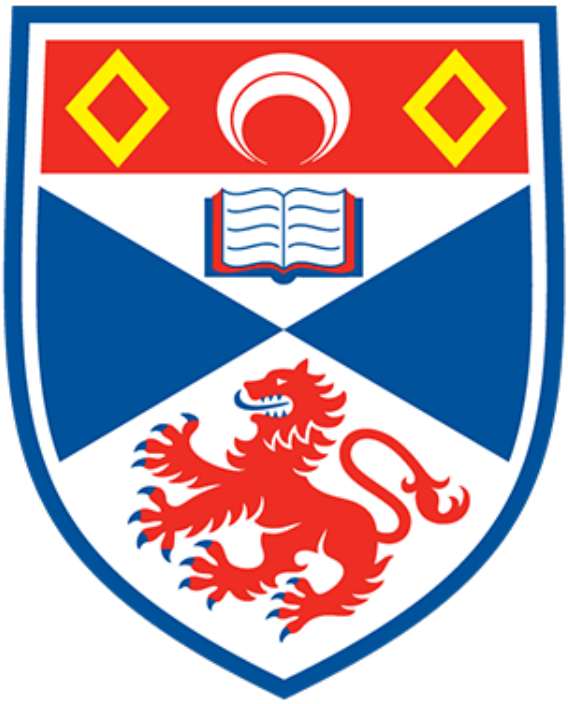
MSc Computer Science Dissertation Project

# Bayesian Unsupervised Learning with Missing Data for Mixture Modelling



University of
St Andrews

**James Zhang (190015412)**

**Supervised By Dr Lei Fang**

August 12, 2025

# Abstract

In real-world datasets, missing data is a pervasive issue that can compromise the validity of downstream statistical analysis. Traditional solutions often rely on pre-processing techniques such as ad-hoc imputation, or even complete deletion of incomplete records. Such methods risk introducing bias, discard valuable information, and generally fail to model the uncertainty inherent in missing values. This ultimately compromises the integrity of subsequent analysis.

Hence, the central goal of this project is to investigate the implementation of one-step, statistically principled inference algorithms that are robust enough to discover underlying structures within the data despite the presence of missing entries. This will be pursued in a fully Bayesian approach where the missing data are treated as hidden variables and integrated into the inference process. With this approach, we preserve and model the uncertainty behind the missing data by learning their distribution rather than rely on simple point estimates.

The specific unsupervised learning task considered in this research is clustering : uncovering latent groupings in data without access to labels. To achieve this, we adopt a generative modeling perspective using Mixture Models, where the data is assumed to arise from a mixture of latent components, each explained by its own probability distribution. This frames the problem in terms of learning the underlying generative process that explains the observed data. Utilizing such an approach not only captures the complexity of a multi-modal dataset, but also offers a principled mechanism for handling missingness through joint inference over latent variables, model parameters, and missing data. This ultimately constitutes a coherent probabilistic framework capable of robustly learning structure from incomplete data.

This research explores two fully Bayesian inference algorithms : Gibbs Sampling and Variational Inference. We additionally benchmark the results of these two approaches against two-step ad-hoc imputation strategies as well as inference methods based on Maximum Likelihood Estimation (MLE). These methods are applied to both Bernoulli Mixture Models for datasets with binary features, as well as Gaussian Mixture Models, for datasets with continuous features. The approaches are evaluated against varying levels of missing data. The evaluation criteria used are clustering performing (Adjusted Random Index), quality of the model fit (log-likelihood), and imputation accuracy using learned parameters (Root Mean Square Error). The results show that fully Bayesian approaches outperform both maximum likelihood and ad-hoc imputation strategies across all metrics in a wide range of settings.

# Declaration

I hereby certify that this dissertation, which is approximately 14825 words in length, has been composed by me, that it is the record of work carried out by me and that it has not been submitted in any previous application for a degree. This project was conducted by me at the University of St Andrews from June/2025 to August/2025 towards fulfillment of the requirements of the University of St Andrews for the degree of MSc Computer Science under the supervision of Dr. Lei Fang. In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

**August 12, 2025**

# Chapter 1

# Acknowledgments

I would like to thank my supervisor, Dr Lei Fang, for his invaluable guidance throughout the development and implementation of the primary components of this project. I am especially grateful for his dedicated support in strengthening my understanding of the mathematical foundations through structured learning and ongoing discussion, which greatly deepened my grasp of the subject matter.

# Contents

# Chapter 2

# Introduction

In practical data analysis and machine learning applications, missing data is a common and often unavoidable challenge. Missing entries can arise naturally in datasets in many real world scenarios. In survey data, respondents may leave answers blank, either intentionally or due to misunderstanding. In sensor networks or digital logging systems, technical failures such as power loss or transmission errors can cause loss of data points [3]. Even in large-scale governmental or political datasets, such as parliamentary voting records, missing entries can occur when Members of Parliament (MPs) abstain from voting, are ineligible to participate in certain divisions, or fail to arrive in time to cast their vote [18].

These omissions can substantially harm downstream analyses. Common solutions for handling missing data are often naive pre-processing techniques such as imputing missing entries with ad-hoc, crude estimates, or deleting incomplete data-points entirely. While computationally inexpensive and simple to implement, these techniques can significantly degrade the quality of inference by discarding valuable information and introducing harmful bias, even at moderate levels of missing data. The consequences of these risks become magnified in multi-modal datasets, where the inherent heterogeneity and complex latent structures render naive summarizing statistics incapable of capturing the true patterns of the data, leading to distorted and misleading conclusions.

In this research, we focus on the unsupervised learning task of clustering : uncovering latent groupings in data without access to labels. Thus we choose a domain where data is naturally multi-modal, where the risks of ad-hoc approaches become particularly apparent. We adopt a generative modeling perspective using mixture models, where the data is assumed to arise from a mixture of latent components, each described by its own probability distribution. This approach frames the clustering task in terms of learning the underlying generative process that explains the observed data. To overcome the deficiencies of ad-hoc methods, we explore principled fully Bayesian techniques for clustering data using the mixture modeling perspective. This approach naturally accommodates a fully Bayesian treatment of missing data, wherein we explicitly model the uncertainty in missing entries, treating them as latent variables to be inferred jointly with other latent variables and mixture model parameters.

This work investigates two fully Bayesian inference algorithms for learning the mixture model generative process: Gibbs Sampling and Variational Inference. These methods are bench-

marked against probabilistic maximum-likelihood approaches and ad-hoc baselines. Thus we highlight the benefits of statistically principled handling of missing entries, and isolate the value added by a fully Bayesian treatment. Performance is evaluated across both binary-feature Bernoulli Mixture Models (BMMs) and continuous-feature Gaussian Mixture Models (GMMs) using multiple metrics: clustering performance (Adjusted Rand Index), model fit quality (log-likelihood), and imputation accuracy (Root Mean Square Error). The aim of this research is to demonstrate that the proposed fully Bayesian approaches consistently deliver superior robustness and performance in the presence of missing data, particularly under high missingness and feature dimensionality.

## 2.1 Project Objectives

The project objectives can be grouped into primary, secondary, and tertiary objectives.

**Primary Objectives**

1. Implement Gibbs Sampling algorithm for both Bernoulli and Gaussian mixture models, treating missing data as hidden variables to infer.
2. Develop principled synthetic data generation process to evaluate the algorithm against. Additionally acquire a number of real datasets.
3. Simulate missing-completely-at-random missingness on synthetic and real-world datasets through principled pre-processing.
4. Benchmark the Gibbs sampling algorithm against Expectation-Maximization Algorithm with ad-hoc mean, median, and mode imputation.

**Secondary Objectives**

1. Implement one-step Expectation Maximization Algorithm for both Bernoulli and Gaussian mixture models, treating missing data as hidden variables.
2. Implement Variational Inference algorithm for both Bernoulli and Gaussian mixture models, treating missing data as hidden variables.
3. Evaluate Gibbs Sampling and Variational Inference algorithms and benchmark against Expectation Maximization algorithm with and without imputation, as well as the K-Means algorithm with imputation.

**Tertiary Objectives**

1. Implement object-oriented class structure for all inference algorithms to streamline downstream usage and research evaluation.
2. Explore incorporating missing-not-at-random (MNAR) missingness mechanism into model and algorithm design.
3. Perform evaluation on clustering performance of MNAR approach.

# Chapter 3

# Context Survey

## 3.1 Missing Data

Missing data in this context refers to the absence of feature values in individual instances within a dataset to be used for analysis. This phenomenon can significantly impact the integrity and reliability of subsequent analysis of statistical models and learning algorithms. The presence of missing data not only reduces the effective sample size but also introduces uncertainty and potential biases into analysis. The nature of missing data can be classified into three categories defined by Rubin and Little [34] :

1. Missing Completely At Random (MCAR) : The probability of a value being missing is completely independent of any data, observed or unobserved.
2. Missing At Random (MAR) : The probability of missingness depends only on the observed data. In other words, the missing features are entirely dependent on the observed features.
3. Missing Not At Random (MNAR) : The probability of the missingness depends on unobserved data such as the missing values themselves, or some external hidden factor.

Understanding the mechanism of missingness is necessary for selecting an appropriate modeling strategy. In this research, we focus specifically on the MCAR setting, which represents the most restrictive and idealized assumption. While rare in real-world data, the MCAR assumption provides a controlled testing ground to evaluate the robustness of inference algorithms. Under MCAR, missingness introduces randomness but not systematic bias, allowing for clear attribution of performance to the inference method rather than to bias in the missingness pattern itself.

We additionally discuss and evaluate a possible extension to handle a restricted MNAR case where the nature of missingness of a data-point is dependent on an un-observable external factor. This assumption does align with real-world phenomenon. For instance, for quality-of-life surveys, sicker patients, particularly those in end-of-life care, are more likely to skip questionnaires [2]. In another example, voting patterns of MPs' in British parliament on legislative divisions are often dependent on the political party of the MP. For instance, the House of Common Library found that voting participation rates varies by political party from 88.7 percent for Conservatives, to 70.5 percent for the SNP party [37]. In this research we propose a potential solution for this form of missingness, however, the main experiment

is restricted to the MCAR case.

## 3.2 Traditional Approaches for Missing Data

Two of the most common baseline strategies for handling missing data are deletion of incomplete instances from the dataset, and two-step imputation methods [16, 15]. The former approach is often referred to as complete case analysis (CC), where only fully observed data instances are used [15]. Such an approach, while simple, reduces statistical power by not only reducing sample size for analysis, but also discards potentially valuable information held in the observed features of incomplete entries. In a clustering context, this not only prevents incomplete data-points from being assigned to clusters, but also limits the available evidence for inferring the structure and characteristics of the clusters themselves.

As an example, a simulation study by Raghunathan [32] demonstrates the risks of complete case analysis in parameter estimation. After generating synthetic datasets with known logistic regression parameters, missingness was introduced in a covariate using a mechanism designed to mimic real-world cohort data. When fitting models only on the complete cases, the estimated regression coefficient for one predictor was consistently biased, centered around 0.20 instead of the true value of 0.5. This illustrates how omitting incomplete cases can lead to substantial bias in downstream inference, even with large sample sizes.

Traditional two-step imputation strategies for missing data are ultimately a data pre-processing technique and therefore are referred to as a two-step approach : we first impute the missing values (possibly based on some initial analysis), then perform our downstream analysis using the imputed dataset. Imputation strategies can vary from basic ad-hoc approaches to more advanced approaches utilizing models to predict missing values.

### 3.2.1 Ad-Hoc Imputation

Examples of ad-hoc approaches include mean, median or mode imputation, wherein missing values are replaced with some statistic based on the observed data for the missing feature [16]. Such ad-hoc approaches are often utilized due to their simplicity, but lead to bias and loss of precision. These risks are amplified in complex, multi-modal datasets, where summary statistics (e.g. the mean) can vary significantly across different subgroups to which a data-point may belong. Applying imputation methods that ignore this heterogeneity can distort the overall data distribution, often leading to an artificial reduction in variance and underestimation of uncertainty [15]. This distortion can have detrimental effects on downstream tasks like clustering, where preserving the true structure and variability of the data is essential [7].

### 3.2.2 Advanced Frameworks

Several widely used frameworks address missing data by fitting predictive models to estimate missing features from observed ones. While often statistically principled, these methods operate as two-step procedures, separating imputation from the subsequent analysis task. This separation can be inherently inefficient, particularly for clustering with mixture models,

where the imputation step may already involve learning the very latent structure that the analysis step seeks to rediscover.

One example is Buck's regression-based imputation [8], which trains regression models on complete cases, estimating the mean vector and covariance matrix from these observations before predicting missing values. Two more prominent approaches are **Multiple Imputation by Chained Equations (MICE)** [9] and **Multiple Imputation** (MI) [34].

MICE iteratively models each incomplete variable as a function of the others, starting with crude initial imputations and cycling through variables in a "chain" until convergence [9]. These models can be deterministic or probabilistic, and the approach flexibly accommodates a range of predictive algorithms.

MI, originally proposed by Rubin [35], creates $m$ completed datasets by sampling from a probabilistic model that reflects uncertainty in the imputations [34, 20]. Each dataset is analyzed independently, and results are combined via Rubin's pooling rules. If the imputations are drawn using posterior predictive distributions, MI can be considered *fully Bayesian* [10].

However, even in fully Bayesian form, MI and MICE still decouple imputation from inference. In clustering with mixture models, this is both unnecessary and inefficient: if the imputation step already fits a mixture model and infers component assignments, re-clustering the imputed data merely duplicates computation without necessarily improving inference. The one-step fully Bayesian methods used in this work achieve the same result more coherently, eliminate potential mismatches between imputation and analysis, and remove an arbitrary pre-processing stage [20]. Moreover, MI and MICE introduce additional hyperparameters (e.g. number of imputations, pooling rules, model specification) that require careful tuning and can influence results, whereas the one-step approach integrates all uncertainty directly within the generative model.

Another popular deterministic imputation framework is K-Nearest Neighbor (KNN) imputation. Here, each incomplete data-point is treated as a vector in a feature space (e.g. Euclidean), and missing values are estimated from its nearest neighbors [7, 17]. While KNN can partially capture multi-modal structure, it suffers from key limitations: high computational cost due to pairwise distance calculations, no principled treatment of uncertainty or outliers, and sensitivity to the choice of $k$ [4].

