuming each cluster has J categories with there mixing probabilities being $\mathbf{p} = (p_1, p_2, \ldots, p_J)$ is

$$\begin{aligned} f(\mathbf{c} \mid (p_1, p_2, \ldots p_J)) &= \prod_{i=1}^{N} f(c_i \mid p_1, p_2, \ldots, p_J) \\ &= \prod_{i=1}^{N} \prod_{j=1}^{J} p_i^{\mathbb{1}\{c_i = j\}} \end{aligned} \tag{33}$$

Priors on parameters

The mixing probability $\mathbf{p} = (p_1, p_2, \ldots, p_J)$ have Dirichlet prior which is denoted as $\text{Dir}(\gamma)$ where $\gamma_0 = (\gamma_1, \gamma_2, \ldots, \gamma_J)_0$.

$$\begin{aligned} f(p_1, \ldots, p_J) &= Dir(\gamma_0) \\ &= \prod_{j=1}^{J} p_j^{\gamma_{j0}-1} \end{aligned} \tag{34}$$

Posterior on parameters

$$
\begin{aligned}
f(\mathbf{p} \mid \mathbf{c}) &\propto f(\mathbf{c} \mid \mathbf{p}) \dot{f}(\mathbf{p} \mid \mathbf{c}) \\
&\propto \prod_{c_i \in \mathbf{c}} \left(\prod_{j=1}^{J} p_j^{\mathbb{1}\{c_i = j\}} \right) \dot{\prod_{j=1}^{J}} p_j^{\gamma_j - 1} \\
&= \prod_{j=1}^{J} p_j^{\gamma_j - 1 + \Sigma_{c_i \in \mathbf{c}} \mathbb{1}\{c_i = j\}}
\end{aligned}
\tag{35}
$$

By comparing the equations (3.1.4) and (3.1.4), it follows that the posterior is Dirichlet distribution with updated γ,

$$
\gamma_j^{'} = \gamma_j + \sum_{c_i \in \mathbf{c}} \mathbb{1}\{c_i = j\} \tag{36}
$$

Marginal likelihood of data

The marginal likelihood for the categorical data is calculated as,

$$
\begin{aligned}
f(\mathbf{c}) &= \int f(\mathbf{p} \mid \gamma) \cdot \prod_{c_i \in \mathbf{c}} f(c_i \mid \mathbf{p}) \cdot d\mathbf{p} \\
&= \int f(p_1, ..., p_J \mid \gamma_1, ..., \gamma_J) \prod_{c_i \in \mathbf{c}} f(c_i \mid p_1, ..., p_J) d\mathbf{S}_J \\
&= \int \frac{\Gamma(\sum_{j=1}^{J} \gamma_j)}{\prod_{j=1}^{J} \Gamma(\gamma_j)} \prod_{j=1}^{J} p_m^{\gamma_m - 1} \prod_{c_i \in \mathbf{c}} \prod_{j=1}^{J} p_j^{\mathbb{1}\{c_i = j\}} d\mathbf{S}_J \\
&= \frac{\Gamma(\sum_{j=1}^{J} \gamma_j)}{\prod_{j=1}^{J} \Gamma(\gamma_j)} \int \prod_{j=1}^{J} p_j^{\sum_{c_i \in \mathbf{c}} \mathbb{1}\{c_i = j\} + \gamma_j - 1} d\mathbf{S}_J \\
&= \frac{\Gamma(\sum_{j=1}^{J} \gamma_j)}{\prod_{j=1}^{J} \Gamma(\gamma_j)} \frac{\prod_{j=1}^{J} \Gamma(\gamma_j^{'})}{\Gamma(N + \sum_{j=1}^{J} \gamma_j)}
\end{aligned}
\tag{37}
$$

Posterior predictive

In below equations, $d\mathbf{S}_J$ denotes integrating $(p_1, ..., p_J)$ with respect to (J - 1) simplex

$$\begin{aligned} f(c = x \mid \mathbf{c}) &= \int f(c = x \mid \mathbf{p}) \cdot f(\mathbf{p} \mid \mathbf{c}) \cdot d\mathbf{p} \\ &= \int f(c = x \mid p_1, ..., p_J) \cdot f(p_1, ..., p_J \mid \mathbf{c}) \cdot d\mathbf{S}_J \\ &= \int p_x \frac{\Gamma(\sum_{j=1}^J \gamma_j^{'})}{\prod_{j=1}^J \Gamma(\gamma_j^{'})} \prod_{j=1}^J p_j^{\gamma_j^{'}-1} d\mathbf{S}_J \\ &= \frac{\Gamma(\sum_{j=1}^J \gamma_j^{'})}{\prod_{j=1}^J \Gamma(\gamma_j^{'})} \int \prod_{j=1}^J p_j^{\mathbb{1}\{x=j\}+\gamma_j^{'}-1} d\mathbf{S}_J \\ &= \frac{\Gamma(\sum_{j=1}^J \gamma_j^{'})}{\prod_{j=1}^J \Gamma(\gamma_j^{'})} \frac{\prod_{j=1}^J \Gamma(\mathbb{1}\{x = j\} + \gamma_j^{'})}{\Gamma(1 + \sum_{j=1}^J \gamma_j^{'})} \\ &= \frac{\gamma_x^{'}}{\sum_{j=1}^J \gamma_j^{'}} \end{aligned} \tag{38}$$

4 Feature Selection

In various models, it is common practice to utilize all available features for inference. However, in many scenarios, this approach unnecessarily complicates the model by incorporating features that lack significant clustering information and do not contribute to detecting group structure. This can lead to issues such as the *curse of dimensionality* [Bellman 1957], as well as identification problems and over-parameterization [Bartholomew et al. 2011]. Consequently, adopting feature selection techniques becomes crucial, as it aids in streamlining model fitting, enhancing the interpretability of results, and improving the quality of data classification. Even in scenarios with moderate or low dimensionality, reducing the number of features used in clustering can yield notable benefits. This chapter provides a review of the available methods for feature selection in model-based clustering.

During the process of selecting features for clustering, the aim is to only retain those features that are relevant. To achieve this, it's essential to establish clear criteria for determining the significance of a feature as "relevant" or "irrelevant." In a recent study [Ritter 2014], the issue of defining relevance within model-based clustering methods has been discussed. Model-based clustering incorporates the clustering task into a formal modeling framework, where the group structure is embedded in the group membership parameters [McLachlan et al. 1988]. Relevant features contain essential clustering information, with their distribution being directly influenced by the mixture component assignment parameters. Conversely, irrelevant features do not contribute any additional information for clustering purposes. These irrelevant features

can be further classified into redundant and uninformative features. Redundant features are conditionally independent of the clustering parameters given the relevant ones, meaning they do not provide any extra information. In contrast, uninformative features do not possess any discriminative information and are akin to noise; their distribution is completely independent of the clustering structure.

The overall strategy for addressing the problem is determined by how the feature selection methods are applied to the model fitting process. The primary differentiation lies in whether the selection is conducted independently or simultaneously with the learning process. The former case corresponds to filter methods, in which selection occurs either before or after the learning phase. The latter case is known as wrapper methods, where feature selection and learning are integrated, essentially "wrapping" the selection methods around the learning phase. Filter approaches are known for their simplicity and computational efficiency. Conversely, wrapper methods often yield superior outcomes, despite with greater complexity [Guyon et al. 2003]. Recently, wrapper methods have garnered significant attention and adoption due to their ability to seamlessly integrate into the model learning process, potentially resulting in improved clustering performance.

However, in our model, we will be using wrapper methods. Within wrapper methods, the following approaches are introduced: Penalization approach, Bayesian approach, and Model selection approach.

4.1 Penalization Approaches

In this context, a penalization term is introduced on the model parameters, and feature selection is performed by inducing sparsity in the estimates. The objective is to optimize a penalized version of the log-likelihood within a mixture model while eliminating features whose parameter estimates either shrink to zero or converge to a uniform value across the mixture components. The log-likelihood with penalization takes the general form:

$$\ell_Q = \sum_{i=1}^{N} \log \left\{ \sum_{k=1}^{K} \tau_g \phi(\mathbf{x}_i; \Theta_k) \right\} - Q_\lambda(\Theta), \tag{39}$$

where $Q_\lambda(\Theta)$ is a function of the probability densities parameters Θ and the penalty parameter λ.

Pioneering research in this category of methodologies includes the technique first introduced by [Pan et al.], which uses an ℓ_1 penalty function. [Bhattacharya et al. 2014] proposed a related approach that considers the size of the mixture component. [Wanf and Zhu 2008] replaced the ℓ_1 norm for the ℓ_∞ norm, treating the parameters corresponding to the same feature as a

group. But unlike our approach, these all approaches heavily depends on BIC [53] to find the true number of clusters.

[Guo et al. 2010] proposed a pairwise fusion penalty to identify features discriminative for specific pairs of mixture components. [Xie et al. 2008] and [Zhou et al.] expanded the penalization approach to Gaussian mixture models with covariance matrices specific to each cluster, where the former follows a diagonal structure and the latter one follows unconstrained nature.

A different method is the sparse k-means proposed by [Witten et al. 2010], which solves an optimization problem involving an ℓ_1 penalty on the variable weights. The weight w_d represents the contribution of feature X_d to the resulting clustering, with $w_d = 0$ indicating that the variable does not contribute.

Another distinct method is the sparse k-means introduced by [Witten et al. 2010], which tackles an optimization problem incorporating an ℓ_1 penalty on the variable weights. In the resultant clustering, the contribution of feature X_d is weighted by w_d. If the feature does not contribute, it implies $w_j = 0$.

The maximization of the penalized likelihood typically involves employing a penalized EM algorithm, which requires pre-specifying the total number of clusters. Model selection involves determining the optimal shrinkage parameter λ and the total number of mixture components. Many models within this framework are deterministic, highly specialized, and not readily adaptable to various distributions. Furthermore, they heavily rely on model selection criteria for choosing the optimal penalty. In contrast, our model utilizes MCMC sampling methods to maximize penalized likelihood that converges to the optimal clustering.