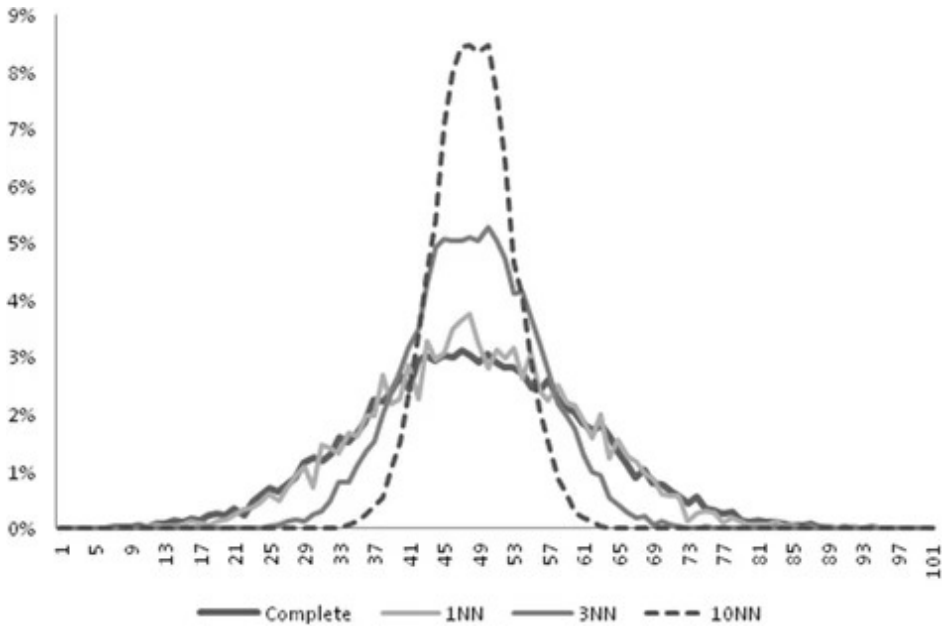



Figure 3.1: Distribution of variable $X_o$ after KNN mean imputation with increasing values of `k` from [4]. We can see how as the number of neighbors used for imputation increases, naturally the variance of the imputed variable decreases.

## 3.3 Fundamentals of Bayesian Inference

Bayesian inference is a probabilistic framework for learning from data by systematically incorporating prior beliefs and updating them in light of new observed data. At its core lies Bayes' theorem, which defines a framework for updating a posterior distribution over unknown quantities (such as model parameters and latent variables) given a notion of prior belief (prior distribution over those unknown quantities) and new data. Bayes' theorem is given by

$$P(\theta|\mathbf{X}) = \frac{P(\mathbf{X} \mid \theta)P(\theta)}{P(\mathbf{X})}$$

Where

- $\mathbf{X}$ is the newly observed data
- $\theta$ are the unknown variables (e.g. model parameters or latent variables)
- $P(\mathbf{X} \mid \boldsymbol{\theta})$ : denotes the data likelihood representing the probability of observing the data $\mathbf{X}$ conditioned on the current estimates of the parameters $\boldsymbol{\theta}$
- $P(\theta \mid \mathbf{X})$ is the **posterior distribution**, representing updated belief in the value of unknown parameters $\theta$ given the newly observed data $\mathbf{X}$
- $P(\theta)$ is the **prior distribution**, representing prior beliefs about $\theta$ before observing $\mathbf{X}$
- $P(\mathbf{x})$ is the marginal likelihood which acts as a normalizing constant for the posterior distribution, ensuring that is a valid distribution

$$P(\mathbf{X}) = \int P(\mathbf{X}|\theta')P(\theta')d\theta' \tag{3.1}$$

Unlike the frequentest perspective, which focuses on deriving point estimates based on $\mathbf{X}$, the Bayesian approach explicitly models uncertainty through the posterior distribution, enabling more robust inference, especially in the face of noisy or limited data. This is particularly advantageous when dealing with missing data, where capturing uncertainty is crucial. The Bayesian framework naturally accommodates this by treating both missing entries as well as the underlying structure of the data (latent variables and model parameters) as random variables in a joint posterior. As a result, the framework incorporates all sources of uncertainty, making inference more coherent.

## 3.4 Mixture Models

In our clustering approach, the hidden structure we aim to discover are latent grouping of data. Naturally the central model used in this research is the **mixture model** which provides a generative probabilistic framework to represent subpopulations within an overall population of data points. Mixture models assume that the data of interest are generated by a finite mixture of components, where the data belonging to each component are modeled by a separate probability distribution. This frames the inference task in terms of learning the underlying generative process that explains the observed data.

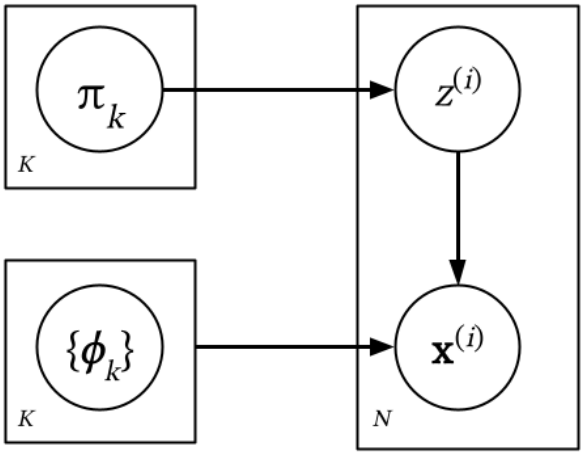



Figure 3.2: General mixture model structure (Bayesian Network)

A general mixture model is a hierarchical model as shown in Figure 3.2 and consists of the following components:

- N random variables corresponding to our observations (data points) $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$. Each data-point is assumed to be distributed according to a mixture of $K$ components where each component is explained by a distribution of the same parametric family.
- A corresponding finite and discrete set of $N$ latent component assignments $\mathbf{z} = \{z_1, ...z_N\}$ for each of the $N$ observations in $\mathbf{X}$ where $z_i \in \{1, ..., K\}$.
- A set of $K$ mixture weights $\boldsymbol{\pi} = \{\pi_1, \pi_2, \cdots, \pi_K\}$ which serve as parameters for a categorical distribution over latent component assignments $\mathbf{z}$. In other words, each component weight $\pi_k$ corresponds to a probability (between 0 and 1 inclusive) of selecting a component $k$, where $\sum_k^K \pi_k = 1$.

- A set of $K$ model parameters $\boldsymbol{\phi} = \{\phi_1, \phi_2, \cdots, \phi_K\}$, each specifying the parameters for the underlying distribution of each of the corresponding $K$ components. Here $\phi$ can denote multiple parameters such as in the case with Gaussian distributions where $\phi_k = \{\mu_k, \Sigma_k\}$.

With this, the probability density of a single observation $\mathbf{x}^{(i)}$ is given by

$$\begin{aligned} p(\mathbf{x}^{(i)}) &= \sum_{z^{(i)}} p(z^{(i)}, \mathbf{x}^{(i)}) \\ &= \sum_{k}^{K} p(z^{(i)} = k) p(\mathbf{x^{(i)}} \mid z^{(i)} = k) \\ &= \sum_{k}^{K} \pi_k p(\mathbf{x}^{(i)} \mid z^{(i)} = k) \end{aligned} \tag{3.2}$$

Here $p(z^{(i)})$ is a categorical distribution over a finite discrete set of latent component assignments $\{1, ..., K\}$, and $p(\mathbf{x}^{(i)} \mid z^{(i)} = k)$ denotes the likelihood of observing data-point $\mathbf{x}^{(i)}$ given that it belongs to the component $k$.

This hierarchical structure makes mixture models a powerful choice of uncovering hidden groups and modeling complex, multi-modal data distributions. By defining a joint probability distribution over observed data, latent component assignments, and model parameters, mixture models support coherent inference of both latent variables and parameters of the underlying generative process. In a fully Bayesian treatment, we treat all unknown quantities, including mixture weights and component parameters, as random variables and place prior distributions over them to model their uncertainty. The resulting posterior distribution takes the following form

$$\begin{aligned} p(\mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\phi} \mid \mathbf{X}) &= \frac{p(\mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\phi}, \mathbf{X})}{p(\mathbf{X})} \\ &= \frac{p(\mathbf{X} \mid \mathbf{z}, \boldsymbol{\phi})\, p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi})\, p(\boldsymbol{\phi})}{p(\mathbf{X})} \end{aligned} \tag{3.3}$$

The denominator $p(\mathbf{X})$ is the marginal likelihood which serves as normalizing constant that enforces that the posterior is a valid distribution. The term $p(\mathbf{X} \mid \mathbf{z}, \boldsymbol{\phi})$ denotes the data likelihood given the component assignments for each data-point $\mathbf{z}$ and the model parameters for each component $\boldsymbol{\phi}$. The term $p(\mathbf{z} \mid \boldsymbol{\pi})$ denotes the prior over component assignments $\mathbf{z}$ given the mixing weights $\boldsymbol{\pi}$. Here, $p(\boldsymbol{\pi})$ and $p(\boldsymbol{\phi})$ denote prior distributions over parameters $\boldsymbol{\phi}$ and mixing weights $\boldsymbol{\pi}$ respectively.

We generally aim to choose prior distributions that are conjugate to their corresponding likelihood function. This allows us to derive closed-form posterior updates in light of new observations that take the same form as the prior. Given that the distribution over latent assignments $p(\mathbf{z} \mid \boldsymbol{\pi})$ is categorical, a common choice for prior distribution for mixing weights $\boldsymbol{\pi}$ is a Dirichlet distribution which is parameterized by a concentration parameter $\boldsymbol{\alpha} = \{\alpha_{0,1}, ..., \alpha_{0,K}\}$. Given this, the posterior distribution over mixing weights $\boldsymbol{\pi}$ given observations over latent component assignments $\mathbf{z}$ takes the closed form of a Dirichlet dis-

tribution

$$p(\boldsymbol{\pi} \mid \mathbf{z}) \propto p(\mathbf{z} \mid \boldsymbol{\pi})p(\boldsymbol{\pi}) \tag{3.4}$$
$$= \text{Dir}(\boldsymbol{\pi} \mid \alpha_{0,1} + N_1, \cdots, \alpha_{0,K} + N_K)$$

where $N_k = \sum_i^N \mathbb{1}(z_i = k)$ and denotes the number of data-points assigned to component $K$.

The choice of priors over model parameters $\boldsymbol{\phi}$ depends on the assumed parametric family of the component distribution (e.g Gaussian, Bernoulli), but again, we aim to choose priors that are conjugate to the data likelihood $p(\mathbf{X} \mid \mathbf{z}, \boldsymbol{\phi})$ to simplify posterior updates in light of data.

This research focuses on two parametric families of mixture models : **Gaussian Mixture Model (GMM)** and **Bernoulli Mixture Model (BMM)**.

### 3.4.1 Gaussian Mixture Model (GMM)

GMMs are used to model continuous multi-modal data. In particular, a GMM assumes that each data-point $\mathbf{x}^{(i)}$ is generated from one of $K$ Gaussian distributions. With this, our model parameters $\boldsymbol{\phi}_k$ are given by a mean vector $\boldsymbol{\mu}_k$ and a covariance matrix $\boldsymbol{\Sigma}_k$ for each component $k \in \{1, ..., K\}$. The model has the following probability density for a single data-point:

$$p(\mathbf{x}^{(i)} \mid \boldsymbol{\phi}) = \sum_k^K \pi_k \mathcal{N}(\mathbf{x}^{(i)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{3.5}$$

Thus the observed data likelihood over all data points factors as follows

$$p(\mathbf{X} \mid \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}, \mathbf{z}) = \prod_i^N \sum_k^K \pi_k \mathcal{N}(\mathbf{x}^{(i)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{3.6}$$

In the Gaussian case, it is sensible to choose either Normal-Wishart (NW) or Normal-Inverse-Wishart (NIW) prior distribution over Gaussian parameters. This research utilizes a NIW prior which is given by the following :

$$p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = p(\boldsymbol{\mu}_k \mid \boldsymbol{\Sigma}_k)p(\boldsymbol{\Sigma}_k) \tag{3.7}$$
$$= \mathcal{NIW}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \mid \mathbf{m}_{0,k}, \kappa_{0,k}, \nu_{0,k}, \mathbf{S}_{0,k}) \tag{3.8}$$
$$= \mathcal{N}(\boldsymbol{\mu}_k \mid \mathbf{m}_{0,k}, \frac{1}{\kappa_{0,k}}\boldsymbol{\Sigma}_k) \times \text{IW}(\boldsymbol{\Sigma}_k \mid \mathbf{S}_{0,k}, \nu_{0,k}) \tag{3.9}$$

Here:

- $\mathbf{m}_0$ is the prior mean,
- $\kappa_0$ is the scaling factor for the mean,
- $\nu_0$ is the degrees of freedom for the inverse-Wishart distribution
- $\mathbf{S}_0$ is the scale matrix.

These NIW hyperparameters are shared across components or can be set per component depending on prior knowledge.

The NIW prior is conjugate to the Gaussian data-likelihood $\mathcal{N}(\mathbf{x}^{(i)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. As such, the posterior over parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ for a component $k$, takes the closed-form of another NIW distribution given by the following [23]

$$
\begin{aligned}
p(\boldsymbol{\mu}, \boldsymbol{\Sigma} \mid \mathbf{X}, \mathbf{z}) &\propto p(\mathbf{X} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{z}) p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\
&= \prod_{k}^{K} \mathrm{NIW}(\mu_k, \Sigma_k | \mathbf{m}_k, \kappa_k, \nu_k, \mathbf{S}_k) \\
&= \prod_{k}^{K} \mathrm{N}(\mu | \mathbf{m}_k, \frac{1}{\kappa_k}\Sigma) \times \mathrm{IW}(\Sigma | \mathbf{S}_k, \nu_k), \\
&\qquad \mathbf{m}_k = \frac{\kappa_{0_k} m_{0_k} + N_k \bar{\mathbf{x}}_k}{\kappa_{0_k} + N_k} \\
&\qquad \kappa_k = \kappa_0 + N_k \\
&\qquad \nu_k = \nu_0 + N_k \\
&\qquad \mathbf{S}_k = \mathbf{S}_{0_k} + \mathbf{S}_{\bar{\mathbf{x}}_k} + \frac{\kappa_{0_k} N_k}{\kappa_{0_k} + N_k} (\bar{\mathbf{x}}_k - m_{0_k})(\bar{\mathbf{x}}_k - m_{0_k})^T
\end{aligned}
\tag{3.10}
$$

where $N_k = |\mathbf{X}_k|$, $\bar{\mathbf{x}}_k = \frac{1}{N_k} \sum_{i:z_i=k} \mathbf{x}_i$, and $\mathbf{S}_{\bar{\mathbf{x}}_k} = \sum_{i:z_i=k} (\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^\top$.

### 3.4.2 Bernoulli Mixture Model (BMM)

BMMs are used to model discrete binary data that are multi-modal. A BMM assumes that each data-point $\mathbf{x}^{(i)}$ is generated from one of $K$ mixture components, where each component models the features of $\mathbf{x}^{(i)}$ as independent Bernoulli random variables. Given this, each component is parameterized by a vector $\boldsymbol{\theta}_k = \{\theta_{1,k}, \cdots, \theta_{D,k}\}$, where $\theta_{d,k}$ is the probability of success for a feature $x_d^{(i)}$ under component $k$. Given the latent component assignment $z^{(i)}$, the likelihood of $\mathbf{x}^{(i)}$ under component $k$ factorizes as:

$$
p(\mathbf{x}^{(i)} \mid z^{(i)} = k, \boldsymbol{\theta}_k) = \prod_{d=1}^{D} \mathrm{Bern}(x_d^{(i)} \mid \theta_{kd}) \tag{3.11}
$$

Marginalizing over the latent assignment, the complete mixture model is given by:

$$
p(\mathbf{x}^{(i)} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} \mathrm{Bern}(x_d^{(i)} \mid \theta_{kd}) \tag{3.12}
$$

With this, the observed data likelihood over all data points factorizes as follows

$$
p(\mathbf{X} \mid \boldsymbol{\theta}, \boldsymbol{\pi}) = \prod_{i}^{N} \sum_{k}^{K} \pi_k \prod_{d}^{D} \mathrm{Bern}(x_d^{(i)} \mid \theta_{kd}) \tag{3.13}
$$

In the Bernoulli case, we choose a Beta prior distribution for $\boldsymbol{\theta}$ over components $k$ and dimensions $d$, which is simply the binary case of the Dirichlet prior distribution used for $\boldsymbol{\pi}$.

$$
p(\boldsymbol{\theta}_k | \mathbf{a}_0, \mathbf{b}_0) = \prod_{d}^{D} \mathrm{Beta}(\theta_{d,k} \mid a_{0,d}, b_{0,d}) \tag{3.14}
$$

Here, $a_{0d}$ and $b_{0d}$ are the hyperparameters for the Beta prior for feature $d$, shared across components or set per component depending on prior knowledge. We choose a Beta prior as it is conjugate to the Bernoulli likelihood. This yields a closed-form posterior over Bernoulli biases $\boldsymbol{\theta}$ over dimensions $d$ and components $k$ after observing the data $\mathbf{X}$:

$$p(\theta_{kd} \mid \boldsymbol{X}, \mathbf{z}) = \text{Beta}(\theta_{kd} \mid a + N_{kd}^{(1)}, b + N_{kd}^{(0)}) \tag{3.15}$$

where $N_{kd}^{(0)} = \sum_i^N \mathbb{1}(x_d^{(i)} = 0, z_i = k)$ and $N_{kd}^{(1)} = \sum_i^N \mathbb{1}(x_d^{(i)} = 1, z_i = k)$.

### 3.4.3 Mixture Models for Clustering

With the primary goal being the discovery of latent clusters within our multi-modal datasets, the primary variable of interest is $\mathbf{z}$ in Equation 3.2. Mixture models naturally provide a framework for assigning data points to clusters using posterior probabilities, otherwise known as responsibility [24]. This contrasts other popular clustering approaches, such as K-Means, which provide hard deterministic assignments. Given model parameters and component weights, the responsibility for a data-point $\mathbf{x}^{(i)}$ for a component $k$ is given by

$$P(z^{(i)} = k \mid \mathbf{x}^{(i)}) = \frac{\pi_k P(\mathbf{x}^{(i)} \mid \boldsymbol{\phi}_k)}{\sum_j^K \pi_j P(\mathbf{x}^{(i)} \mid \boldsymbol{\phi}_j)} \tag{3.16}$$

## 3.5 Mixture Model Inference

Performing inference on mixture models ultimately presents a fundamental chicken-egg dilemma. We aim to estimate two sets of coupled unknowns that explain the observed data : the **model parameters** $\{\boldsymbol{\pi}, \boldsymbol{\phi}\}$, and the **latent cluster assignments** $z^{(i)}$ (represented via responsibilities $p(z^{(i)} \mid \mathbf{x}^{(i)})$). Each depends on the other where we need cluster assignments to estimate parameters accurately, and we need parameters to compute cluster assignments. The circular dependency is an underlying central challenge of inference in probabilistic mixture models.

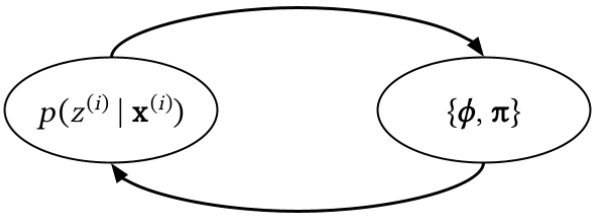



Figure 3.3: Circular dependency structure of mixture model inference

As noted in Equation 3.3, the fully Bayesian treatment offers a principled hierarchical posterior distribution over all unknowns, which in theory enables joint inference of both parameters and latent variables. However, in practice, performing fully Bayesian exact inference on the joint posterior is intractable due to the number of variables in the latent space, and the heterogeneous combination of distributions involved (e.g Dirichlet, Categorical, Gaussian/Bernoulli). This means that there is no closed-form solution that can be derived through conjugacy between the different distributions involved in the posterior. Consequently, computing the posterior for exact inference requires evaluating the marginal likelihood $P(\mathbf{X})$,

as in equation 3.3, which serves as the normalizing constant. Take, for instance, the GMM case where we have the following marginal likelihood

$$p(\mathbf{X}) = \int\int\int \sum_{\mathbf{z}} \Big[ \prod_{i}^{N} p(\mathbf{x}^{(i)}|z^{(i)}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) p(z^{(i)} = k|\boldsymbol{\pi}) \Big] \, p(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k) p(\boldsymbol{\Sigma}_k) p(\boldsymbol{\pi}) \, d\boldsymbol{\mu} d\boldsymbol{\Sigma} d\boldsymbol{\pi} \tag{3.17}$$

To compute the marginal likelihood, we have to enumerate all possible satisfying assignments of the unknown variables, a problem that grows exponentially with the number of data-points and mixture components. Performing exact inference in this case is *at least* an NP-Hard problem [25, 33].

To address this, it is necessary to turn to **approximate inference** algorithms which seek to approximate intractable distributions as tightly as possible. This research utilizes two approximate inference techniques for posterior estimation : **Gibbs Sampling** and **Variational Inference**.

### 3.5.1 Gibbs Sampling

Gibbs sampling is a type of Markov Chain Monte Carlo (MCMC) inference algorithm that provides a framework for approximating complex joint probability distributions by iteratively sampling from simpler full conditional (posterior) distributions of each variable given the others [26]. Instead of attempting to sample directly from the intractable full joint posterior, Gibbs sampling breaks the inference problem into smaller, more manageable steps. At each iteration, a single variable (or block of variables) is sampled from its full conditional distribution while treating all other variables as observed. As this process is repeated, the sequence of samples forms a Markov Chain whose stationary distribution converges to the target joint distribution [12]. The empirical distribution of these samples approximates the posterior and captures uncertainty over the variables of interest. This method naturally avoids the need for closed-form posterior expressions.

For inferring a mixture model, Gibbs sampling has the following procedure:

1. Initialize unknown variables $\mathbf{z}, \boldsymbol{\phi}, \boldsymbol{\pi}$
2. For $t \in 1, ..., T$ iterations

   Sample $\mathbf{z}^{(t)} \sim p(\mathbf{z}^{(t)} \mid \mathbf{X}, \boldsymbol{\pi}^{(t-1)}, \boldsymbol{\phi}^{(t-1)})$
   Sample $\boldsymbol{\pi}^{(t)} \sim p(\boldsymbol{\pi}^{(t)} \mid \mathbf{z}^{(t)})$
   Sample $\boldsymbol{\phi}^{(t)} \sim p(\boldsymbol{\phi}^{(t)} \mid \mathbf{X}, \mathbf{z}^{(t)})$

3. Discard some number of samples ($\sim 25\%$) as burn in.

The result is a practical algorithm that enables approximate posterior inference by sampling from known conditional distributions, even when the joint distribution is analytically intractable.

Gibbs sampling does come with some difficulties. For one, the number of iterations must be configured to ensure the algorithm eventually reaches a stationary distribution (convergence)

which therefore approximates the target posterior. Another closely coupled difficulty is determining the number of samples to burn where ideally all remaining samples are collected after the Markov chain has reached the stationary distribution. This is due to the fact that we typically randomly initialize variables which initially produces poor samples.

Once convergence is achieved and burn-in is completed, Gibbs sampling produces a collection of samples from the posterior distribution. These samples can be treated as an empirical approximation of the true posterior. However, in models such as mixture models, Gibbs sampling suffers from a foundational weakness known as the **label switching** problem which makes it difficult to summarize the resulting empirical posterior distribution [36]. In particular, the parameters and cluster assignments are unidentifiable, meaning we can arbitrarily permute the hidden labels without affecting likelihood [26]. As a result, different samples may assign different labels to the same underlying component. In other words, what one sample might consider to be "component 1" may not be what another sample (or the ground truth data) considers to be "component 1", despite each sample being internally coherent. This makes it difficult to summarize the posterior, especially when computing posterior means of parameters or responsibilities since averaging over misaligned labels yield meaningless results.

In clustering tasks, it is common to avoid directly comparing raw cluster labels, since they are inherently unidentifiable [26]. For example, rather than look at classification accuracy, a common alternative is to consider **Adjusted Random Index** which computes the similarity between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings [1]. For selecting a representative clustering from the samples, a common statistically principled strategy is to take the sample with the highest posterior probability, which serves as an approximate MAP estimate [26].

However, for model parameters, we often want to obtain posterior means rather than rely on a single sample (unlike cluster assignments which are discrete). For this objective, it is therefore necessary to apply a label alignment strategy across samples to allow for coherent averaging. A common unsupervised alignment strategy is the **Hungarian Algorithm**.

### Hungarian Algorithm

The Hungarian algorithm solves the assignment by finding the cheapest path through a cost matrix [38]. As such, for two different samples $A$ and $B$, we can define a cost matrix $\mathbf{C}$ where each entry $\mathbf{C}_{i,j}$ denotes the number of data points assigned to label $i$ in sample $B$ and label $j$ in sample $A$. The cost function is equivalent to maximizing the number of matching labels (i.e. minimizing the Hamming distance [21]), implemented here as the negative of the per-cluster match counts. This can be viewed as a form of confusion matrix between two sets of assignments. Then by identifying the cheapest path through the matrix, the algorithm can determine the optimal permutation to re-label one sample to match the other sample [38].

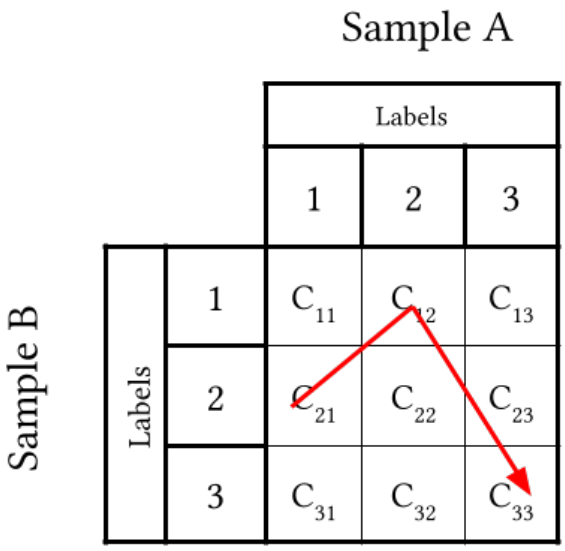



Figure 3.4: Simple example of optimal path through cost matrix between two samples $A$ and $B$ in a scenario consisting of three components where the red arrow denotes cheapest path. In this example, the optimal path indicates that sample $B$ can be aligned to sample $A$ by re-labeling $2 \rightarrow 1$, $1 \rightarrow 2$, and $3 \rightarrow 3$

### 3.5.2 Variational Inference

Variational Inference (VI) is another approximation algorithm that frames the approximation task as an optimization problem. To do so, it first introduces an approximate distribution $q$ from some tractable family with some simplifying assumptions, where the goal is to optimize the free parameters of $q$ to minimize the Kullback-Leibler (KL) divergence between $q$ and the true posterior distribution $p$ [27]. As such, the KL-divergence serves as a cost function to minimize the dissimilarity between the two distributions.

$$\mathbb{KL}(q \mid\mid p) = \int q(\mathbf{Z}) \log \frac{q(\mathbf{Z})}{p(\mathbf{Z} \mid \mathbf{X})} \, d\mathbf{Z} \tag{3.18}$$

Where $\mathbf{Z} = \{z_1, \cdots, z_n\}$ represents all unknown variables. This quantity is intractable to compute directly due to the marginal likelihood $p(\mathbf{X})$ in $p(\mathbf{Z} \mid \mathbf{X}) = \frac{p(\mathbf{X},\mathbf{Z})}{p(\mathbf{X})}$. To proceed, we can instead define and optimize the Evidence Lower Bound (ELBO) which will be denoted as $\mathcal{L}$. This is defined as follows

$$\begin{aligned} \mathcal{L}(q) &= \log p(\mathbf{X}) - \mathbb{KL}(q \mid\mid p) \\ &= \mathbb{E}_{q(\mathbf{Z})}[\log p(\mathbf{X}, \mathbf{Z})] - \mathbb{E}_{q(\mathbf{Z})}[\log q(\mathbf{Z})] \\ &= \log p(\mathbf{X}) - \mathrm{KL}(q(\mathbf{Z}) \parallel p(\mathbf{Z} \mid \mathbf{X})) \end{aligned} \tag{3.19}$$

Where,

$$\begin{aligned} q* &\leftarrow \underset{q}{\operatorname{argmax}} \{ \underbrace{\log p(\mathbf{X})}_{\text{constnt}} - \mathrm{KL}(q(\mathbf{Z}) \parallel p(\mathbf{Z} \mid \mathbf{X})) \} \\ &= \underset{q}{\operatorname{argmax}} \{ -\mathrm{KL}(q(\mathbf{Z}) \parallel p(\mathbf{Z} \mid \mathbf{X})) \} \\ &= \underset{q}{\operatorname{argmin}} \{ \mathrm{KL}(q(\mathbf{Z}) \parallel p(\mathbf{Z} \mid \mathbf{X})) \} \end{aligned} \tag{3.20}$$

Here we can see that maximizing $\mathcal{L}$ is equivalent to minimizing $\mathbb{KL}(q \mid\mid p)$ since $\log p(\mathbf{X})$ is constant, however, we avoid having to compute the intractable posterior $p(\mathbf{X}|\mathbf{Z}) = \frac{p(\mathbf{X}|\mathbf{Z})}{p(\mathbf{X})}$. Thus the optimal variational distribution $q$ is

$$\hat{q} = \arg\max_{q} \mathcal{L}(q) \tag{3.21}$$

For mixture model inference, this research utilizes a form of VI called **Variational Bayes Expectation Maximization (VBEM)**. This algorithm constitutes a coordinate ascent algorithm to optimize (maximize) the ELBO function $\mathcal{L}$ and thereby minimize $\mathbb{KL}(q \mid\mid p)$ [27]. To setup VBEM, we make a simplifying assumption for the variational posterior such that is factorizes in the following manner

$$q(\boldsymbol{\phi}, \mathbf{Z}|\mathbf{X}) = q(\boldsymbol{\phi}) \prod_{i} q(z^{(i)}) \tag{3.22}$$

This is a **mean-field** assumption that assumes independence between parameters and latent variables across data points.