4.2 Bayesian approaches

Developments within the Bayesian framework have led to various methods for simultaneous feature selection and model-based clustering. Central to these methods is the introduction of a variable ϕ, typically following a distribution $f(\phi)$. This variable partitions $\mathbf{X}$ into two subsets: $\mathbf{X}_C$, comprising features defining a mixture distribution, and $\mathbf{X}_{NC}$, representing features indicating a mixture component. The conditional distribution of the data given $\mathbf{z}$ is expressed as:

$$f(\mathbf{X}|\mathbf{z}, \Omega, \phi) = f(\mathbf{X}_C|\mathbf{z}, \Theta, \beta) f(\mathbf{X}_{NC}|\beta_0, \phi), \tag{40}$$

with $f(\mathbf{X}_C|\mathbf{z}, \beta, \Phi)$ denoting distribution of a mixture component with parameters β, $f(\mathbf{X}_{NC}|\beta_0, \Phi)$ represents a probability distribution with parameters β_0, and Ω encompasses all parameters, denoted as β, β_0. Subsequently, the focus shifts to drawing inference from the posterior distri-

bution with the goal of feature selection and clustering:

$$f(\mathbf{z}, \phi, \beta, \beta_0|\mathbf{X}) \propto f(\mathbf{X}|\mathbf{z}, \phi, \beta, \beta_0) f(\mathbf{z}|\beta, \beta_0) f(\phi) f(\beta, \beta_0). \tag{41}$$

The anchor mode model proposed by [Liu et al. 2003] selects the most informative principal components (or factors) of the data for feature selection. Assuming that only a subset of these components is informative for clustering and distributed according to a mixture of Gaussians, while the remaining components follow a simple Gaussian distribution, inference on the number of relevant factors is conducted using a Markov Chain Monte-Carlo (MCMC) scheme. However, it's important to note that in principal component analysis, we assess the weights of features in their linear combination rather than directly sampling from the features space.

Instead of adopting the "hard selection" method, [law et al. 2004] present a Bayesian framework that incorporates the notion of feature saliency. As introduced initially, let $\phi = (\phi_1, \ldots, \phi_d, \ldots, \phi_D)$ denote a binary variable, where $\phi_d = 1$ indicates that $\mathbf{X}_d$ is relevant, otherwise $\phi_d = 0$. Thus, the saliency of feature $\mathbf{X}_d$ represents its importance in characterizing the cluster structure of the data, computed as $f(\phi_d = 1)$. Employing a minimum message length [Baxter 2002] criterion, the authors implement an EM algorithm for maximum a posteriori (MAP) estimation of the saliences.

Assuming a prior distribution on ϕ, as formulated by [Tadesse et al. 2005]:

$$f(\phi|\eta) = \prod_{d=1}^{D} \eta^{\phi_d} (1-\eta)^{1-\phi_d}, \tag{42}$$

interpreting η as the hyperparameter denoting the expected proportion of discriminative features among the groups, inference is conducted using an MCMC scheme, where the vector ϕ undergoes updates via a Metropolis search. Posterior inference on ϕ is carried out subsequent to parameter integration, focusing on the marginal posterior $f(\phi_d = 1|\mathbf{X})$. Identifying the most significant clustering features involves examining those with the highest marginal posterior, $f(\phi_d = 1|\mathbf{X}) > t$, employing a specified threshold t. Alternatively, selection may involve considering the entire vector ϕ with the greatest posterior probability among all vectors visited throughout the chain. In this method, particularly for high-dimensional spaces or complex distributions, Metropolis sampling can require significant computational time to explore the parameter space. In contrast, our approach employs collapsed Gibbs sampling (see Section 2.2.2), capitalizing on the conjugacy within the Bayesian framework to sample a single parameter at a time, enhancing computational efficiency.

The method introduced by [Kim et al. 2006] broadens its scope by presenting clustering as an infinite mixture of Gaussian distributions using Dirichlet process mixtures. Conversely,

[Swartz et al. 2008] expands the approach to model data characterized by a predetermined structure imposed from an experimental design.

Our novel approach offers a flexible and principled framework for simultaneous clustering and feature selection, allowing for the incorporation of prior knowledge and providing a natural way to handle uncertainty in the feature selection process. However, the computational complexity of MCMC-based methods may be a limitation, especially for large-scale datasets. Additionally, the performance of these methods could depend on prior distributions and the convergence of the MCMC sampler.

4.3 Model Selection Approaches

In this category of approaches, the task of selecting relevant features is reformulated as a model selection challenge. According to the clustering assignments, $\mathbf{z}$, various models are specified with respect to their role assigned to the features. Using a model selection criterion, these models are compared to get the optimal model and according to that to get the relevant features.

[Raftery et al. 2006] introduced this framework, where $\mathbf{X}$ is divided according to the three roles as given below:

- $\mathbf{X}_C$, represents the current features;
- $\mathbf{X}_P$, denotes the feature(s) proposed for addition or removal from the feature set;
- $\mathbf{X}_{NC}$, signifies the set of irrelevant features for clustering.

Then, the decision for inclusion or exclusion of $\mathbf{X}_P$ is taken by comparing the models $\mathcal{M}_A$ and $\mathcal{M}_B$ using the BIC approximation to their marginal likelihoods.

Suppose we are given with the two models, $\mathcal{M}_A$ and $\mathcal{M}_B$, the decision of $\mathbf{X}_P$ to be added to $\mathbf{X}_C$ or $\mathbf{X}_{NC}$ depends on the BIC approximation to their marginal likelihoods.

[Maugis et al. 2008] extended this framework by allowing $\mathbf{X}_P$ to be related only to a subset $\mathbf{X}_R \subseteq \mathbf{X}_C$ of the features in the conditional distribution $f(\mathbf{X}_P|\mathbf{X}_C)$. They compared models $\mathcal{M}_A$ and $\mathcal{M}_C$ using the BIC, where $\mathcal{M}_C$ incorporates the regression term $f(\mathbf{X}_P|\mathbf{X}_R \subseteq \mathbf{X}_C)$.

[Maugis et al. 2009] further expanded the framework by considering an additional role for $\mathbf{X}_P$, explicitly accounting for the case where it can be independent of $\mathbf{X}_C$. They proposed comparing three models ($\mathcal{M}_A$, $\mathcal{M}_C$, and $\mathcal{M}_D$) using the BIC, where $\mathcal{M}_D$ incorporates the density $f(\mathbf{X}_P)$ of a multivariate Gaussian distribution.

[Marbac and Sedki 2017] proposed a procedure that relies on the ICL criterion and does not require multiple calls of the EM algorithm. Their approach involves finding the binary vector

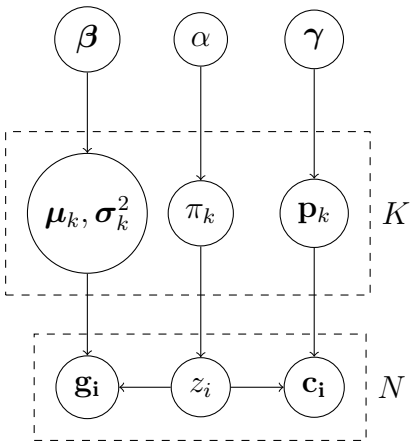


Figure 3: Hierarchical mixture model with heterogeneous data, Gaussian and categorical.

ϕ that maximizes the Maximum Integrated Complete-data Likelihood (MICL) criterion, where $\phi_d = 1$ if variable $\mathbf{X}_d$ is relevant, and 0 otherwise.

5 The Model

5.1 Clustering Heterogeneous Data

We have N samples, with D number of features, where each sample belongs to one of the K component. Now, we have two types of features, Gaussian and categorical with total number of Gaussian features being l and total number of categorical features being m. Total number of categories in categorical data is J. So, $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ we have $\mathbf{x}_i = \{g_1, ..., g_l, c_1, ..., c_m\}_i$ sampled i.i.d., from a finite mixture distribution with density

$$f(\mathbf{x} \mid \pi, \mu, \sigma^2, \mathbf{P}) = \sum_{k=1}^{K} \pi_k \cdot \prod_{j=1}^{l} \mathcal{N}(\mathbf{g}_j \mid \mu_{kj}, \sigma_{kj}^2) \cdot \prod_{j=1}^{m} \text{Cat}(\mathbf{c}_j \mid \mathbf{p}_{kj}) \tag{43}$$

5.1.1 The Bayesian framework

We define a Dirichlet prior α over component mixing probabilities. A Normal-Inverse-Chi-Squared ($\mathcal{NI}\chi^2$) prior with hyperparameters $\beta_0 = (m_0, \kappa_0, \nu_0, S_0)$ is defined over the mean μ and variance σ^2 of the Gaussian feature of a component. A Dirichlet prior $\gamma_0 = (\gamma_{0,1}, \ldots, \gamma_{0,J})$ is defined over the probability vector $\mathbf{P} = (p_1, p_2, \ldots, p_J)$. Refer to Figure 3 for the graphical representation of the framework.

$$
\begin{aligned}
f(\pi \mid \alpha) &= Dir(\pi \mid \alpha) \\
f(\mu, \sigma^2 \mid \beta_0) &= \mathcal{NI}\chi^{\mathbf{2}}(\mu, \sigma^2 \mid m_0, \kappa_0, \nu_0, s_0) \\
f(\mathbf{p} \mid \gamma_0) &= Dir(\mathbf{p} \mid \gamma_{0,1}, \ldots, \gamma_{0,J})
\end{aligned}
$$

As per the caclculations in (29), posterior of the μ and σ^2 parameters under $\mathcal{NI}\chi^{\mathbf{2}}$ is given as,

$$
f(\mu, \sigma^2 \mid \mathbf{x}, \beta) = \mathcal{NI}\chi^{\mathbf{2}}(\mu, \sigma^2 \mid m_N, \kappa_N, s_N, \nu_N), \tag{44}
$$

where,

$$
\begin{aligned}
m_N &= \frac{n\bar{x} + m_0\kappa_0}{n + \kappa_0} \\
\kappa_N &= \kappa_0 + n \\
s_N &= s_0 + \kappa_0 m_0^2 + \sum_{i=1}^{n} x_i^2 - \kappa_N m_N^2 \\
\nu_N &= \nu_0 + n
\end{aligned} \tag{45}
$$

Similarly, posterior of $\mathbf{p}$ under Dirichlet prior is,

$$
f(\mathbf{p} \mid \mathbf{c}) = \text{Dir}(\mathbf{p} \mid \gamma_1, ..., \gamma_D), \tag{46}
$$

where,

$$
\gamma_m = \gamma_0 + \sum_{c_i \in \mathbf{c}} \mathbb{1}\{c_i = m\} \tag{47}
$$

with total J categories and $c_i \in \{1, ..., J\}$.