In VBEM we maximize the ELBO by alternating between update steps of the latent variable distribution $q(\mathbf{z}|\mathbf{X})$ and parameter distribution $q(\boldsymbol{\phi})$, holding the other fixed in each step. This consists of alternating between a variational expectation step (E-step), and a variational maximization step (M-step):

- **Variational E-Step** : Holding $q(\boldsymbol{\phi}|\mathbf{X})$ fixed, we perform the following optimal update using the expectation of the joint log-likelihood with respect to the current fixed variational distribution over parameters.

$$q(z^{(i)}) \propto \exp\left(\mathbb{E}_{q(\boldsymbol{\phi})}\Big[\log p(\mathbf{x}^{(i)}, z^{(i)}, \boldsymbol{\phi})\Big]\right) \tag{3.23}$$

  The updated $q(\mathbf{z}|\mathbf{X})$ gives us our mixture model responsibilities for the current iteration.

- **Variational M-Step** : Holding $q(\mathbf{z}|\mathbf{X})$ as fixed, use the updated responsibilities to compute expected sufficient statistics to update the (hyperparameters of) variational posterior over parameters $q(\boldsymbol{\phi})$.

$$q(\boldsymbol{\phi}) \propto \exp\left(\mathbb{E}_{q(\mathbf{z})}\Big[\log p(\mathbf{x}^{(i)}, z^{(i)}, \boldsymbol{\phi})\Big]\right) \tag{3.24}$$

Each iteration of VBEM increases the ELBO (or leaves it unchanged), where the procedure eventually converges to an optimum of the ELBO [27]. In practice, the VBEM algorithm is initialized with random hyperparameter values for the variational posteriors, which initially will lead to a suboptimal ELBO. The repeated variational E-steps and M-steps iteratively improve the hyperparameter estimates, increasing the ELBO until convergence.

VBEM can converge at a local optimum and is sensitive to initialization of parameters [5]. This is primarily due to the non-convex shape of the ELBO under the mean-field assumption, and the multi-modality of the mixture model posterior. The primary consequence is that there can be multiple different but equally likely explanations for the same data [28]. One of the primary causes of this multi-modality is the fact that label assignments are invariant to

permutations of component labels [24] (the same label-switching issue as seen with Gibbs). For a K-component mixture will have a total of K! equivalent modes of the posterior corresponding to the K! different ways of assigning K sets of parameters to K components [6]. VBEM minimizes the reverse KL-divergence, which is known to be mode seeking [27]. As such, it tends to converge to one of these posterior modes which may not be guaranteed to be the globally optimal.

This motivates the use of multiple random restarts during inference in order to improve chances of finding an optimal solution. There are multiple strategies for choosing the best restart. If the ground truth labels are provided, the choice is trivially the random restart that provides the highest clustering performance. In completely unsupervised settings, a common strategy is to choose the run with the highest ELBO. As with Gibbs, the final solution is likely permuted re-alignment is necessary to directly compared against the ground truth.

### 3.5.3 Expectation Maximization (EM) Algorithm

The Expectation Maximization (EM) algorithm provides a frequentist alternative to fully Bayesian inference, offering an iterative procedure for computing maximum likelihood (MLE) (or maximum a posteriori (MAP)) estimates in the presence of latent variables. Unlike VBEM, which approximates the intractable posterior via variational distributions, EM computes point estimates of the model parameters $\boldsymbol{\phi}$ directly. These estimates are used to infer the latent structure of the data via posterior responsibilities [28].

The goal of the EM algorithm is to obtain estimates for latent variables and parameters that maximize the observed data likelihood. In mixture models, direct maximization of the observed data likelihood $p(\mathbf{X} \mid \boldsymbol{\phi})$ is intractable due to latent component assignments $\mathbf{z}$ [6]. EM instead maximizes the expected complete-data log-likelihood (CDLL), which assumes both data $\mathbf{X}$ and latent variables $\mathbf{z}$ are known:

$$\ln p(\mathbf{X}, \mathbf{z} \mid \boldsymbol{\phi}) = \ln p(\mathbf{X} \mid \mathbf{z}, \boldsymbol{\phi}) + \ln p(\mathbf{z} \mid \boldsymbol{\phi}) \tag{3.25}$$

However, since the component assignments $\mathbf{z}$ are in reality unobserved, we cannot compute the CDLL directly. Instead, we compute its expectation under the posterior distribution of $\mathbf{z}$, given the observed data $\mathbf{X}$ and current parameter estimates [6]. This quantity is called the expected complete-data log-likelihood, and is the target of optimization in the EM algorithm. It is formally defined as:

$$Q(\boldsymbol{\phi} \mid \boldsymbol{\phi}^{(t-1)}) = \mathbb{E}_{p(\mathbf{z}|\mathbf{X},\boldsymbol{\phi}^{(t-1)})} \left[\log p(\mathbf{X}, \mathbf{z} \mid \boldsymbol{\phi})\right] \tag{3.26}$$

where the expectation is taken with respect to the posterior over $\mathbf{z}$ under the current parameter estimates $\boldsymbol{\phi}^{(t-1)}$.

The EM algorithm then alternates between two steps:

- **E-step** : Evaluate the posterior distribution over the latent variables $p(\mathbf{z} \mid \mathbf{X}, \boldsymbol{\phi}^{(t-1)})$ using the current parameter estimates and compute the expected CDLL [28]:

$$Q(\boldsymbol{\phi}|\boldsymbol{\phi}^{(t-1)}) = \mathbb{E}_{p(\mathbf{z}|\mathbf{X},\boldsymbol{\phi}^{(t-1)})}\Big[\log p(\mathbf{X}, \mathbf{z} \mid \boldsymbol{\phi})\Big] \tag{3.27}$$

- **M-step** : Maximize the $Q$ function with respect to the model parameters to obtain updated estimates [28]:

$$\boldsymbol{\phi}^{(t)} = \underset{\boldsymbol{\phi}}{\operatorname{argmax}}\, Q(\boldsymbol{\phi}|\boldsymbol{\phi}^{(t-1)}) \tag{3.28}$$

  Here

  - $\boldsymbol{\phi}^{(t)}$ denotes the new updated parameters.

  Note : this is MLE estimation rather than MAP

This procedure is guaranteed to monotonically increase (or maintain) the observed-data log-likelihood at each iteration, as it maximizes a lower bound on it, similar to VBEM. However, unlike VBEM, which maintains distributions over the model parameters, the EM algorithm performs point estimation by directly maximizing the expected CDLL using current parameter estimates. In the VBEM framework, expectations of the joint log-likelihood are taken with respect to the variational distributions over both the latent variables and the model parameters, rather than relying on fixed point estimates. In the variational M-step, VBEM updates the parameters of the variational distribution over the model parameters to maximize the ELBO. Thus, EM can be viewed as the frequentest analogue of VBEM and serves as a non-fully-Bayesian benchmark in our implementation.

In practice, the EM algorithm is initialized with random parameter values, which leads to a suboptimal expected CDLL. Repeated E-steps and M-steps iteratively improve the parameter estimates, increasing the expected CDLL until convergence. Due to the non-convex nature of the likelihood function and exhibits multiple competing modes (due to label symmetry), the EM algorithm may converge to a local optimum, motivating the use of multiple random restarts [28].

# Chapter 4

# Software Engineering Practice

The nature of this project lent itself well to a standard agile iterative and incremental development process as outlined in the objectives section. Due to the experimental and exploratory nature of this project, the initial development process was conducted primarily in `Python` notebooks which provided a flexible environment for prototyping algorithms alongside their mathematical derivations using embedded `Markdown`. This approach allowed for rapid testing, debugging, and refinement of each component in isolation.

The development cycle of each algorithm consisted first of a learning phase, a theoretical derivation phase to formalize the formulae of each algorithm while incorporating missing data treatment, finally followed by implementation. For most algorithms, a version assuming complete data was first implemented and validated to ensure correctness. This baseline was then extended to support incomplete data, allowing for careful evaluation of the modifications introduced by the missing data treatment.

Each algorithm was implemented over the course of one to two week sprints. Supervisor meetings were held on a weekly basis which served to reinforce newly learned concepts, verify the correctness of the work implemented before preceding to the next sprint, and to assess the ongoing direction of the research while organizing priorities for upcoming tasks.

After successfully implementing and testing each algorithm in notebooks, an object oriented class structure was created for the algorithms to abstract away their complexity into self-contained objects as to improve code re-use and streamline downstream usage and evaluation. The overall design of the model classes was to mimic that of popular machine-learning libraries such as `Scikit-Learn` with `fit` and `predict` methods.

# Chapter 5

# Ethics

This research has no major ethical concerns. An ethics application for the use of the Public Whip MP Voting dataset [31] was submitted to the School of Computer Science Ethics Committee and was approved (See Appendix 11.1).

# Chapter 6

# Implementation

This project explores a fully Bayesian approach to inference with missing data where missing values are not imputed before analysis, but rather treated as latent variables and jointly inferred within the model. Thus, missing variables are fully integrated into posterior inference. In this approach, we model uncertainty over the missing data, the model parameters, and latent variables, which avoids the decoupling between an imputation and analysis stage.

In this section, we discuss the details regarding the formulation of the mixture model posterior with missing data in both the Gaussian and Bernoulli case, as well as the various approaches and details for performing inference on this joint posterior using Gibbs Sampling, the EM algorithm, and the VBEM algorithm.

## 6.1 Mixture Model with Missing Data

With missing data, the posterior for a general mixture model can be formulated as follows

$$p(\mathbf{X}_h, \mathbf{z}, \boldsymbol{\phi}, \boldsymbol{\pi} \mid \mathbf{X}_o) \propto p(\mathbf{X}_o, \mathbf{X}_h \mid \mathbf{z}, \boldsymbol{\phi})\, p(\boldsymbol{\phi})\, p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi}) \tag{6.1}$$

- $\mathbf{X}_o$ : the observed values of the data $\mathbf{X}$
- $\mathbf{X}_h$ : the missing (*hidden*) values of the data $\mathbf{X}$.

Here we have introduced $\mathbf{X}_h$ as a random variable denoting the missing parts of the data where $\mathbf{X}_h = \{\mathbf{x}_{ih}\}_i^N$. The data-likelihood component of the posterior, $p(\mathbf{X}_o, \mathbf{X}_h \mid \mathbf{z}, \boldsymbol{\phi})$, is the joint likelihood of the observed and missing parts of the data and can be further factorized in the Bernoulli and Gaussian case.

### 6.1.1 BMM : Likelihood with Missing Data

In the BMM case, we make the assumption that the features of each data-point are independent Bernoulli variables. In particular, for a data-point $\mathbf{x}_i$,

$$p(\mathbf{x}_i \mid \boldsymbol{\theta}_k, z_i = k) = \prod_d^D \text{Bern}(x_i^d \mid \theta_{k,d})$$

where $\theta_{k,d}$ represents the success probability (probability of observing a 1) for feature $d$ in component $k$, and $x_i^d \in \{0, 1\}$ denotes the value of feature $d$ of observation $i$. This independence assumption is standard and natural for multivariate Bernoulli models. It is worth noting that feature dependencies in binary data can be incorporated into mixture modeling (e.g. the Ising model [22]), but these approaches introduce complexities beyond the scope of this research.

With this assumption, the data-likelihood component of the BMM posterior factorizes as

$$p(\mathbf{X}_o, \mathbf{X}_h \mid \mathbf{z}, \boldsymbol{\phi}) = p(\mathbf{X}_o \mid \boldsymbol{\theta}, \mathbf{z})\, p(\mathbf{X}_h \mid \boldsymbol{\theta}, \mathbf{z}) = \prod_i^N p(\mathbf{x}_{io} \mid \boldsymbol{\theta}, z_i)\, p(\mathbf{x}_{ih} \mid \boldsymbol{\theta}, z_i) \tag{6.2}$$

Thus, the observed and missing data components are conditionally independent given the component assignment and the component model parameters. Assuming a Beta prior over the model parameters, the full posterior for the BMM case is as follows :

$$p(\mathbf{X}_h, \mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\pi} \mid \mathbf{X}_o) \propto p(\mathbf{X}_o \mid \boldsymbol{\theta}, \mathbf{z})\, p(\mathbf{X}_h \mid \boldsymbol{\theta}, \mathbf{z})\, p(\boldsymbol{\theta} \mid \mathbf{a}, \mathbf{b})\, p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) \tag{6.3}$$

### 6.1.2 GMM : Likelihood With Missing Data

The multivariate Gaussian distribution is parameterized by both a mean parameter $\mu$ as well as a covariance parameter $\Sigma$ whose diagonal entries are variances and off-diagonals are pairwise covariances capturing linear dependence among features. Consequently, features of the data are not conditionally independent where the missing parts of each data-point may be dependent on the observed parts. Therefore in the context of Gaussian mixture modeling with missing data, the data-likelihood component of the mixture model posterior factorizes as

$$\begin{aligned} p(\mathbf{X}_o, \mathbf{X}_h \mid \mathbf{z}, \boldsymbol{\phi} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}) &= p(\mathbf{X}_o \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{z})\, p(\mathbf{X}_h \mid \mathbf{X}_o, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{z}) \\ &= \prod_i^N p(\mathbf{x}_{io} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, z_i)\, p(\mathbf{x}_{ih} \mid \mathbf{x}_{io}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, z_i) \end{aligned} \tag{6.4}$$

The covariance matrix $\Sigma$ thus serves as a built-in mechanism for modeling feature dependencies, enabling the model to make principled, data-informed imputations of missing entries based on the relationships it learns from the observed dimensions. This expresses the fact that the likelihood of the missing features depends on both the component assignment and the observed features through the multivariate Gaussian conditional distribution. Given the component assignment of a data-point, this conditional distribution, $p(\mathbf{x}_h^{(i)} \mid \mathbf{x}_o^{(i)}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is given by the following [29]

$$\begin{aligned} p(\mathbf{x}_{ih} \mid \mathbf{x}_{io}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) &= \mathcal{N}(\mathbf{x}_{ih} \mid \mathbf{m}_{ik}^{h|o}, \mathbf{V}_{ik}^{h|o}) \\ \mathbf{m}_{ik}^{h|o} &= \boldsymbol{\mu}_k^h + \boldsymbol{\Sigma}_k^{ho} \boldsymbol{\Sigma}_k^{oo-1} (\mathbf{x}_{io} - \boldsymbol{\mu}_k^o) \\ \mathbf{V}_{ik}^{h|o} &= \boldsymbol{\Sigma}_k^{hh} - \boldsymbol{\Sigma}_k^{ho} \boldsymbol{\Sigma}_k^{oo-1} \boldsymbol{\Sigma}_k^{oh} \end{aligned} \tag{6.5}$$

where

- $\boldsymbol{\mu}_k^h$ and $\boldsymbol{\mu}_k^o$ denote the sub-vectors of the component mean $\boldsymbol{\mu}_k$ corresponding to the missing and observed dimensions, respectively

- $\boldsymbol{\Sigma}_k^{hh}$ denotes the covariance matrix over the missing dimensions
- $\boldsymbol{\Sigma}_k^{oo}$ denotes the covariance matrix over the observed dimensions.
- $\boldsymbol{\Sigma}_k^{oh}$ and $\boldsymbol{\Sigma}_k^{ho}$ are the cross-covariance matrices between missing and observed features

This conditional structure allows the model to use the observed parts of each data-point to infer the distribution of its missing entries. This is one of the main advantages of the Gaussian mixture case when handling missing data as the imputation is guided not just by marginal statistics (as is the case with BMMs), but by relationships between features.

This conditional distribution allows computing **expected sufficient statistics** which are necessary for the EM and VBEM parameter updates. For a data-point consisting of both observed and missing parts $\mathbf{x}_i = \{\mathbf{x}_{io}, \mathbf{x}_{ih}\}$, the expected sufficient statistics are as follows

$$\mathbb{E}_k[\mathbf{x}_i] = (\mathbb{E}_k[\mathbf{x}_{ih}]; \mathbf{x}_{io}) = (\mathbf{m}_i^{h|o}; \mathbf{x}_o^{(i)}) \tag{6.6}$$

$$\mathbb{E}_k[\mathbf{x}_i \mathbf{x}_i^T] = \begin{bmatrix} \mathbb{E}_k[\mathbf{x}_{ih}\mathbf{x}_{ih}^T] & \mathbb{E}_k[\mathbf{x}_{ih}]\mathbf{x}_{io}^T \\ \mathbf{x}_{io}\mathbb{E}_k[\mathbf{x}_{ih}]^T & \mathbf{x}_{io}\mathbf{x}_{io}^T \end{bmatrix} \tag{6.7}$$

$$\mathbb{E}_k[\mathbf{x}_{ih}\mathbf{x}_{ih}^T] = \mathbb{E}[\mathbf{x}_{ih}]\mathbb{E}[\mathbf{x}_{ih}]^T + \mathbf{V}_i^{h|o} \tag{6.8}$$

These expectations ensure that the contributions of missing values are handled in a principled probabilistic manner.

## 6.2 Gibbs Sampling

As mentioned in Section 3.5.1, the Gibbs sampling algorithm works by iteratively sampling from the full conditionals of each unknown variable in the joint posterior while holding all other variables as fixed. In this section, we outline the details of the algorithm for the complete data and missing data cases for both BMMs and GMMs.

### 6.2.1 BMM Approach

#### Complete Data Approach

With complete data, the full joint distribution for the BMM case factors into the following

$$p(\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\pi}, \mathbf{X}) = p(\mathbf{X} \mid \boldsymbol{\theta}, \mathbf{z})\, p(\boldsymbol{\theta} \mid \mathbf{a}_0, \mathbf{b}_0)\, p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}_0) \tag{6.9}$$

By leveraging conjugacy, as outlined in Section 3.4, we can ensure that the full conditional posterior distributions we aim to sample from have closed-form expressions. This facilitates more efficient sampling during each iteration of the Gibbs algorithm. As such, we have the following full conditionals for each variable

1. Mixing weights $\boldsymbol{\pi}$

$$p(\boldsymbol{\pi} \mid \mathbf{z}) = \text{Dir}(\boldsymbol{\pi} \mid \alpha_{0,1} + N_1, \cdots, \alpha_{0,K} + N_K)$$

2. Bernoulli bias $\theta_{k,d}$ for each component $k$ and dimension $d$

$$p(\theta_{kd} \mid \boldsymbol{X}, \mathbf{z}) = \text{Beta}(\theta_{kd} \mid a_{0d} + N_{kd}^{(1)}, b_{0d} + N_{kd}^{(0)})$$

3. Latent component assignments for each for each data-point $i$ and component $k$

$$p(z_i = k \mid \mathbf{x}_i, \boldsymbol{\theta}, \boldsymbol{\pi}) = \pi_k \prod_d^D \left[\theta_{kd}^{x_{id}} (1 - \theta_{kd})^{(1-x_{id})}\right]$$

Normalizing over all component gives responsibilities for the sampling update

$$r_{ik} = \text{Cat}(z_i = k \mid \mathbf{x}_i, \boldsymbol{\theta}, \boldsymbol{\pi}) = \frac{\pi_k \prod_d^D \left[\theta_{kd}^{x_{id}} (1 - \theta_{kd})^{(1-x_{id})}\right]}{\sum_k^K \pi_j \left[\prod_d^D \theta_{jd}^{x_{id}} (1 - \theta_{jd})^{(1-x_{id})}\right]}$$

With full conditionals for all of the variables in the joint distribution, we perform the following sampling steps at each iteration $t$ of the Gibbs sampler:

$$\text{Sample } \boldsymbol{\pi}^{(t)} \sim \text{Dir}\left(\alpha_{0,1} + N_1^{(t-1)}, \ldots, \alpha_{0,K} + N_K^{(t-1)}\right)$$

$$\text{Sample } \theta_{kd}^{(t)} \sim \text{Beta}\left(a_{0,k} + N_{kd}^{(1)},\ b_{0,k} + N_{kd}^{(0)}\right), \quad \text{for all } k \in [K],\ d \in [D]$$

$$\text{Sample } z_i^{(t)} \sim \text{Categorical}\left(p(z_i = k \mid \mathbf{x}_i, \boldsymbol{\theta}^{(t)}, \boldsymbol{\pi}^{(t)})\right), \quad \text{for all } i \in [N]$$

where:

- $N_k^{(t-1)} = \sum_{i=1}^N \mathbb{1}(z_i^{(t-1)} = k)$ is the number of data points assigned to component $k$ in the previous iteration.
- $N_{kd}^{(1)} = \sum_{i=1}^N \mathbb{1}(x_{id} = 1, z_i^{(t)} = k)$, the number of times feature $d$ is 1 in component $k$.
- $N_{kd}^{(0)} = \sum_{i=1}^N \mathbb{1}(x_{id} = 0, z_i^{(t)} = k)$, the number of times feature $d$ is 0 in component $k$.

Each Gibbs iteration proceeds by first updating the global parameters $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ conditioned on the current cluster assignments, followed by resampling the cluster assignment for each data-point conditioned on the updated parameters. After sufficient burn-in iterations, the collected samples from the chain approximate the true posterior distribution over $\{\mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\theta}\}$.

### Missing Data Treatment

To handle missing data within the Gibbs sampling framework, we work directly with the full joint posterior that includes the missing entries $\mathbf{X}_h$ as latent variables to be inferred, as given in Equation 6.3. Under the assumption of conditional independence, the data likelihood naturally factorizes into observed and missing components: $p(\mathbf{X}_o \mid \boldsymbol{\theta}, \mathbf{z}), p(\mathbf{X}_h \mid \boldsymbol{\theta}, \mathbf{z})$. This structure permits a straightforward extension of the Gibbs sampler to incorporate missing values. From here, there are two approaches for doing so within this fully Bayesian framework.

The natural approach is to, at each iteration, sample and impute the missing entries of the data from its full conditional. This can be referred to as a data augmentation approach

[13], where given the current cluster assignment and Bernoulli parameters, we sample each missing entry $x_{ih}^{d}$ for data-point $i$ and dimension $d$ as follows

$$x_{id}{}^{(t)} \sim \text{Bern}(\theta_{z_i,d}^{(t-1)}), \quad \text{if } x_{id} \text{ is missing} \tag{6.10}$$

Here, each value is drawn from the component-specific Bernoulli distribution according to the current cluster assignment of the corresponding data-point. The remaining sampling steps proceed normally, treating the imputed data as complete. This approach follows the standard Gibbs treatment of hidden variables by including them into the approximated posterior, allowing for explicit quantification of uncertainty in the missing entries.

An alternative approach, particularly well-suited to BMMs, is to marginalize over the missing data whereby we only use the observed entries to inform and update our conditional distributions. Due to the feature-wise independence assumption inherent to the multivariate Bernoulli model, the data likelihood factorizes across dimensions. This makes it straightforward to exclude missing entries when computing likelihoods, enabling inference to proceed using only the observed values. In this approach, the sampling of component assignments $\mathbf{z}$ and model parameters $\boldsymbol{\theta}$ (the two variables that depend on the data $\mathbf{X}$) are as follows

$$\text{Sample } z_i^{(t)} \sim p(z_i \mid \mathbf{x}_{io}, \boldsymbol{\pi}, \boldsymbol{\theta}), \quad \text{For } i \in 1, ..., N \tag{6.11}$$

$$\text{Sample } \boldsymbol{\theta}_k^{(t)} \sim p(\boldsymbol{\theta}_k \mid \mathbf{X}_o, \mathbf{z}), \quad \text{For } k \in 1, ..., K \tag{6.12}$$

This marginalization strategy effectively collapses out the missing entries, removing them from the latent variable set. Although the final posterior over model parameters and cluster assignments remains unchanged, marginalizing over the missing data reduces the dimensionality of the posterior and can lead to lower-variance estimates [30]. As a result, the Gibbs sampler often converges more rapidly and requires fewer samples to reach a stationary point [11, 13]. This ultimately results in a more efficient sampler, requiring fewer samples to reach convergence and often reducing computational complexity [11].

In fact, the Gibbs sampler can be further collapsed by integrating out additional variables such as component parameters and mixing weights. This process is known as Rao-Blackwellisation [30, 13]. However, the process of collapsing the Gibbs sampler may not always be easy to implement where the collapsed posterior distribution may be harder to work with [13]. Therefore the choice to do so is dependent on whether integrating out variables can be done quickly [30]. In this project, we opt to retain variables such as $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ in the posterior distribution, as they are of substantive interest for understanding the data generation process and downstream predictive tasks. Therefore, we strike a balance between computational efficiency and model interpretability by marginalizing over missing entries but sampling the core model parameters.

**BMM Gibbs Sampling Algorithms**

**Algorithm 1** Gibbs Sampling for BMM with Complete Data

**Input:** Data $\mathbf{X}$ with missing values, number of clusters $K$

**Initialize:**

Hyperparameters : $\boldsymbol{\alpha_0}, \mathbf{a}_0, \mathbf{b_0}$

Variables : $\boldsymbol{\theta}^{(0)}$, $\boldsymbol{\pi}^{(0)}$, $\mathbf{z}^{(0)}$

**for** $t = 1$ to $T$ **do**

Sample $\boldsymbol{\pi}^{(t)} \sim \text{Dirichlet}(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = \{\alpha_{0,1} + N_1, ..., \alpha_{0,K} + N_K\}$

**for** $i = 1$ to N **do**

Sample $z_i^{(t)} \sim p(z_i \mid \mathbf{x}_i, \boldsymbol{\theta}^{(t-1)}, \boldsymbol{\pi}^{(t-1)})$

**end for**

**for** $k = 1$ to $K$ and $d = 1$ to $D$ **do**

Sample $\theta_{kd}^{(t)} \sim \text{Beta}(a_{0,k} + N_{kd}^{(1)}, b_{0,k} + N_{kd}^{(0)})$

**end for**

**end for**

**return** Samples

**Algorithm 2** Gibbs Sampling for BMM with Missing Data

**Input:** Data $\mathbf{X}$ with missing values, number of clusters $K$

**Initialize:**

Hyperparameters : $\boldsymbol{\alpha_0}, \mathbf{a}_0, \mathbf{b_0}$

Variables : $\boldsymbol{\theta}^{(0)}$, $\boldsymbol{\pi}^{(0)}$, $\mathbf{z}^{(0)}$

**for** $t = 1$ to $T$ **do**

Sample $\mathbf{X}_h^{(t)} \sim p(\mathbf{X}_h \mid \mathbf{z}^{(t-1)}, \boldsymbol{\theta}^{(t-1)})$

Sample $\boldsymbol{\pi}^{(t)} \sim \text{Dirichlet}(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = \{\alpha_{0,1} + N_1, ..., \alpha_{0,K} + N_K\}$

**for** $i = 1$ to N **do**

Sample $z_i^{(t)} \sim p(z_i \mid \mathbf{x}_i^{(t)}, \boldsymbol{\theta}^{(t-1)}, \boldsymbol{\pi}^{(t-1)})$

**end for**

**for** $k = 1$ to $K$ and $d = 1$ to $D$ **do**

Sample $\theta_{kd}^{(t)} \sim \text{Beta}(a_{0,k} + N_{kd}^{(1)}, b_{0,k} + N_{kd}^{(0)})$ where $N_{nk}^{(v)} = \mathbb{1}(x_{kd}^{(t)} = v, z_i = k)$

**end for**

**end for**

**return** Samples

**Algorithm 3** Collapsed Gibbs Sampling for BMM with Missing Data

**Input:** Data $\mathbf{X}$ with missing values, number of clusters $K$

**Initialize:**

Hyperparameters : $\boldsymbol{\alpha_0}, \mathbf{a}_0, \mathbf{b_0}$

Variables : $\boldsymbol{\theta}^{(0)}$, $\boldsymbol{\pi}^{(0)}$, $\mathbf{z}^{(0)}$

**for** $t = 1$ to $T$ **do**

Sample $\boldsymbol{\pi}^{(t)} \sim \text{Dirichlet}(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = \{\alpha_{0,1} + N_1, ..., \alpha_{0,K} + N_K\}$