5.1.2 Collapsed Gibbs sampling algorithm

For clustering the data, we use collapsed Gibbs sampling techniques for the reasons discussed in Section 2.2.2. Here, we choose $f(\pi \mid \alpha)$, $f(\mu, \sigma^2 \mid \beta)$, and $f(\mathbf{P} \mid \gamma)$ to be conjugate. We will only sample $\mathbf{z}$ analytically, integrating out the model parameters. This means we aim to maximize the posterior marginal probability.

$$
f(\mathbf{z} \mid \mathbf{X}) \propto f(\mathbf{X} \mid \mathbf{z}) \times f(\mathbf{z}) \tag{48}
$$

The α hyperparameter is set to be α/K (known as *symmetrical Dirichlet prior*) representing a lack of prior information or belief in any particular distribution of probabilities. The collapsed Gibbs sampling is done as,

$$\begin{aligned}
f(z_i = k \mid \mathbf{z}_{[-i]}, \mathbf{X}, \mathcal{H}) &\propto f(\mathbf{X} \mid \mathbf{z}_{[-i]}, z_i = k, \mathcal{H}) \times f(z_i = k \mid \mathbf{z}_{[-i]}, \mathcal{H}) \\
&= f(\mathbf{X} \mid \mathbf{z}_{[-i]}, z_i = k, \cancel{\alpha}, \beta, \gamma) \times f(z_i = k \mid \mathbf{z}_{[-i]}, \alpha, \cancel{\beta, \gamma}) \\
&= f(\mathbf{X} \mid \mathbf{z}_{[-i]}, z_i = k, \beta, \gamma) \times f(z_i = k \mid \mathbf{z}_{[-i]}, \alpha) \\
&= f(\mathbf{x}_i \mid \mathbf{z}_{[-i]}, z_i = k, \beta, \gamma) \times f(\mathbf{X}_{[-i]} \cancel{\mid z_i = k}, \mathbf{z}_{[-i]}, \beta, \gamma) \times f(z_i = k \mid \mathbf{z}_{[-i]}, \alpha) \\
&\propto f(\mathbf{g}_i, \mathbf{c}_i \mid \mathbf{X}_{[-i]}, \mathbf{z}_{[-i]}, z_i = k, \beta, \gamma) \times f(z_i = k \mid \mathbf{z}_{[-i]}, \alpha) \\
&= f(\mathbf{g}_i \mid \mathbf{X}_{[-i]}, z_i = k, \mathbf{z}_{[-i]}, \beta) \times f(\mathbf{c}_i \mid \mathbf{z}_{[-i]}, z_i = k, \gamma) \times f(z_i = k \mid \mathbf{z}_{[-i]}, \alpha)
\end{aligned} \tag{49}$$

Third term

The expression for $f(z_i = k \mid \mathbf{z}_{[-i]}, \alpha)$ can be written as,

$$f(z_i = k \mid \mathbf{z}_{[-i]}, \alpha) = \frac{f(z_i = k, \mathbf{z}_{[-i]} \mid \alpha)}{f(\mathbf{z}_{[-i]} \mid \alpha)} = \frac{f(\mathbf{z} \mid \alpha)}{f(\mathbf{z}_{[-i]} \mid \alpha)} \tag{50}$$

It will be easy to find both numerator and denominator in above expression, if we can find an expression for $f(\mathbf{z} \mid \alpha)$.

We proceed by marginalizing out π,

$$\begin{aligned}
f(\mathbf{z} \mid \alpha) &= \int_\pi f(\mathbf{z} \mid \pi) f(\pi \mid \alpha) d\pi \\
&= \int_\pi \prod_{k=1}^{K} \pi_k^{N_k} \frac{1}{B(\alpha)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} d\pi \\
&= \frac{1}{B(\alpha)} \int_\alpha \prod_{k=1}^{K} \pi_k^{N_k + \alpha_k - 1} d\pi \\
&= \frac{\Gamma(\alpha_0)}{\Gamma(N + \alpha_0)} \prod_{k=1}^{K} \frac{\Gamma(N_k + \alpha_k)}{\Gamma(\alpha_k)}
\end{aligned} \tag{51}$$

Now,

$$
\begin{aligned}
f(z_i = k \mid \mathbf{z}_{[-i]}, \alpha) &= \frac{f(z_i = k, \mathbf{z}_{[-i]})}{f(\mathbf{z}_{[-i]} \mid \alpha)} = \frac{f(\mathbf{z} \mid \alpha)}{f(\mathbf{z}_{[-i]} \mid \alpha)} \\
&= \frac{\frac{\Gamma(\alpha_0)}{\Gamma(N+\alpha_0)} \frac{\Gamma(N_k+\alpha_k)}{\Gamma(\alpha_k)} \prod_{j=1, j \ll k}^{K} \frac{\Gamma(N_j)+\alpha_j}{\Gamma(\alpha_j)}}{\frac{\Gamma(\alpha_0)}{\Gamma(N+\alpha_0-1)} \frac{\Gamma(N_{k[-i]}+\alpha_k)}{\Gamma(\alpha_k)} \prod_{j=1, j \ll k}^{K} \frac{\Gamma(N_j)+\alpha_j}{\Gamma(\alpha_j)}} \\
&= \frac{\Gamma(N + \alpha_0 - 1)}{\Gamma(N + \alpha_0)} \frac{N_k + \alpha_k}{N_{k[-i]} - \alpha_k} \\
&= \frac{N_{k[-i]} + \alpha_k}{N + \alpha_0 - 1} \\
&= \frac{N_{k[-i]} + \alpha/K}{N + \alpha - 1}
\end{aligned}
\tag{52}
$$

First and Second term

In general these terms can be written as $f(\mathbf{g}_i \mid \mathbf{X}_{[-i]}, z_i = k, \mathbf{z}_{[-i]}, \beta) \times f(\mathbf{c}_i \mid z_i = k, \mathbf{z}_{[-i]}, \gamma) = f(\mathbf{x}_i \mid \mathbf{X}_{[-i]}, z_i = k, \mathbf{z}_{[-i]}, \mathcal{H}) = f(\mathbf{x}_i \mid \mathbf{X}_{[-i],k})$.

where $\mathbf{X}_{[-i],k} = \{\mathbf{x}_j : z_j = k, j \neq i\}$ denotes all the data assigned to the mixture component k excluding $\mathbf{x}_i$. Employing a conjugate prior for $\{\mu_k, \sigma_k^2, \mathbf{P}\}$ enables the computation of $f(\mathbf{x}_i \mid \mathbf{X}_{[-i],k})$ in a closed form. These expressions correspond precisely to the posterior predictives calculated in (3.1.3) and (3.1.4). So, to get the above expression, we evaluate, $\mathbf{X}_i = \{\mathbf{g}_i, \mathbf{c}_i\}$ under the posterior predictive values of each cluster by excluding $\mathbf{X}_i$'s statistics from its current cluster, z_i, and then evaluating $\mathbf{X}_i$ under each cluster's posterior predictive.

5.1.3 Model Selection

Typically, we run the model using multiple different initial assignments, $\mathbf{z}$ and various values for the number of clusters, denoted as K. Unlike maximum likelihood approaches such as the EM algorithm, we are not required to predefine K in this model. It tends to converge towards the optimal number of clusters by emptying out the excess clusters. We only need to provide it with the maximum number of clusters. However, depending on the initialization, it may converge to different values of K. Therefore, we employ the Bayesian Information Criterion (BIC) to select the most suitable model

BIC $= q \ln(N) - 2 \ln(\widehat{L})$. Where,

- q = Number of parameters estimated by the model
- $\widehat{L}$ = Posterior marginal likelihood

$$\begin{aligned} \text{BIC} &= q\ln(N) - 2\ln(\widehat{L}) \\ &= (2l + Jm) \cdot K \cdot ln(N) - 2 \cdot ln(f(\mathbf{z} \mid \mathbf{X})) \end{aligned} \tag{53}$$

Mainly BIC penalises the number of parameters and supports the likelihood. So, lower BIC is preferred.

Algorithm 1: Collapsed Gibbs Sampler for the mixture model

Input: Choose an initial z;
for *T iterations* **do**
  for $i \leftarrow 1$ **to** N **do**
    Remove x_i's statistics from component z_i;
    for $k \leftarrow 1$ **to** K **do**
      Calculate $f(z_i = k \mid \mathbf{z}_{[-i]}, \mathbf{X}, \mathcal{H})$ from 49 ;
    end
    Sample k_{new} from $f(z_i = k \mid \mathbf{z}_{[-i]}, \mathbf{X}, \mathcal{H})$ with Gumbel max-trick;
    Update changed clusters' statistics after assigning $z_i = k_{\text{new}}$;
  end
  Calculate $f(\mathbf{z} \mid \mathbf{X})$;
end

5.2 Clustering Heterogeneous Data with Feature Selection

In this section, we present our approach, which shares certain similarities and have some trade-offs with methodologies discussed in Sections 4.2 and 4.1. We define a matrix denoted as $\mathbf{\Phi}$. For the k-th cluster and d-th feature, Φ_{kd} takes the value 1 if the d-th feature is important for the k-th cluster; otherwise, Φ_{kd} is set to 0. For a general model, $\mathbf{x}$, and hyperparameters set, $\mathcal{H}$, probabilities $f(\Phi_{kd} = 0)$ and $f(\Phi_{kd} = 1)$ are calculated as follows.