**for** $i = 1$ to N **do**

Sample $z_i^{(t)} \sim p(z_i \mid \mathbf{x}_{io}, \boldsymbol{\theta}^{(t-1)}, \boldsymbol{\pi}^{(t-1)})$

**end for**

**for** $k = 1$ to $K$ and $d = 1$ to $D$ **do**

Sample $\theta_{kd}^{(t)} \sim \text{Beta}(a_{0,k} + N_{kdo}^{(1)}, b_{0,k} + N_{kdo}^{(0)})$ where $N_{nko}^{(v)} = \mathbb{1}(x_{kd}^{(t)} = v, z_i = k, d \in o)$

**end for**

**end for**

**return** Samples

### 6.2.2 GMM Approach

#### Complete Data Approach

With complete data, the full joint distribution for the GMM case factors into the following

$$p(\mathbf{z}, \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}, \boldsymbol{\pi}, \mathbf{X}) = p(\mathbf{X} \mid \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}, \mathbf{z})\, p(\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\} \mid \mathbf{m}_0, \boldsymbol{\kappa}_0, \mathbf{S}_0, \boldsymbol{\nu}_0)\, p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}_0) \tag{6.13}$$

Again we leverage conjugacy (from Section 3.4) to derive closed form expressions for full conditionals from which we aim to sample from. The full conditional for mixing weights $\boldsymbol{\pi}$ remains the same. The rest are given as follows

1. Gaussian parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma} \mid \mathbf{X}, \mathbf{z}) = \prod_k^K \text{NIW}(\mu_k, \Sigma_k | \mathbf{m}_k, \kappa_k, \nu_k, \mathbf{S}_k) \tag{6.14}$$

$$\mathbf{m}_k = \frac{\kappa_{0_k} m_{0_k} + N_k \bar{\mathbf{x}}_k}{\kappa_{0_k} + N_k}$$

$$\kappa_k = \kappa_0 + N_k$$

$$\nu_k = \nu_0 + N_k$$

$$\mathbf{S}_k = \mathbf{S}_{0_k} + \mathbf{S}_{\bar{\mathbf{x}}_k} + \frac{\kappa_{0_k} N_k}{\kappa_{0_k} + N_k} (\bar{\mathbf{x}}_k - m_{0_k})(\bar{\mathbf{x}}_k - m_{0_k})^T$$

2. Latent component assignments for each for each data-point $i$ and component $k$

$$p(z_i = k \mid \mathbf{x}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Normalizing over all component gives responsibilities for the sampling update

$$r_{ik} = \text{Cat}(z_i = k \mid \mathbf{x}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j^K \pi_j \mathcal{N}(\mathbf{x}_j \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

With the full conditionals for all variables in the joint distribution, we perform the following sampling steps at each iteration $t$ of the Gibbs sampler:

$$\text{Sample } \boldsymbol{\pi}^{(t)} \sim \text{Dir}\left(\alpha_{0,1} + N_1^{(t-1)}, \ldots, \alpha_{0,K} + N_K^{(t-1)}\right)$$

$$\text{Sample } \boldsymbol{\Sigma}^{(t)}, \boldsymbol{\mu}^{(t)} \sim \text{NIW}(\mu, \Sigma | \mathbf{m}_N, \kappa_N, \nu_N, \mathbf{S}_N)$$

$$\text{Sample } z_i^{(t)} \sim \text{Categorical}\left(p(z_i = k \mid \mathbf{x}_i, \{\boldsymbol{\Sigma}^{(t)}, \boldsymbol{\mu}^{(t)}\}, \boldsymbol{\pi}^{(t)})\right), \quad \text{for all } i \in [N]$$

where:

- $N_k^{(t-1)} = \sum_{i=1}^N \mathbb{1}(z_i^{(t-1)} = k)$ is the number of data points assigned to component $k$ in the previous iteration.
- $\mathbf{m}_{N_k} = \frac{\kappa_{0_k} m_{0_k} + N_k \bar{\mathbf{x}}_k}{\kappa_{0_k} + N_k}$
- $\kappa_{N_k} = \kappa_0 + N_k$
- $\nu_{N_k} = \nu_0 + N_k$
- $\mathbf{S}_{N_k} = \mathbf{S}_{0_k} + \mathbf{S}_{\bar{\mathbf{x}}_k} + \frac{\kappa_{0_k} N_k}{\kappa_{0_k} + N_k}(\bar{\mathbf{x}}_k - m_{0_k})(\bar{\mathbf{x}}_k - m_{0_k})^T$

**Missing Data Treatment**

In the GMM case, by including the missing entries $\mathbf{X}_h$ in the full joint posterior, the data-likelihood term factorizes into the following observed and missing components : $p(\mathbf{X}_o \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{z}), p(\mathbf{X}_h \mid \mathbf{X}_o, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{z})$. While this permits missing entries to be treated as latent variables, marginalizing over the missing dimensions when sampling the Gaussian model parameters is generally computationally expensive due to the inter-feature dependence introduced by the covariance parameter. This contrasts the Bernoulli case, where likelihoods factor independently across dimensions. As a result, marginalizing the missing values requires computing expectations under the conditional Gaussian distribution for each incomplete data-point. This makes it computationally expensive to compute sufficient statistics, such as component-wise empirical means and covariances. While this in theory allows for convergence in fewer samples, each sample becomes more computationally expensive, potentially offsetting gains in convergence speed [11, 13]. Consequently, the standard approach in GMMs is to adopt full data augmentation, where missing values are sampled at each iteration from their conditional Gaussian distribution given the observed entries, current cluster assignment, and component parameters (as discussed in Section 6.1.2).

$$\text{Sample } \mathbf{X}_h^{(t)} \sim \mathcal{N}(\mathbf{X}_h \mid \mathbf{m}^{h|o}, \mathbf{V}^{h|o})$$

$$\text{Sample } \boldsymbol{\pi}^{(t)} \sim \text{Dir}\left(\alpha_{0,1} + N_1^{(t-1)}, \ldots, \alpha_{0,K} + N_K^{(t-1)}\right)$$

$$\text{Sample } \boldsymbol{\Sigma}^{(t)}, \boldsymbol{\mu}^{(t)} \sim \text{NIW}(\mu, \Sigma | \mathbf{m}_N, \kappa_N, \nu_N, \mathbf{S}_N), \quad \text{Using } \mathbf{X}^{(t)} \text{ for parameter updates}$$

$$\text{Sample } z_i^{(t)} \sim \text{Categorical}\left(p(z_i = k \mid \mathbf{x}_i^{(t)}, \{\boldsymbol{\Sigma}^{(t)}, \boldsymbol{\mu}^{(t)}\}, \boldsymbol{\pi}^{(t)})\right), \quad \text{for all } i \in [N]$$

Partially collapsed approaches involving selectively marginalizing unknowns in certain steps are also possible to speed convergence without compromising the computational efficiency of each iteration [13, 19]. A hybrid approach can be taken where component assignments $\mathbf{z}$ are sampled using only the observed entries, while missing values are imputed solely for updating model parameters. This reduces sensitivity in cluster assignment to uncertainty in imputation, while retaining the benefits of conjugate parameter updates via completed data [13].

Sample $\mathbf{X}_h^{(t)} \sim \mathcal{N}(\mathbf{X}_h \mid \mathbf{m}^{h|o}, \mathbf{V}^{h|o})$

Sample $\boldsymbol{\pi}^{(t)} \sim \text{Dir}\left(\alpha_{0,1} + N_1^{(t-1)}, \ldots, \alpha_{0,K} + N_K^{(t-1)}\right)$

Sample $\boldsymbol{\Sigma}^{(t)}, \boldsymbol{\mu}^{(t)} \sim \text{NIW}(\mu, \Sigma | \mathbf{m}_N, \kappa_N, \nu_N, \mathbf{S}_N)$, Using $\mathbf{X}^{(t)}$ for parameter updates

Sample $z_i^{(t)} \sim \text{Categorical}\left(p(z_i = k \mid \mathbf{x}_{io}, \{\boldsymbol{\Sigma}^{(t)}, \boldsymbol{\mu}^{(t)}\}, \boldsymbol{\pi}^{(t)})\right)$, for all $i \in [N]$

While partially collapsed strategies aim to reduce sampling inefficiencies by marginalizing only a subset of latent variables, they are not without potential drawbacks. In particular, partially collapsing only certain variables (e.g., $\mathbf{z}$ but not $\mathbf{X}_h$) can lead to residual posterior correlations between the retained and marginalized variables [13]. These correlations may manifest as slower mixing or auto-correlated samples, especially if the collapsed and uncollapsed blocks are strongly dependent. In our setting, sampling $z$ from the observed-only marginal while still imputing $\mathbf{X}_h$ for parameter updates can introduce sampling lag which is most noticeable early in the chain. As Dyk and Park [13] emphasize, partial collapse helps only when the marginalized variable meaningfully reduces posterior dependence in the step where it is removed. Otherwise the gains may be small or even negative in practice.

In this research, for both the BMM and GMM case, we use the data-augmentation approach for evaluation, but implement the option to use the partially-collapsed variants.

**GMM Gibbs Sampling Algorithms**

**Algorithm 4** Gibbs Sampling for GMM with Complete Data

**Input:** Data $\mathbf{X}$ with missing values, number of clusters $K$

**Initialize:**

Hyperparameters : $\boldsymbol{\alpha_0}, \mathbf{m}_0, \boldsymbol{\kappa}_0, \mathbf{S}_0, \boldsymbol{\nu}_0$

Variables : $\boldsymbol{\mu}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$, $\boldsymbol{\pi}^{(0)}$, $\mathbf{z}^{(0)}$

**for** $t = 1$ to $T$ **do**

Sample $\boldsymbol{\pi}^{(t)} \sim \text{Dirichlet}(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = \{\alpha_{0,1} + N_1, ..., \alpha_{0,K} + N_K\}$

**for** $i = 1$ to N **do**

Sample $z_i^{(t)} \sim p(z_i \mid \mathbf{x}_i, \boldsymbol{\mu}^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)}, \boldsymbol{\pi}^{(t-1)})$

**end for**

**for** $k = 1$ to $K$ **do**

Sample $\boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)} \sim \text{NIW}(\mu_k, \Sigma_k | \mathbf{m}_k, \kappa_k, \nu_k, \mathbf{S}_k)$

**end for**

**end for**

**Algorithm 5** Gibbs Sampling for GMM with Missing Data

**Input:** Data $\mathbf{X}$ with missing values, number of clusters $K$

**Initialize:**

Hyperparameters : $\boldsymbol{\alpha_0}, \mathbf{m}_0, \boldsymbol{\kappa}_0, \mathbf{S}_0, \boldsymbol{\nu}_0$

Variables : $\boldsymbol{\mu}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$, $\boldsymbol{\pi}^{(0)}$, $\mathbf{z}^{(0)}$

**for** $t = 1$ to $T$ **do**

Sample $\mathbf{X}_h^{(t)} \sim p(\mathbf{X}_h \mid \mathbf{X}_o, \mathbf{z}, \boldsymbol{\mu}^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)})$

Sample $\boldsymbol{\pi}^{(t)} \sim \text{Dirichlet}(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = \{\alpha_{0,1} + N_1, ..., \alpha_{0,K} + N_K\}$

**for** $i = 1$ to N **do**

Sample $z_i^{(t)} \sim p(z_i \mid \mathbf{x}_i^{(t)}, \boldsymbol{\mu}^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)}, \boldsymbol{\pi}^{(t-1)})$

**end for**

**for** $k = 1$ to $K$ **do**

Sample $\boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)} \sim \text{NIW}(\mu_k, \Sigma_k | \mathbf{m}_k, \kappa_k, \nu_k, \mathbf{S}_k)$, where parameters depend on $\mathbf{X}^{(t)}$

**end for**

**end for**

**return** Samples

**Algorithm 6** Partially Collapsed Gibbs Sampling for GMM with Missing Data

**Input:** Data $\mathbf{X}$ with missing values, number of clusters $K$

**Initialize:**

Hyperparameters : $\boldsymbol{\alpha_0}, \mathbf{m}_0, \boldsymbol{\kappa}_0, \mathbf{S}_0, \boldsymbol{\nu}_0$

Variables : $\boldsymbol{\mu}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$, $\boldsymbol{\pi}^{(0)}$, $\mathbf{z}^{(0)}$

**for** $t = 1$ to $T$ **do**

Sample $\mathbf{X}_h^{(t)} \sim p(\mathbf{X}_h \mid \mathbf{X}_o, \mathbf{z}, \boldsymbol{\mu}^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)})$

Sample $\boldsymbol{\pi}^{(t)} \sim \text{Dirichlet}(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = \{\alpha_{0,1} + N_1, ..., \alpha_{0,K} + N_K\}$

**for** $i = 1$ to N **do**

Sample $z_i^{(t)} \sim p(z_i \mid \mathbf{x}_i^{(t)}, \boldsymbol{\mu}^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)}, \boldsymbol{\pi}^{(t-1)})$

**end for**

**for** $k = 1$ to $K$ **do**

Sample $\boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)} \mid \mathbf{X}_o \sim \text{NIW}(\mu_k, \Sigma_k | \mathbf{m}_k, \kappa_k, \nu_k, \mathbf{S}_k)$

**end for**

**end for**

**return** Samples

## 6.3 VBEM Algorithm

As discussed in Section 3.5.2, the Variational Bayes Expectation-Maximization (VBEM) algorithm approximates the intractable true posterior distribution $p$ by minimizing the Kullback–Leibler (KL) divergence to a tractable, factorized distribution $q$. This is achieved by maximizing the Evidence Lower Bound (ELBO), which serves as a lower bound on the marginal likelihood. The VBEM algorithm does so iteratively through two coordinated steps:

- **Variational E-step:** Update the variational distributions over the latent variables (component assignments and missing data) while keeping the variational parameters of the model parameters fixed. This involves computing the expected sufficient statistics under the current variational posterior.
- **Variational M-step:** Update the variational distributions over the model parameters (mixture weights and component parameters) by maximizing the ELBO with respect to those distributions, holding the latent variable distributions fixed.

This alternating procedure continues until convergence. By factorizing the variational posterior and leveraging conjugacy, VBEM allows for efficient closed-form updates to variational posteriors.

This section outlines the algorithmic details, including the specific update steps for each variable in both BMM and GMM settings. For each case, we present the complete data formulation first to establish the core inference mechanism, followed by an extension to incorporate missing data treatment.