The posterior predictive step of the collapsed Gibbs sampling algorithm will be modified as:

$$\begin{aligned} f(\mathbf{x}_i \mid \mathbf{X}_{[-i]}, z_i = k, \mathbf{z}_{[-i]}, \mathcal{H}) = \prod_{d=1}^{D} &\left\{ f(\mathbf{x}_{id} \mid \mathbf{X}_{[-i]}, z_i = k, \mathbf{z}_{[-i]}, \mathcal{H}_{kd}) \right\}^{\mathbb{1}\{\Phi_{kd}=1\}} \\ &\times \left\{ f(\mathbf{x}_{id} \mid \mathbf{X}_{[-i]}, z_i = k, \mathbf{z}_{[-i]}, \mathcal{H}_{0d}) \right\}^{\mathbb{1}\{\Phi_{kd}=0\}} \end{aligned} \tag{54}$$

Let $\mathcal{H}$ denote the set of all hyperparameters, $\mathcal{H} = \{\beta, \gamma\}$. $\mathcal{H}_{kd}$ denotes the set of hyperparameters for the k-th cluster and the d-th feature, while $\mathcal{H}_{0d}$ represents the hyperparameters of

all unimportant data of the d-th feature. Now, we will see how we calculate it.

After, each iteration, for each cluster k and feature d, we sample Φ_{kd}.

$$f(\Phi_{kd} = 1 \mid \mathbf{X}, \mathcal{H}, \mathbf{z}) \propto f(\mathbf{X} \mid \mathbf{z}, \Phi_{kd} = 1) \times f(\phi_{kd} = 1 \mid \mathbf{z}) \tag{55}$$

Similarly,

$$f(\Phi_{kd} = 0 \mid \mathbf{X}, \mathcal{H}, \mathbf{z}) \propto f(\mathbf{X} \mid \mathbf{z}, \Phi_{kd} = 0) \times f(\phi_{kd} = 0 \mid \mathbf{z}) \tag{56}$$

Probabilities $f(\phi_{kd} = 1 \mid \mathbf{z})$ and $f(\phi_{kd} = 0 \mid \mathbf{z})$ are the prior probabilities as they are only conditional on $\mathbf{z}$.

We define,

$$\mathbf{X}_{Cd} = \bigcup_{c \in C} \mathbf{X}_{cj}, \text{ with } C = \{c : \phi_{cd} = 0, c \neq k, \forall c \in \{1, ..., K\}\}$$
$$\mathbf{X}_{C_0 d} = \bigcup_{c \in C_0} \mathbf{X}_{cj}, \text{ with } C_0 = \{k\} \cup \{c : \phi_{cd} = 0, \forall c \in \{1, ..., K\}\}$$

Now, equations (55) and (56) can be written as,

$$\begin{aligned} f(\Phi_{kd} = 1 \mid \ldots) &\propto f(\mathbf{X} \mid \mathbf{z}, \Phi_{kd} = 1) \\ &= f(\mathbf{X}_{kd} \mid \mathcal{H}_{kd}) \times f(\mathbf{X}_{Cd} \mid \mathcal{H}_{Cd}) \end{aligned} \tag{57}$$

$$\begin{aligned} f(\Phi_{kd} = 0 \mid \ldots) &\propto f(\mathbf{X} \mid \mathbf{z}, \Phi_{kd} = 0) \\ &= f(\mathbf{X}_{C_0 d} \mid \mathcal{H}_{0d}) \end{aligned} \tag{58}$$

Expressions (59) and (60) are then simplified by using the marginal probability expression given in (31) (3.1.4). To keep the number of relevant features low we penalize these probabilities by introducing a λ penalty, $e^{-\lambda \sum_{k=1}^{K} \sum_{d=1}^{D} \Phi_{kd}}$. This is a regularization term, which will be multiplied to the equations, $f(\Phi_{kd} = 1)$ and $f(\Phi_{kd} = 0)$. Appropriate value of λ is obtained by using Bayesian Inference Criterion (BIC)[53].

So, the equations (59) and (60) are updated as,

$$f(\Phi_{kd} = 1 \mid \ldots) \propto f(\mathbf{X}_{kd} \mid \mathcal{H}_{kd}) \times f(\mathbf{X}_{Cd} \mid \mathcal{H}_{Cd}) \times \exp^{-\lambda \sum_{k=1}^{K} \sum_{d=1}^{D} \Phi_{kd}} \tag{59}$$

$$f(\Phi_{kd} = 0 \mid \ldots) \propto f(\mathbf{X}_{C_0 d} \mid \mathcal{H}_{0d}) \times \exp^{-\lambda \sum_{k=1}^{K} \sum_{d=1}^{D} \Phi_{kd}} \tag{60}$$

Algorithm 2: Collapsed Gibbs Sampler for a mixture model with feature selection

Input: Choose an initial z;
for *T iterations* **do**
  for $i \leftarrow 1$ **to** N **do**
    Remove x_i's statistics from component z_i;
    for $k \leftarrow 1$ **to** K **do**
      Calculate $f(\mathbf{x}_i \mid \mathbf{X}_{[-i]}, z_i = k, \mathbf{z}_{[-i]}, \mathcal{H})$;
      Get $f(z_i = k \mid \mathbf{z}_{[-i]}, \mathbf{X}, \mathcal{H})$ by multiplying the above equation with equation (52);
    end
    Sample k_{new} from $f(z_i = k \mid \mathbf{z}_{[-i]}, \mathbf{X}, \langle)$ with Gumbel max-trick;
    Update changed clusters' statistics after assigning $z_i = k_{\text{new}}$;
  end
  Calculate $f(\phi_{kj} = 0)$ and $f(\phi_{kj} = 1)$;
  Sample ϕ_{kj} ;
  Calculate $f(\mathbf{z} \mid \mathbf{X})$;
end

Algorithm and Implementation

Refer to the Algorithm 1 for the collapsed Gibbs Sampling for the mixture model without feature selection and Algorithm 2 with feature selection. Both algorithms have a time complexity of $\mathcal{O}(T \cdot N \cdot K \cdot D)$, where T is the number of iterations, N is the number of samples, K is the number of clusters, and D is the number of features.

We used some methods and tricks for the efficient and optimal implementations, which are given below:

Gumbel-max: While in implementing, all probability functions are computed in logarithmic spaces to prevent underflow errors. This method is employed for efficient sampling from non-normalized log probabilities, ensuring robustness against underflow issues. It exploits the property that if $X_1, X_2, \ldots, X_n$ are independent and identically distributed Gumbel random variables, then for any set of real numbers $x_1, x_2, \ldots, x_n$, the variable $Y = \arg\max_i(x_i + X_i)$

has the same distribution as $\arg\max_i x_i$.

Rank one update in Cholesky decomposition: Efficient computation of the determinant of the full covariance matrix in the Wishart prior relies on employing Cholesky decomposition. Thus, when updating the covariance matrix with a rank-one update, expressed as $S_0 = S + xx^T$ [Seeger, 2004], leveraging the previously computed Cholesky factor L of S allows for the efficient computation of the Cholesky factor L_0 of S_0. Notably, S_0 differs from S solely by three symmetric rank-one matrices. Consequently, L_0 can be computed from L using three rank-one Cholesky updates, each requiring $O(D^2)$ operations, thereby providing savings from $O(D^3)$ compared to recomputing L, the Cholesky decomposition of S, from scratch.

We predefine the number of iterations, but sampling is terminated before reaching the total number of iterations if the posterior marginal likelihood equation (48) converges. Convergence is determined by monitoring the relative change in the likelihood function over the last 5 iterations, where convergence is achieved if for $\left|1 - \frac{f_i(\mathbf{z}|\mathbf{X})}{f_j(\mathbf{z}|\mathbf{X})}\right| < \delta \quad \forall\{i,j\} \in \{1,\ldots,5\}$, with δ being a predefined threshold. By default, δ is set to 10^{-4}.

Greedy run: Following the sampling process, an additional iteration of the model was conducted using the maximum a posteriori (MAP) parameters as an alternative initialization. This approach aimed to refine the model further and enhance its performance.

Additionally, we have enhanced the efficiency of updating predictive likelihoods, marginals by storing and reusing the sufficient statistics for each cluster.

The algorithm was implemented and benchmarked on an Intel Xeon Silver 4210R CPU with 40 cores (10 cores per socket, 2 sockets).

Code availability

The implementation of the algorithms is available on:https://github.com/SamarthPardhi/mixture-models under the MIT licence. They are implemented in Python.

6 Synthetic Data and Results

We compare this Bayesian approach with an alternative frequentist approach where the mixture models are estimated into a Maximum Likelihood Estimator(MLE) with Expectation-Maximization (EM). The model parameters $(\alpha, \mu, \sigma^2, \gamma)$ are estimated maximising mixture model's likelihood. EM-algorithms has its own advantages like it can deal with different, size, shape and orientations. Unlike Bayesian methods, it may not eventually reach to the target distribution [Fraley and Raftery, 2007]. We use Scikit-learn's GMM package [Scikit-learn] for this purpose.

Initially, we present clustering results obtained from synthetically generated heterogeneous data without employing feature selection. Later, we move to analyzing large datasets with and without the application of feature selection (FS). We also compare them with some existing methods in the literature.

6.1 Clustering Heterogeneous Data Without Feature Selection

We generate univariate Gaussian data by uniformly sampling hyperparameters. For example, to compare our model with existing methods, we fix the number of samples, $N = 1000$, the number of Gaussian features, $l = 10$, and the number of clusters, $K = 3$. Then, for each cluster k and feature d, we uniformly sample hyperparameters $\beta = (m_0, \kappa_0, S_0, \nu_0)$ as follows:

$$
\begin{aligned}
m_0 &\sim \text{Uni}(2, 4) \\
\kappa_0 &\sim \text{Uni}(3, 5) \\
S_0 &\sim \text{Uni}(0, 0.1) \\
\nu_0 &\sim \text{Uni}(4, 8)
\end{aligned}
$$

Refer to the Section 1.1.4 to get the intuition of the hyperparameters. Subsequently, mean and variance are sampled from the distribution $\mathcal{NI}\chi^{\mathbf{2}}(\beta)$. For the k-th cluster and d-th feature, Gaussian data are generated using μ_k and σ_k^2.