### 6.3.1 BMM Approach

#### Complete Data Approach

In the BMM case, our goal is to approximate the full joint posterior over latent variables and parameters given the observed data. The complete-data joint distribution factorizes as:

$$p(\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\pi}, \mathbf{X}) = p(\mathbf{X} \mid \boldsymbol{\theta}, \mathbf{z})\, p(\boldsymbol{\theta})\, p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi}) \tag{6.15}$$

To enable tractable variational inference, we select conjugate priors consistent with the Gibbs sampling formulation:

- Mixing Weights : $p(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi} \mid \boldsymbol{\alpha}_0)$
- Component Means : $p(\boldsymbol{\theta}) = \mathrm{Beta}(\boldsymbol{\theta} \mid \mathbf{a}_0, \mathbf{b}_0)$

In the complete-data setting, we adopt a mean-field approximation, where the variational posterior factorizes as:

$$q(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{z}) = \prod_{i}^{N} q(z_i) \prod_{k}^{K} q(\boldsymbol{\theta}_k) q(\pi_k) \tag{6.16}$$

This factorization assumes independence between the global parameters $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, and the latent component assignments $\mathbf{z}$, across all data points. Given this structure, we now derive the coordinate ascent update steps for each variational factor. In the mean-field setting, this corresponds to computing expectations of the full joint data log likelihood :

$$\begin{aligned} \ln p(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{z}, \mathbf{X}) &= \ln p(\boldsymbol{\pi}) + \ln p(\boldsymbol{\theta}) + \ln p(\mathbf{z} \mid \boldsymbol{\pi}) + \ln p(\mathbf{X} \mid \boldsymbol{\theta}, \mathbf{z}) \\ &= \ln p(\boldsymbol{\pi}) + \sum_{k}^{K} \ln p(\boldsymbol{\theta}_k) + \sum_{i}^{N} \sum_{k}^{K} z_{ik} \ln \pi_k + \sum_{i}^{N} \sum_{k}^{K} z_{ik} \ln p(\mathbf{x}_i \mid \boldsymbol{\theta}_k) \end{aligned} \tag{6.17}$$

where $z_{ik}$ is indicator $\mathbb{1}(z_i = k)$

For each factor variational distribution for each of our variables, we take the following expectations with respect to the joint log-likelihood to derive our update steps

- $q(\boldsymbol{\pi}, \boldsymbol{\theta}) = \exp\left(\mathbb{E}_{q(\mathbf{z})}\left[\ln p(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{z}, \mathbf{X})\right] + \text{const}\right)$, where

$$q(\boldsymbol{\pi}) = \exp\left(\mathbb{E}_{q(\mathbf{z})}\left[\sum_{i}^{N} \sum_{k}^{K} z_{ik} \ln \pi_k\right] + \ln p(\boldsymbol{\pi}) + \text{const}\right)$$

$$q(\boldsymbol{\theta}) = \exp\left(\mathbb{E}_{q(\mathbf{z})}\left[\sum_{i}^{N} \sum_{k}^{K} z_{ik} \ln p(\mathbf{x}_i \mid \boldsymbol{\theta}_k)\right] + \sum_{k}^{K} \ln p(\boldsymbol{\theta}_k) + \text{const}\right)$$

- $q(\mathbf{z}, \mathbf{X}_h) = \exp\left(\mathbb{E}_{q(\boldsymbol{\pi}, \boldsymbol{\theta})}\left[\ln p(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{z}, \mathbf{X})\right] + \text{const}\right)$, where

$$q(\mathbf{z}) = \exp\left(\mathbb{E}_{q(\boldsymbol{\pi}, \boldsymbol{\theta})}\left[\sum_{i}^{N} \sum_{k}^{K} z_{ik} \ln \pi_k + \sum_{i}^{N} \sum_{k}^{K} z_{ik} \ln p(\mathbf{x}_i \mid \boldsymbol{\theta}_k)\right] + \text{const}\right)$$

For mixing weights, $\boldsymbol{\pi}$, we use the conjugacy between the Dirichlet prior and the categorical distribution over latent component assignments to derive the following closed-form variational posterior

$$q(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi} \mid \alpha_{0,1} + N_1, ..., \alpha_{0,K} + N_K) \tag{6.18}$$

Derivation in Appendix 11.3.5

where $N_k = \sum_i^N r_{ik}$ represents the current total variational responsibility of component $k$ for data-point $i$. The updated Dirichlet parameters can be interpreted as the sum of the prior pseudo-counts $\boldsymbol{\alpha}_0$ and the expected number of data points assigned to component $k$.

Using the Beta-Bernoulli conjugacy, the variational posterior over Bernoulli means takes the following closed-form for each component $k$ and feature dimension $d$

$$q(\theta_{kd}) = \text{Beta}(a_{kd}, b_{kd}), \text{where} \tag{6.19}$$

$$a_{kd} = a_{0d} + \sum_i^N r_{ik} x_{id}$$

$$b_{kd} = b_{0d} + \sum_i^N r_{ik}(1 - x_{id})$$

Derivation in Appendix 11.3.6

where $r_{ik} = q(z_i = k)$. These updates add the expected sufficient statistics to the prior counts $(a_{0,d}, b_{0,d})$.

Finally the variational posterior for latent component assignments $\mathbf{z}$ take the form of the following categorical distribution for each data-point $i$ and each component $k$

$$q(z_i = k) = \mathbb{E}\big[\ln \pi_k\big] + \sum_d^D x_{id} \mathbb{E}_{q(\theta)}[\ln \theta_{kd}] + (1 - x_{id}) \mathbb{E}_{q(\theta)}\big[\ln(1 - \theta_{kd})\big] \tag{6.20}$$

where,

- $\mathbb{E}\big[\ln \pi_k\big] = \psi(\alpha_k) - \psi(\sum_j^K \alpha_j)$
- $\mathbb{E}_{q(\theta)}[\ln \theta_{kd}] = \psi(a_{kd}) - \psi(a_{kd} + b_{kd})$
- $\mathbb{E}_{q(\theta)}\big[\ln(1 - \theta_{kd})\big] = \psi(b_{kd}) - \psi(a_{kd} + b_{kd})$

Derivation in Appendix 11.3.7

The ELBO for the BMM complete data case, which these update steps maximize, is given

by the following [27]

$$\text{ELBO} = \mathbb{E}_q\Big[\ln p(\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\pi}, \mathbf{X})\Big] - \mathbb{E}_q\Big[\ln q(\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\pi})\Big] \tag{6.21}$$

$$= \mathbb{E}[\ln p(\boldsymbol{\pi})] + \sum_k^K \mathbb{E}[\ln p(\boldsymbol{\theta}_k)] + \sum_i^N \sum_k^K \mathbb{E}[z_{ik} \ln \pi_k] + \tag{6.22}$$

$$\sum_i^N \sum_k^K \mathbb{E}[z_{ik} \ln p(\mathbf{x}_i \mid \boldsymbol{\theta}_k)] - \mathbb{E}[\ln q(\mathbf{z})] - \sum_k^K \mathbb{E}[\ln q(\boldsymbol{\theta}_k, \pi_k)]$$

**Missing Data Approach**

In the missing data approach, we include missing entries $\mathbf{X}_h$ as a variable to infer in the full joint posterior

$$p(\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\pi}, \mathbf{X}_h, \mathbf{X}_o) = p(\mathbf{X}_o, \mathbf{X}_h \mid \boldsymbol{\theta}, \mathbf{z})\, p(\boldsymbol{\theta})\, p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi}) \tag{6.23}$$

In this incomplete data setting, we adopt the following mean-field variational approximation which includes our missing data variable $\mathbf{X}_h$

$$q(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{z}, \mathbf{X}_h) = \prod_i^N q(z_i, \mathbf{x}_{ih}) \prod_k^K q(\boldsymbol{\theta}_k) q(\pi_k) \tag{6.24}$$

$$= \prod_i^N q(z_i) q(\mathbf{x}_{ih} \mid z_i) \prod_k^K q(\boldsymbol{\theta}_k) q(\pi_k)$$

Here we have made a **structured** mean field assumption that the missing entries $\mathbf{X}_h$ depend conditionally on the latent component assignments $\mathbf{z}$. Maintaining this dependency in the variational posterior allows the imputation of missing values to remain consistent with the inferred cluster structure. This conditional structure reflects the generative process while keeping the variational updates tractable.

The joint log-likelihood for the missing data case for BMMs is given by the following

$$\ln p(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{z}, \mathbf{X}_h, \mathbf{X}_o) = \ln p(\boldsymbol{\pi}) + \ln p(\boldsymbol{\theta}) + \ln p(\mathbf{z} \mid \boldsymbol{\pi}) + \ln p(\mathbf{X}_o, \mathbf{X}_h \mid \boldsymbol{\theta}, \mathbf{z}) \tag{6.25}$$

$$= \ln p(\boldsymbol{\pi}) + \sum_k^K \ln p(\boldsymbol{\theta}_k) + \sum_i^N \sum_k^K z_{ik} \ln \pi_k +$$

$$\sum_i^N \sum_k^K z_{ik} \ln p(\mathbf{x}_{io}, \mathbf{x}_{ih} \mid \boldsymbol{\theta}_k)$$

Where $z_{ik}$ is indicator $\mathbb{1}(z_i = k)$

For each variational distribution, we take the following expectations with respect to the joint log-likelihood to derive the VBEM update steps

- $q(\boldsymbol{\pi}, \boldsymbol{\theta}) = \exp\left(\mathbb{E}_{q(\mathbf{z}, \mathbf{X}_h)}\left[\ln p(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{z}, \mathbf{X}_h, \mathbf{X}_o)\right] + \text{const}\right)$, where

$$q(\boldsymbol{\pi}) = \exp\left(\mathbb{E}_{q(\mathbf{z})}\left[\sum_i^N \sum_k^K z_{ik} \ln \pi_k\right] + \ln p(\boldsymbol{\pi}) + \text{const}\right)$$

$$q(\boldsymbol{\theta}) = \exp\left(\mathbb{E}_{q(\mathbf{z}, \mathbf{X}_h)}\left[\sum_i^N \sum_k^K z_{ik} \ln p(\mathbf{x}_{io}, \mathbf{x}_{ih} \mid \boldsymbol{\theta}_k)\right] + \sum_k^K \ln p(\boldsymbol{\theta}_k) + \text{const}\right)$$

- $q(\mathbf{z}, \mathbf{X}_h) = \exp\left(\mathbb{E}_{q(\boldsymbol{\pi}, \boldsymbol{\theta})}\left[\ln p(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{z}, \mathbf{X}_h, \mathbf{X}_o)\right] + \text{const}\right)$, where

$$q(\mathbf{z}, \mathbf{X}_h) = \exp\left(\mathbb{E}_{q(\theta, \boldsymbol{\pi})}\left[\sum_i^N \sum_k^K z_{ik} \ln \pi_k + \sum_i^N \sum_k^K z_{ik} \ln p(\mathbf{x}_{io}, \mathbf{x}_{ih} \mid \boldsymbol{\theta}_k)\right] + \text{const}\right)$$

$$q(\mathbf{X}_h \mid \mathbf{z}) = \exp\left(\mathbb{E}_{q(\theta)}\left[\sum_i^N \sum_k^K z_{ik} \ln p(\mathbf{x}_{io}, \mathbf{x}_{ih} \mid \boldsymbol{\theta}_k)\right] + \text{const}\right)$$

$$q(\mathbf{z}) = \exp\Big(\ln q(\mathbf{z}, \mathbf{X}_h) - \ln q(\mathbf{X}_h \mid \mathbf{z}) + \text{const}\Big)$$

The update step for the mixing weights remains the same as the complete data case as its variational distribution does not depend on the missing features.

$$q(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi} \mid \alpha_{0,1} + N_1, ..., \alpha_{0,K} + N_K), \quad \text{where } N_k = \sum_i^N r_{ik} \tag{6.26}$$

For the missing data entries $\mathbf{X}_h$, the variational posterior is a Bernoulli distribution with a bias given by $\tau_{idk}$ for each data-point $i$, component $k$, and missing feature dimension $d$

$$q(x_d^{(i)} \mid z_i = k) = \tau_{idk} \tag{6.27}$$

Where

$$\tau_{idk} = \frac{\exp\Big(\mathbb{E}_k[\ln \theta_{kd}]\Big)}{\exp\Big(\mathbb{E}_k[\ln \theta_{kd}]\Big) + \exp\Big(\mathbb{E}_k[\ln(1 - \theta_{kd})]\Big)} \tag{6.28}$$

For the Bernoulli means $\boldsymbol{\theta}$, due to conjugacy, the variational posterior again takes the form of a Beta distribution for each component $k$ and dimension $d$

$$q(\theta_{kd}) = \text{Beta}(a_{kd}, b_{kd}), \text{where} \tag{6.29}$$

$$a_{kd} = a_{0d} + \sum_i^N r_{ik} \cdot \mathbb{E}_k[x_{id}]$$

$$b_{kd} = b_{0d} + \sum_i^N r_{ik} \cdot (1 - \mathbb{E}_k[x_{id}])$$

Derivation in Appendix 11.3.8

This takes the exact same form as the complete data case, with the caveat that we now use expected sufficient statistics given by

$$\mathbb{E}_k[x_{id}] = \begin{cases} \tau_{idk}, & \text{if } d \in h \\ x_{id}, & \text{if } d \in o \end{cases} \tag{6.30}$$

The update for the variational distribution for component assignments $z$ is exactly the same form as the complete case, with the caveat that we only use observed feature dimensions

$$q(z_i = k) = \mathbb{E}\big[\ln \pi_k\big] + \sum_{d \in o}^{D} x_{id}\mathbb{E}_{q(\theta)}[\ln \theta_{kd}] + (1 - x_{id})\mathbb{E}_{q(\theta)}\big[\ln(1 - \theta_{kd})\big] \tag{6.31}$$

where

- $\mathbb{E}\big[\ln \pi_k\big] = \psi(\alpha_k) - \psi\Big(\sum_{k'} \alpha_{k'}\Big)$
- $\mathbb{E}_{q(\theta)}[\ln \theta_{kd}] = \Big(\psi(a_{k,d}) - \psi(a_{k,d} + b_{k,d})\Big)$
- $\mathbb{E}_{q(\theta)}\big[\ln(1 - \theta_{kd})\big] = \Big(\psi(b_{k,d}) - \psi(a_{k,d} + b_{k,d})\Big)$