Similarly, categorical data for J categories are generated by uniformly sampling through hyperparameter γ_0 for each cluster, where $\gamma_0 \sim \text{Uni}(0.15, 0.45)$. Then, the $\mathbf{P}$ vector is sampled from $\text{Dir}(\boldsymbol{\gamma_0})$, and the corresponding categorical data are generated for the specific cluster and feature. Later, the cluster assignments, $\mathbf{z}$ is sampled uniformly for the given $K = 3$ and $N = 1000$, and the data are assigned accordingly. These are our standard methods for generating data, same methods are used for generating other similar data in subsequent methods.

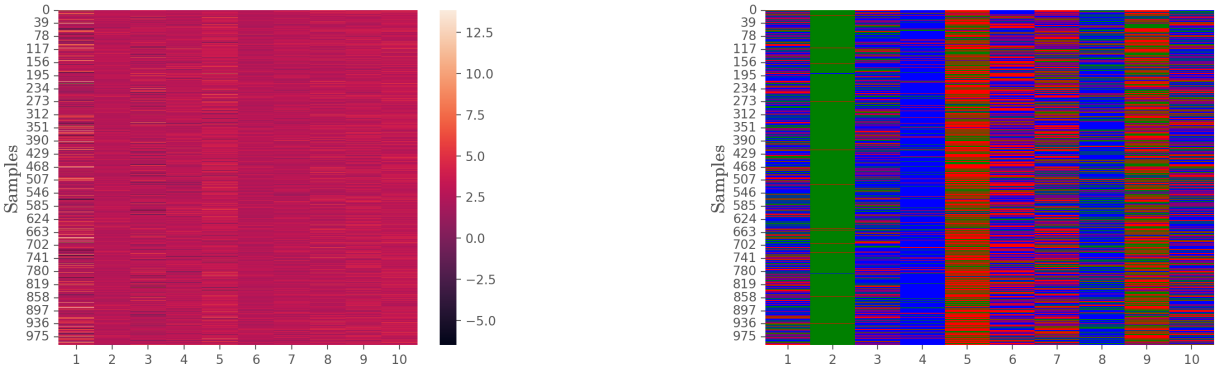


Figure 4: The heatmap of the mixed heterogeneous data with $N = 1000$ number of samples, $K = 3$ number of components, 10 Gaussian features, and 10 categorical features. Each categorical feature has maximum 3 number of categories and each color represent a category: ■ represents category 0, ■ represents category 1, ■ represents category 2.

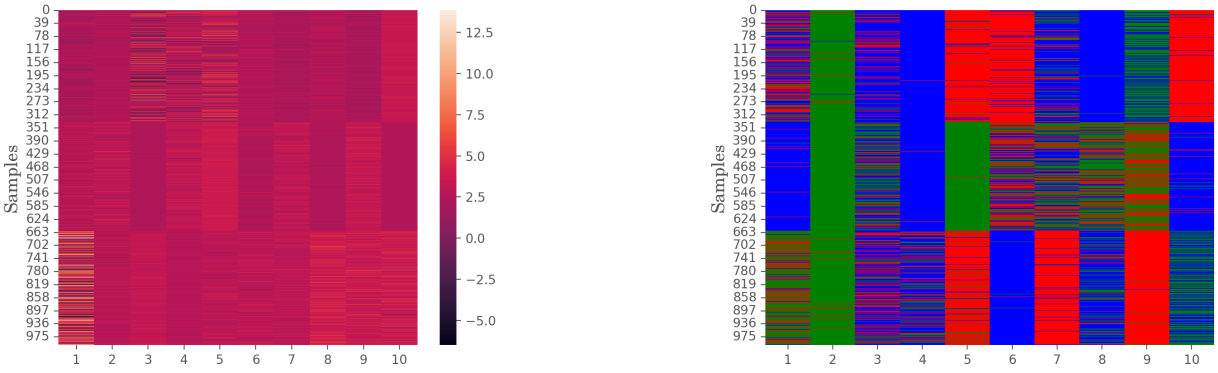


Figure 5: Sorted data given Figure 4 according to true cluster assignments.

On the heterogeneous dataset, we executed our model both without feature selection, alongside a comparison with the EM algorithm. Further detailed findings are depicted in Figure 6. The comparison involved using the Scikit-learn GMM implementation, with outcomes illustrated in Figure 7.

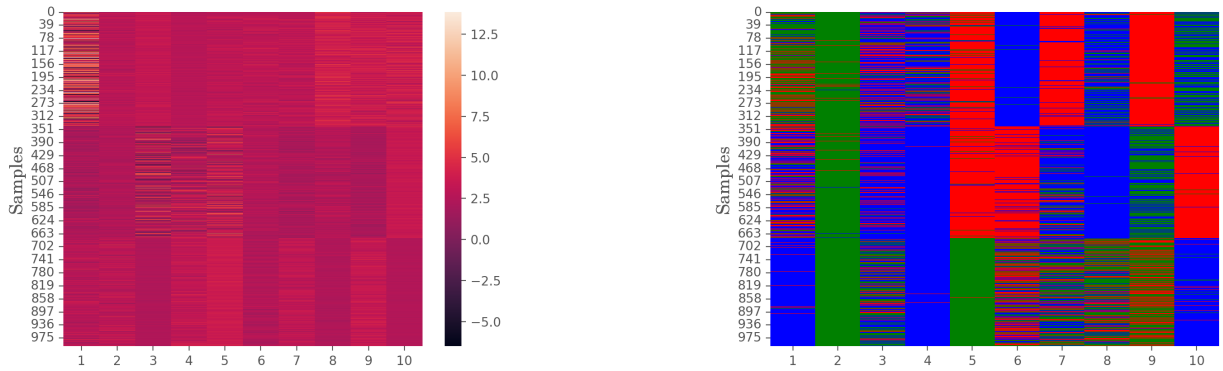
Adjusted Rand Index (ARI)

To evaluate the model, we use ARI [Hubert et al. 1985] score. It measures the agreement between two clustering assignments, accounting for chance. It is calculated using the formula:

$$\text{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}$$

where n_{ij} represents the number of data points in common between clusters in two cluster-

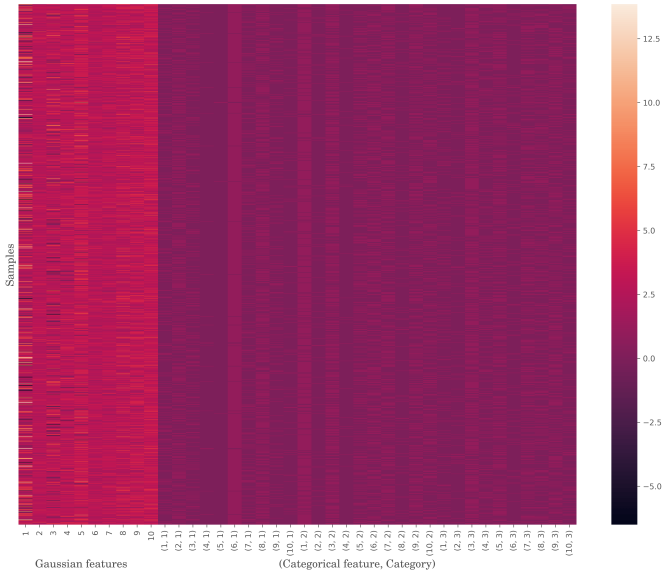
ings, a_i represents the total number of data points in cluster i, b_j represents the total number of data points in cluster j, and $\binom{n}{2}$ is the total number of pairs of data points. ARI ranges from -1 to 1, where 1 indicates perfect agreement, 0 indicates random agreement, and negative values indicate disagreement beyond random chance.



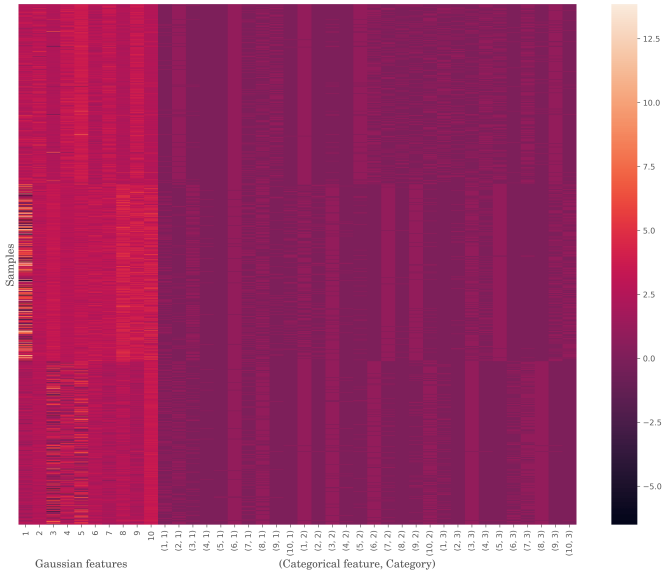
Gaussian Features

Categorical Features

Figure 6: Predicted mixture components of the data in Figure 4, we ran our model on the same data starting with initial $K = 20$ and then the model converged to the optimal $K = 3$. It took 17 seconds for 2 training runs of 20 iterations each. We got an ARI [See (6.1) for more info] of 0.99



(a)



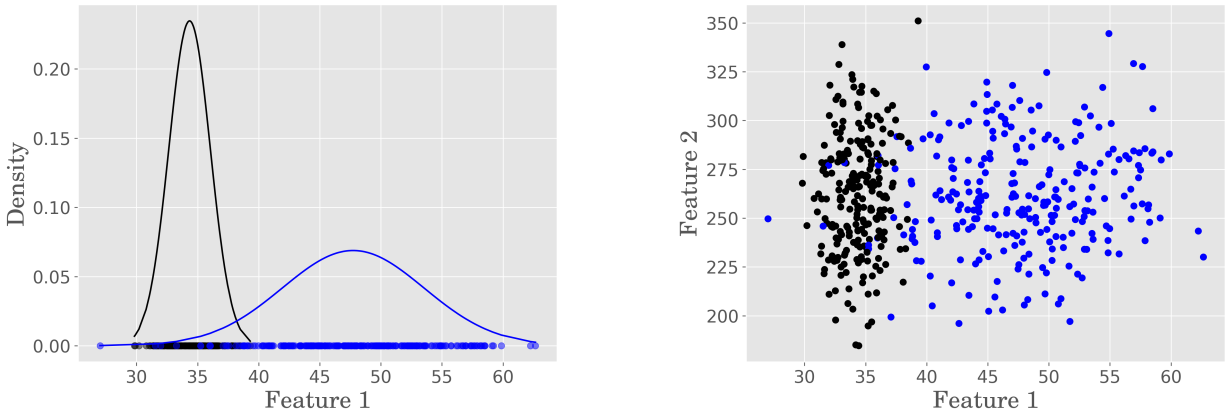
(b)

Figure 7: We used scikit-GMM to run the data given in Figure 4 with EM algorithm. For 40 training runs it took 14 seconds to run the model. We got an ARI of 0.91 (a) In order to feed categorical data to the model, we convert it to one-hot-vector. (b) Same data with samples permuted according to true cluster assignments.

6.2 Clustering Performance with Non Relevant Features (noise) and Feature Selection

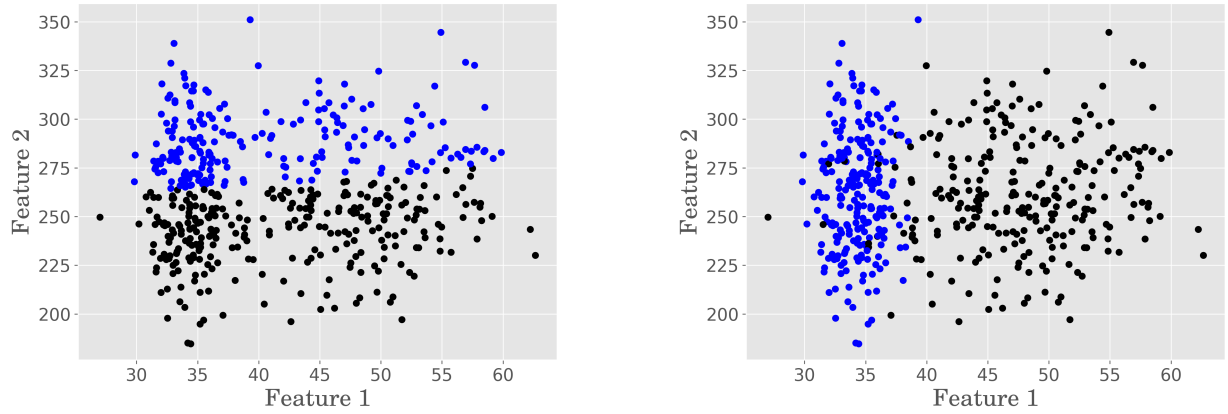
To show the importance of feature selection in clustering, we demonstrate an example. We generate 500 data samples for $K = 2$ clusters and a Gaussian features as given in Section 6.1.

Then we add a feature with Gaussian noise. Refer to Figure 8. Then the importance of feature selection can be explained with Figure 9.



(a) (b)

Figure 8: (a) Data with 2 mixture components with only 1 feature relevant for the clustering. (b) Data with an additional feature 2 being added as a Gaussian noise with, $\mu = 264.06$, $\sigma^2 = 28.3$

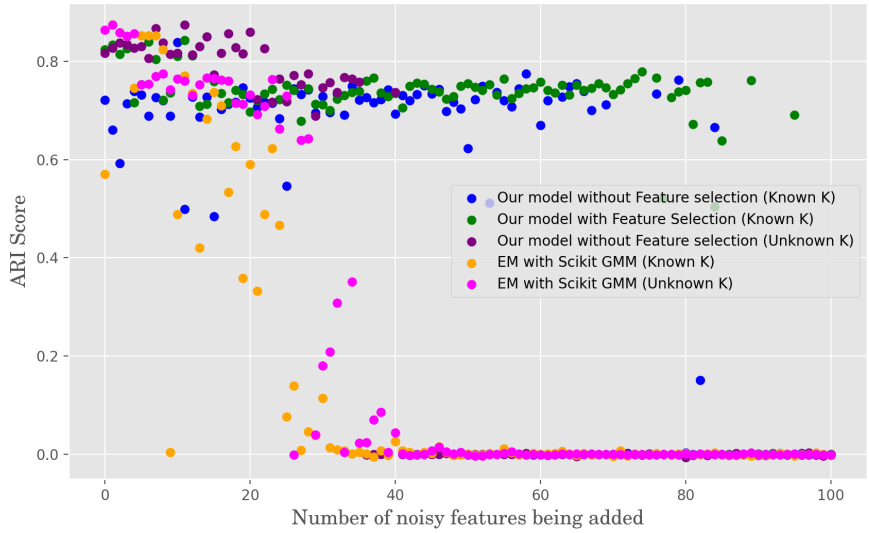


(a) (b)

Figure 9: (a) Fit by EM algorithm for GMM. For the 998 runs of 1000 different initialisation EM algorithm failed to predict the clustering (b) For any initialisation, our model with Feature selection, it predicts the noisy feature successfully giving the ARI of 0.98.

Only Gaussian (Higher Dimensions) with Feature Selection

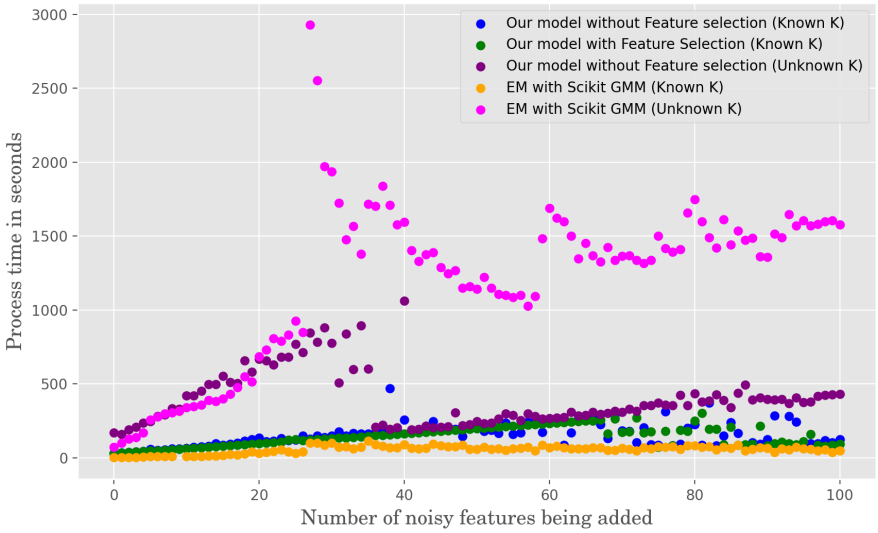
First we generated Gaussian mixtures with total clusters, $K = 5$, number of data samples, $N = 1000$ and total number of features, $D = 5$. Then added extra features with noise and went up to total 100 features.



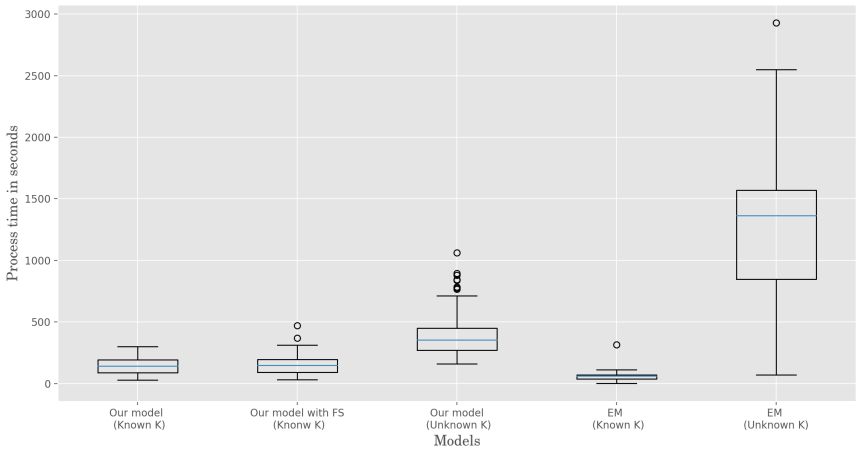
(a)

Figure 10: For Only Gaussian data, the ARI score comparison of the models where each model is ran with 2 initializations. This is the Scatter plot of the ARI score while adding noise to the data.

The results are shown in Figure 10. We got superior results with feature selection. In the case of unknown value of total number of clusters, we started our model with initial $K = 20$, and it got converged to the optimal K. In EM algorithm approach, we used BIC to figure out the optimal value of K among 1 to 20. This comparison is shown in Figure 12. The running time comparison is shown in Figure 11. First the running time increases linearly with number of features, but later it decreases because the model terminates the sampling when the posterior converges or stays within a very small bound for a few iterations, Section 15 for the implementation in detailed.



(a)



(b)

Figure 11: For Only Gaussian data, process time comparison for the ARI results given in Figure 10

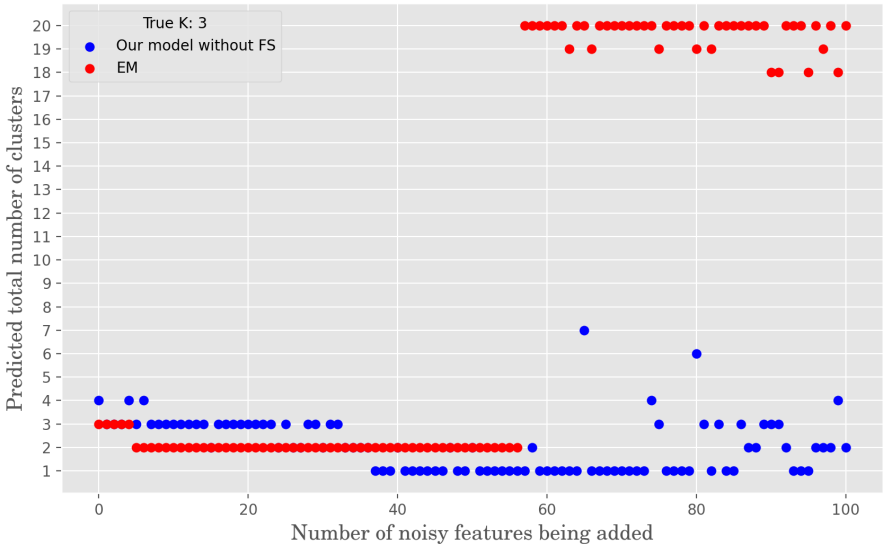
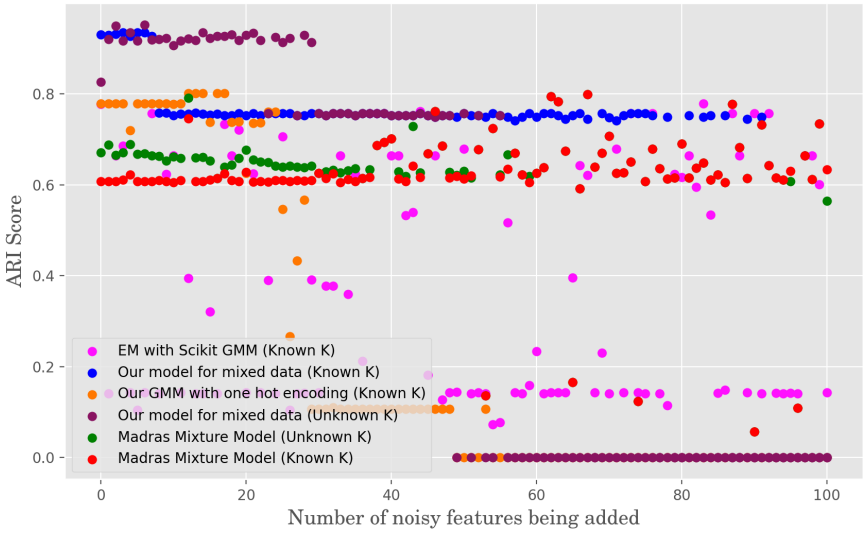


Figure 12: For Only Gaussian data, models comparison for total number of clusters prediction.

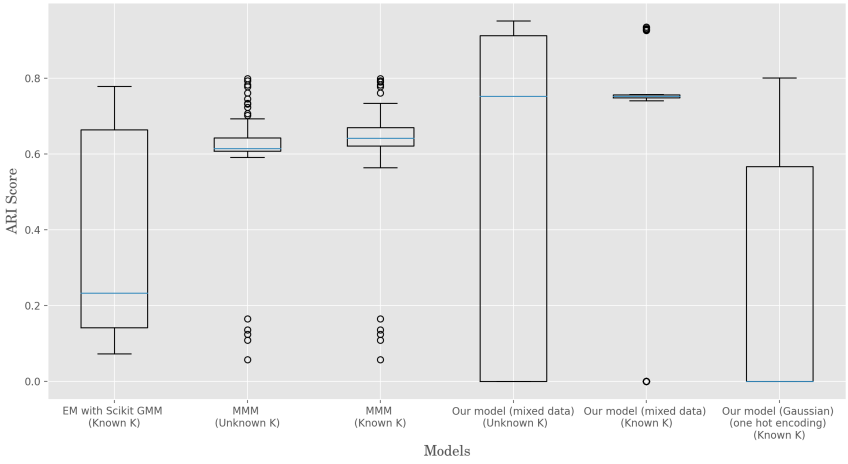
Heterogeneous Data (Gaussian and Categorical) with Noise

The data for this experiment was generated using a mixture of Gaussian and categorical distributions. The dataset comprises 1000 data points distributed among 3 mixture components. First we start with each data point with 4 Gaussian features (gD) and 4 categorical features (cD), with each categorical feature having 3 categories (J). Then we start adding a feature of Gaussian noise for Gaussian data and Categorical noise (uniformly select categories for each feature) for categorical data. This process continues until 50 noise features are added to each distribution. In total, 101 datasets are generated, each containing a varying number of noisy features ranging from 0 to 100.

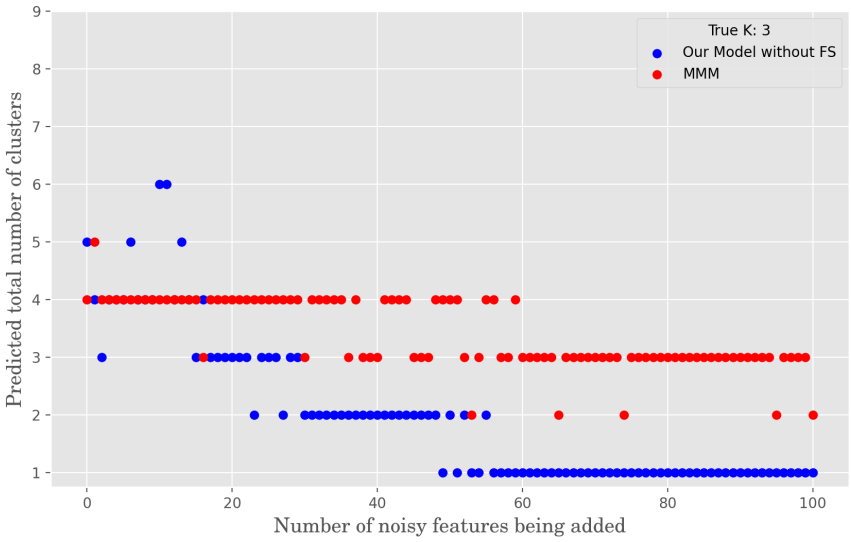
To evaluate our algorithm, we benchmarked it against a model from the literature, namely, the Madras Mixture Model for heterogeneous data (MMM) [Kumari and Siddharthan, 2023]. This model uses an EM algorithm to cluster heterogeneous data. We also compared our algorithm with Gaussian mixture models converting the categorical data into one-hot vectors. One is Scikit-GMM based on EM and other one is our algorithm for GMMs. Refer to Section 15 for more information. To gauge the efficacy of each clustering algorithm, two primary metrics were employed: Adjusted Rand Index (ARI) and Processing Time. Refer to the Figure 13 and Figure 14.



(a)



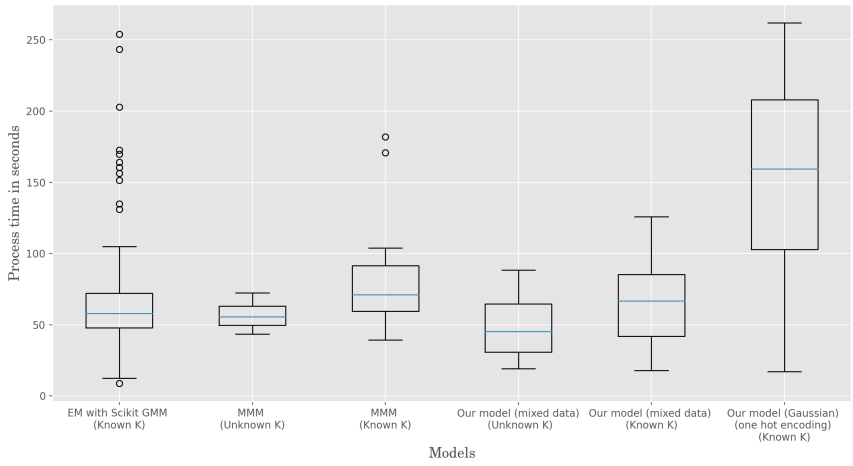
(b)



(c)

Figure 13: For the heterogeneous data (Categorical + Gaussian), ARI score comparison of the models where each model is ran with 2 initializations. (a) Scatter plot of the ARI score while adding noise to the data. (b) Boxplot for the same. (c) Comparison of predicted total number of clusters.

(a)



(b)

Figure 14: For the heterogeneous data (Categorical + Gaussian), process time comparison for the ARI results given in Figure 13 (a) Scatter plot of the ARI score while adding noise to the data. (b) Boxplot for the same.

Real Data

We used following datasets from the UCI Machine Learning Repository [Dua D et al. 2017],

- Breast (Breast Cancer Wisconsin): $N = 569$, 30 continuous and 1 categorical features
- Abalone: $N = 4177$, 8 continuous and 1 categorical features
- Ecoli: $N = 336$, 5 continuous and 2 categorical features
- Ionosphere: $N = 351$, 34 continuous features

These datasets contains both continuous and categorical data, primarily designed for classification tasks. However, based on the output variables, they exhibit notable tendencies towards

clustering. We ran our Gaussian Mixture Model with feature selection, focusing only on Gaussian features. We also ran our heterogeneous data model, and then compared them with the Madras Mixture Model (MMM) and the EM-algorithm of Scikit-GMM. Detailed results are presented in Table ??. In certain datasets, our model encountered difficulties in prediction, as it assumed all continuous data to be Gaussian. To address this limitation, additional distributions can be incorporated into the mixture model.

	Real-world Data (True K)			
	Breast (2)	Abalone (28)	Ecoli (2)	Ionosphere (8)
Our GMM with FS	0.678 (2)	0.068 (12)	0.39 (2)	0.241 (7)
Our model for mixed data without FS	0.644 (2)	0.069 (6)	0.391 (2)	0.201 (10)
MMM	0.644 (2)	0.066 (4)	0.448 (2)	0.213 (5)
EM algorithm	0.680 (2)	0.063 (15)	0.41 (4)	0.213 (3)

Table 1: Comparison of Models on Real-world Data. The total number of clusters predicted is indicated within the parentheses. K is known only in the case of our GMM with Feature Selection.

7 Conclusion

In this study, we presented a comprehensive evaluation of various clustering algorithms for handling heterogeneous data comprising both continuous Gaussian features and categorical features. Unlike traditional frequentist approaches like Expectation-Maximization (EM), our approach does not require pre-specifying the number of clusters and tends to converge to the optimal number by automatically adapting the mixture model complexity.

We demonstrated the effectiveness of the our models on synthetically generated data, achieving impressive Adjusted Rand Index (ARI) scores that outperform the EM algorithm implemented in Scikit-learn's Gaussian Mixture Model (GMM). On a heterogeneous dataset with $N = 1000$ samples, $K = 3$ clusters, 10 Gaussian features, and 10 categorical features (each with 3 categories), our Bayesian approach attained impressive ARI compare to Madras Mixture Model and EM algorithm, while exhibiting comparable runtime.

Furthermore, we have incorporated feature selection into our models, enabling the identification of relevant clustering variables. By introducing a binary matrix $\mathbf{\Phi}$ that indicates the importance of each feature for each cluster, our models can automatically select the most informative features during the clustering process. We have demonstrated the effectiveness of this feature selection approach on synthetic data with added noise features. Even in the presence of 50 noisy features (both Gaussian and categorical), our approach with feature selection has

achieved significant accuracy.

Our proposed models offer several advantages over existing methods:

1. Ability to handle heterogeneous data containing both continuous and categorical features.
2. Automatic determination of the optimal number of clusters without pre-specification.
3. Integrated feature selection to identify relevant clustering features.
4. Flexibility to incorporate prior knowledge through the choice of prior distributions.
5. Improved clustering performance, as demonstrated by higher ARI scores on synthetic data with comparable runtime.

Our models have the potential for broad applicability in various domains involving heterogeneous data, such as customer segmentation, genomic data analysis, and text clustering. Future work could involve extending the models to handle other data types, such as ordinal or count data, and exploring more complex prior distributions to encode domain-specific knowledge.

While MCMC sampling makes implementation slow for large datasets, this can be improved by employing parallel computing. Nonetheless, the results remain interpretable and accurate. We also need to incorporate different types of distributions to account for better clustering.

Overall, our approach has successfully developed the ways of putting both clustering and feature selection together in Bayesian framework, has demonstrated superior performance compared to traditional frequentist methods like EM. The flexibility and robustness of our models, along with their ability to handle diverse data types and automatically determine the optimal number of clusters, make them promising candidates for a wide range of real-world clustering applications.

References

[Fowlkes et al., 1988] Fowlkes, E. B., Gnanadesikan, R., and Kettering, J. R. (1988). *Variable Selection in Clustering*. Journal of Classification, 5, 205–228.

[Milligan, 1989] Milligan, G. W. (1989). *A Validation Study of a Variable Weighting Algorithm for Cluster Analysis*, Journal of Classification, 6, 53–71.

[Gnanadesikan, R. et al, 1995] Gnanadesikan, R., Kettering, J. R., and Tao, S. L. (1995). *Weighting and Selection of Variables for Cluster Analysis*. Journal of Classification, 12, 113–136.

[Brusco et al., 2001] Brusco, M. J., and Cradit, J. D. (2001). *A Variable Selection Heuristic for k-Means Clustering*. Psychometrika, 66, 249–270.

[Gilks et al., 1996] Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (eds.) (1996). *Markov Chain Monte Carlo in Practice*. London: Chapman & Hall.

[Andrieu et al., 2003] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan (2003). *An introduction to mcmc for machine learning. Machine learning 50(1):5–43*

[Bonawitz 2008] Keith Allen Bonawitz.(2008) *Composable Probabilistic Inference with Blaise*. PhD thesis, Massachusetts Institute of Technology,

[Geyer 2011] Charles Geyer (2011). *Introduction to markov chain monte carlo*. Handbook of markov chain montecarlo, pages 3–48.

[Turchin 1971] Valentin F Turchin (1971). *On the computation of multidimensional integrals by the monte-carlo method*. Theory of Probability & Its Applications, 16(4):720–724

[Geman and Geman 1984] Stuart Geman and Donald Geman. *Stochastic relaxation, gibbs distributions, and the bayesian restoration of images*. IEEE Transactions on pattern analysis and machine intelligence.

[Liu et. al. 1994] Jun S. Liu, Wing Hung Wong, and Augustine Kong (1994). *Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes*. Biometrika, 81(1):27–40,

[Bellman 1957] Bellman, R. (1957). *Dynamic Programming*. Princeton University Press

[Bartholomew et al. 2011] Bartholomew, D., Knott, M. and Moustaki, I. (2011). *Latent Variable Models and Factor Analysis*. Wiley.

[Ritter 2014] Ritter, G. (2014). *Robust cluster analysis and variable selection*. CRC Press

[McLachlan et al. 1988] McLachlan, G. J. and Basford, K. E. (1988). *Mixture models: Inference and applications to clustering*. Marcel Dekker.

[Guyon et al. 2003] Guyon, I. and Elisseeff, A. (2003). *An introduction to variable and feature selection*. Journal of Machine Learning Research 3 1157–1182.

[Liu et al. 2003] Liu, J. S., Zhang, J. L., Palumbo, M. J. and Lawrence, C. E. (2003). *Bayesian clustering with variable and transformation selections*. In Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting.

[law et al. 2004] Law, M. H. C., Figueiredo, M. A. T. and Jain, A. K. (2004). *Simultaneous feature selection and clustering using mixture models*. Pattern Analysis and Machine Intelligence, IEEE Transactions on 26 1154–1166.

[Tadesse et al. 2005] Tadesse, M. G., Sha, N. and Vannucci, M. (2005). *Bayesian variable selection in clustering high-dimensional data*. Journal of the American Statistical Association 100 602–617.

[Kim et al. 2006] Kim, S., Tadesse, M. G. and Vannucci, M. (2006). *Variable selection in clustering via Dirichlet process mixture models*. Biometrika 93 877–893.

[Swartz et al. 2008] Swartz, M. D., Mo, Q., Murphy, M. E., Lupton, J. R., Turner, N. D., Hong, M. Y. and Vannucci, M. (2008). Bayesian variable selection in clustering high-dimensional data with substructure. Journal of Agricultural, Biological, and Environmental Statistics 13 407

[Pan et al.] Pan, W. and Shen, X. (2007). *Penalized model-based clustering with application to variable selection*. Journal of Machine Learning Research 8 1145–1164

[Bhattacharya et al. 2014] Bhattacharya, S. and McNicholas, P. D. (2014). *A LASSO-penalized BIC for mixture model selection*. Advances in Data Analysis and Classification 8 45–61

[Wanf and Zhu 2008] Wang, S. and Zhu, J. (2008). *Variable selection for model-based highdimensional clustering and its application to microarray data*. Biometrics 64 440–448.

[Guo et al. 2010] Guo, J., Levina, E., Michailidis, G. and Zhu, J. (2010). *Pairwise variable selection for high-dimensional model-based clustering*. Biometrics 66 793–804

[Xie et al. 2008] Xie, B., Pan, W. and Shen, X. (2008a). *Variable selection in penalized model-based clustering via regularization on grouped parameters*. Biometrics 64 921–930.

[Zhou et al.] Zhou, H., Pan, W. and Shen, X. (2009). *Penalized model-based clustering with unconstrained covariance matrices*. Electronic Journal of Statistics 3 1473–1496.

[Witten et al. 2010] Witten, D. M. and Tibshirani, R. (2010). *A framework for feature selection in clustering*. Journal of the American Statistical Association 105 713–726.

[Raftery et al. 2006] Raftery, A. E. and Dean, N. (2006). Variable selection for model-based clustering. Journal of the American Statistical Association 101 168–178

[Maugis et al. 2008] Maugis, C., Celeux, G. and Martin-Magniette, M. L. (2009a). Variable selection for clustering with Gaussian mixture models. Biometrics 65 701– 709

[Maugis et al. 2009] Maugis, C., Celeux, G. and Martin-Magniette, M. L. (2009b). Variable selection in model-based clustering: A general variable role modeling. Computational Statistics and Data Analysis 53 3872–3882.

[Maugis et al. 2012] Maugis-Rabusseau, C., Martin-Magniette, M.-L. and Pelletier, S.(2012). SelvarClustMV: Variable selection approach in model-based clustering allowing for missing values. Journal de la Société Française de Statistique 153 21–36

[Baxter 2002] Baxter, Rohan. (2002). Minimum Message Length Inference: Theory and Applications.

[Marbac and Sedki 2017] Marbac, M. and Sedki, M. (2017a). Variable selection for model-based clustering using the integrated complete-data likelihood. Statistics and Computing 27 1049–1063

[Scikit-learn] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E (2011). *Scikit-learn: Machine Learning in **P**ython*. Journal of Machine Learning Research.

[Clogg, 1998] Clogg, C.C. (1988). *Latent Class Models for Measuring*. In: Langeheine, R., Rost, J. (eds) Latent Trait and Latent Class Models. Springer, Boston, MA.

[Murphy, 2007] Kevin P. Murphy (2007). *Conjugate Bayesian analysis of the Gaussian distribution*. Technical report, University of British Columbia.

[Hoff, 2009] Peter D Hoff (2009). *A first course in Bayesian statistical methods*. Springer Science & Business Media.

[Bernardo et al., 2003] M Bernardo, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, and M West (2003). *Bayesian clustering with variable and transformation selections.* In Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting, page 249. Oxford University Press, USA.

[Jain et al., 2004] Law, M. H. C., Figueiredo, M. A. T. and Jain, A. K. (2004). *Simultaneous feature selection and clustering using mixture models.* Pattern Analysis and Machine Intelligence, IEEE Transactions on 26 1154–1166

[Neal, 2009] Neal, R. M. (2000). *Markov Chain Sampling Methods for Dirichlet Process Mixture Models.* Journal of Computational and Graphical Statistics, 9(2), 249–265.

[Murphy, 2012] Kevin P. Murphy (2012). *Machine learning: A Probabilistic Perspective.* The MIT Press.

[Fraley and Raftery, 2007] Fraley, C., Raftery (2007), *A. Bayesian Regularization for Normal Mixture Estimation and Model-Based Clustering.* Journal of Classification 24, 155–181.

[Seeger, 2004] Matthias Seeger (2004). *Low rank updates for the cholesky decomposition.* Technical report.

[Mengersen, 2011] Judith Rousseau and Kerrie Mengersen (2011). *Asymptotic behaviour of the posterior distribution in overfitted mixture models.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), 73(5):689–710.

[Hubert et al. 1985] Hubert L, Arabie P. (1985) *Comparing partitions.* Journal of classification.

[Kumari and Siddharthan, 2023] Chandrani Kumari, Rahul Siddharthan (2023) *MMM and MMMSynth: Clustering of heterogeneous tabular data, and synthetic data generation.* arXiv:2310.19454

[Dua D et al. 2017] Dua D, Graff C, et al (2017). UCI machine learning repository;.