Derivation in Appendix 11.3.9

The ELBO for the missing data case is given by the following

$$\begin{aligned} \text{ELBO} &= \mathbb{E}_q\Big[\ln p(\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\pi}, \mathbf{X}_o, \mathbf{X}_h)\Big] - \mathbb{E}_q\Big[\ln q(\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\pi}, \mathbf{X}_h)\Big] \\ &= \mathbb{E}_q[\ln p(\boldsymbol{\pi})] + \sum_{k}^{K} \mathbb{E}_q[\ln p(\boldsymbol{\theta}_k)] + \sum_{i}^{N}\sum_{k}^{K} \mathbb{E}_q[z_{ik} \ln \pi_k] + \\ &\quad \sum_{i}^{N}\sum_{k}^{K} \mathbb{E}_q[z_{ik} \ln p(\mathbf{x}_{io}, \mathbf{x}_{ih} \mid \boldsymbol{\theta}_k)] - \mathbb{E}_q[\ln q(\mathbf{z}, \mathbf{X}_h)] - \sum_{k}^{K} \mathbb{E}_q[\ln q(\boldsymbol{\theta}_k, \pi_k)] \end{aligned} \tag{6.32}$$

**BMM VBEM Algorithms**

**Algorithm 7** VBEM Algorithm for BMM with Complete Data

**Input:** Data $\mathbf{X}$, number of clusters $K$

**Initialize:**

Hyperparameters : $\boldsymbol{\alpha_0}, \mathbf{a}_0, \mathbf{b_0}$

Variational Parameters : $\mathbf{a}, \mathbf{b}, \boldsymbol{\alpha}$

**for** $t = 1$ to $T$ **do**

Update $q(\mathbf{z})$ : compute responsibilities $r_{ik}$

Update $q(\boldsymbol{\pi})$ : compute new parameters for each component $k$

$\boldsymbol{\alpha}_k = \alpha_{0k} + \sum_i^N r_{ik}$

Update $q(\boldsymbol{\theta})$ : compute new parameters for component $k$ and dimension $d$

$a_{kd} = a_0 + \sum_i^N r_{ik} x_{id}$

$b_{kd} = b_0 + \sum_i^N r_{ik}(1 - x_{id})$

Compute ELBO and check for convergence

**end for**

**return** Variational Parameters : $\mathbf{a}, \mathbf{b}, \boldsymbol{\alpha}$

**Algorithm 8** VBEM Algorithm for BMM with Missing Data

**Input:** Data $\mathbf{X}$ with missing values, number of clusters $K$

**Initialize:**

Hyperparameters : $\boldsymbol{\alpha_0}, \mathbf{a}_0, \mathbf{b_0}$

Variational Parameters : $\mathbf{a}, \mathbf{b}, \boldsymbol{\alpha}$

**for** $t = 1$ to $T$ **do**

**for** $i = 1$ to $N$, $k = 1$ to $K$, $d = 1$ to $D$ **do**

Compute Expected Sufficient Stats

$$\mathbb{E}_k[x_{id}] = \begin{cases} \tau_{idk}, & \text{if } d \in h \\ x_{id}, & \text{if } d \in o \end{cases}$$

**end for**

Update $q(\mathbf{z})$ : compute responsibilities $r_{ik}$

Update $q(\boldsymbol{\pi})$ : compute new parameters for each component $k$

$\boldsymbol{\alpha}_k = \alpha_{0k} + \sum_i^N r_{ik}$

Update $q(\boldsymbol{\theta})$ : compute new parameters for component $k$ and dimension $d$

$a_{kd} = a_0 + \sum_i^N r_{ik} \mathbb{E}_k[x_{id}]$

$b_{kd} = b_0 + \sum_i^N r_{ik}(1 - \mathbb{E}_k[x_{id}])$

Compute ELBO and check for convergence

**end for**

**return** Variational Parameters : $\mathbf{a}, \mathbf{b}, \boldsymbol{\alpha}$

### 6.3.2 GMM Approach

#### Complete Data Approach

In the GMM case with complete data, the full joint posterior factorizes as the following

$$p(\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}, \mathbf{X}) \propto p(\mathbf{X} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{z})\, p(\boldsymbol{\mu}, \boldsymbol{\Sigma})\, p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi}) \tag{6.33}$$

To enable tractable variational inference, we select conjugate priors consistent with the Gibbs sampling formulation for GMMs

- Mixing Weights : $p(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi} \mid \boldsymbol{\alpha}_0)$
- Component Means : $p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \text{NIW}(\boldsymbol{\mu}, \boldsymbol{\Sigma} \mid \mathbf{m}_0, \boldsymbol{\kappa}_0, \mathbf{S}_0, \boldsymbol{\nu}_0)$

In this complete data setting, we adopt a structured mean-field variational approximation which factorizes as

$$q(\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \prod_i^N q(z_i) \prod_k^K q(\boldsymbol{\mu}_k \mid \boldsymbol{\Sigma}_k) q(\boldsymbol{\Sigma}_k) q(\pi_k) \tag{6.34}$$

The joint log-likelihood for the GMM complete data case is given by the following

$$\begin{aligned} \ln p(\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}, \mathbf{X}) &= \ln p(\boldsymbol{\pi}) + \ln p(\mathbf{z}|\boldsymbol{\pi}) + \ln p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) + \ln p(\mathbf{X} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{z}) \\ &= \ln p(\boldsymbol{\pi}) + \sum_i^N \sum_k^K z_{ik} \ln \pi_k + \sum_k^K \ln p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \sum_i^N \sum_k^K z_{ik} \ln p(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \end{aligned} \tag{6.35}$$

For each factored variational distribution, we take the following expectations with respect to the joint log-likelihood to derive our update steps in order to maximize the ELBO

- $q(\boldsymbol{\pi}, \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}) = \exp\left(\mathbb{E}_{q(\mathbf{z})}\left[\ln p(\boldsymbol{\pi}, \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}, \mathbf{z}, \mathbf{X})\right] + \text{const}\right)$, where

$$q(\boldsymbol{\pi}) = \exp\left(\mathbb{E}_{q(\mathbf{z})}\left[\sum_i^N \sum_k^K z_{ik} \ln \pi_k\right] + \ln p(\boldsymbol{\pi}) + \text{const}\right)$$

$$q(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \exp\left(\mathbb{E}_{q(\mathbf{z})}\left[\sum_i^N \sum_k^K z_{ik} \ln p(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right] + \sum_k^K \ln p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \text{const}\right)$$

- $q(\mathbf{z}) = \exp\left(\mathbb{E}_{q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}\left[\ln p(\boldsymbol{\pi}, \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}, \mathbf{z}, \mathbf{X})\right] + \text{const}\right)$, where

$$q(\mathbf{z}) = \exp\left(\mathbb{E}_{q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}\left[\sum_i^N \sum_k^K z_{ik} \ln \pi_k + \sum_i^N \sum_k^K z_{ik} \ln p(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right] + \text{const}\right)$$

For mixing weights $\boldsymbol{\pi}$, we use the exact same result as in the BMM case

$$q(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi} \mid \alpha_{0,1} + N_1, ..., \alpha_{0,K} + N_K), \quad \text{where } N_k = \sum_i^N r_{ik} \tag{6.36}$$

For the multivariate Gaussian component parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$, due to the conjugacy of the NIW prior and the Gaussian likelihood, the variational posterior takes the form of a NIW distribution where the variational update step results in the following principled updates to the NIW variational posterior

$$q(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \mathcal{N}(\boldsymbol{\mu}_k \mid \mathbf{m}_k, \frac{1}{\kappa_k}\boldsymbol{\Sigma}_k)\mathcal{IW}(\boldsymbol{\Sigma}_k \mid \mathbf{S}_k, \nu_k) \tag{6.37}$$

$$N_k = \sum_i^N r_{ik}$$

$$\bar{\mathbf{x}}_k = \frac{1}{N_k}\sum_i^N r_{ik}\mathbf{x}_i$$

$$\kappa_k = \kappa_{0k} + N_k$$

$$\nu_k = \nu_{0k} + N_k$$

$$\mathbf{m}_k = \frac{1}{\kappa_k}(\kappa_{0k}\mathbf{m}_{0k} + N_k\bar{\mathbf{x}}_k)$$

$$\mathbf{S}_k = \sum_i^N r_{ik}(\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^T$$

Derivation in Appendix 11.3.10

Similar to the BMM case, the variational posterior for the latent component assignments $\mathbf{z}$ takes the same form of categorical distribution for each data-point $i$ and component $k$

$$q(z_i = k) = \mathbb{E}_{q(\pi)}\big[\ln \pi_k\big] + \mathbb{E}_{q(\mu_k,\Sigma_k)}\big[\ln p(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\big] + \text{const} \tag{6.38}$$

Full Derivation in Appendix 11.3.11

The ELBO for the complete data case for GMMs is given by the following [27]

$$\text{ELBO} = \mathbb{E}_q\Big[\ln p(\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}, \mathbf{X})\Big] - \mathbb{E}_q\Big[\ln q(\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})\Big] \tag{6.39}$$

$$= \mathbb{E}_q[\ln p(\boldsymbol{\pi})] + \sum_k^K \mathbb{E}_q[\ln p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] + \sum_i^N\sum_k^K \mathbb{E}_q[z_{ik}\ln \pi_k] +$$

$$\sum_i^N\sum_k^K \mathbb{E}_q[z_{ik}\ln p(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] - \mathbb{E}_q[\ln q(\mathbf{z})] - \sum_k^K \mathbb{E}_q[\ln q(\pi_k)] - \sum_k^K \mathbb{E}_q[\ln q(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

**Missing Data Approach**

In the missing data approach, the full joint posterior for the GMM case factors as follows

$$p(\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}, \mathbf{X}_h, \mathbf{X}_o) = p(\mathbf{X}_o, \mathbf{X}_h \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{z})\, p(\boldsymbol{\mu}, \boldsymbol{\Sigma})\, p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi}) \tag{6.40}$$

To perform inference in the presence of missing data, we adopt a hybrid VBEM approach in which the covariance matrices $\boldsymbol{\Sigma}$ are updated via MAP estimation in an M-step, rather

than placing a variational distribution over them. This choice is motivated by the fact that placing a variational distribution over the covariance matrices renders the variational update for the missing entries $q(\mathbf{X}_h)$ intractable. The MAP estimate of the covariance parameters is denoted by $\hat{\boldsymbol{\Sigma}}$.

Given this, in the missing data setting, we adopt the following mean-field variational approximation

$$\begin{aligned} q(\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{z}, \mathbf{X}_h) &= \prod_{i=1}^{N} q(z_i, \mathbf{x}_{ih}) \prod_{k=1}^{K} q(\boldsymbol{\mu}_k) q(\pi_k) \\ &= \prod_{i=1}^{N} q(z_i) q(\mathbf{x}_{ih} \mid z_i) \prod_{k=1}^{K} q(\boldsymbol{\mu}_k) q(\pi_k) \end{aligned}$$

The joint log-likelihood in the presence of missing entries becomes:

$$\begin{aligned} \ln p(\mathbf{z}, \boldsymbol{\mu}, \hat{\boldsymbol{\Sigma}}, \boldsymbol{\pi}, \mathbf{X}_o, \mathbf{X}_h) &= \ln p(\boldsymbol{\pi}) + \ln p(\mathbf{z}|\boldsymbol{\pi}) + \ln p(\boldsymbol{\mu}, \hat{\boldsymbol{\Sigma}}) + \ln p(\mathbf{X}_o, \mathbf{X}_h \mid \boldsymbol{\mu}, \hat{\boldsymbol{\Sigma}}, \mathbf{z}) \quad (6.41) \\ &= \ln p(\boldsymbol{\pi}) + \sum_{i}^{N} \sum_{k}^{K} z_{ik} \ln \pi_k + \sum_{k}^{K} \ln p(\boldsymbol{\mu}_k, \hat{\boldsymbol{\Sigma}}_k) \\ &\quad + \sum_{i}^{N} \sum_{k}^{K} z_{ik} \ln p(\mathbf{x}_{io}, \mathbf{x}_{ih} \mid \boldsymbol{\mu}_k, \hat{\boldsymbol{\Sigma}}) \end{aligned}$$

We derive the VBEM update steps by taking expectations of the joint log-likelihood with respect to each factor

- $q(\boldsymbol{\pi}, \boldsymbol{\theta}) = \exp\left(\mathbb{E}_{q(\mathbf{z}, \mathbf{X}_h)}\left[\ln p(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{z}, \mathbf{X}_h, \mathbf{X}_o)\right] + \text{const}\right)$, where

$$q(\boldsymbol{\pi}) = \exp\left(\mathbb{E}_{q(\mathbf{z})}\left[\sum_{i}^{N} \sum_{k}^{K} z_{ik} \ln \pi_k\right] + \ln p(\boldsymbol{\pi}) + \text{const}\right)$$

$$q(\boldsymbol{\mu}) = \exp\left(\mathbb{E}_{q(\mathbf{z}, \mathbf{X}_h)}\left[\sum_{i}^{N} \sum_{k}^{K} z_{ik} \ln p(\mathbf{x}_{io}, \mathbf{x}_{ih} \mid \boldsymbol{\mu}_k, \hat{\boldsymbol{\Sigma}}_k)\right] + \sum_{k}^{K} \ln p(\boldsymbol{\mu}_k, \hat{\boldsymbol{\Sigma}}_k) + \text{const}\right)$$

- $q(\mathbf{z}, \mathbf{X}_h) = \exp\left(\mathbb{E}_{q(\boldsymbol{\pi}, \boldsymbol{\mu})}\left[\ln p(\boldsymbol{\pi}, \boldsymbol{\mu}, \hat{\boldsymbol{\Sigma}}, \mathbf{z}, \mathbf{X}_h, \mathbf{X}_o)\right] + \text{const}\right)$, where

$$q(\mathbf{z}, \mathbf{X}_h) = \exp\left(\mathbb{E}_{q(\mu, \boldsymbol{\pi})}\left[\sum_{i}^{N} \sum_{k}^{K} z_{ik} \ln \pi_k + \sum_{i}^{N} \sum_{k}^{K} z_{ik} \ln p(\mathbf{x}_{io}, \mathbf{x}_{ih} \mid \boldsymbol{\mu}_k, \hat{\boldsymbol{\Sigma}}_k)\right] + \text{const}\right)$$

$$q(\mathbf{X}_h \mid \mathbf{z}) = \exp\left(\mathbb{E}_{q(\mu)}\left[\sum_{i}^{N} \sum_{k}^{K} z_{ik} \ln p(\mathbf{x}_{io}, \mathbf{x}_{ih} \mid \boldsymbol{\mu}_k, \hat{\boldsymbol{\Sigma}}_k)\right] + \text{const}\right)$$

$$q(\mathbf{z}) = \exp\left(\ln q(\mathbf{z}, \mathbf{X}_h) - \ln q(\mathbf{X}_h \mid \mathbf{z}) + \text{const}\right)$$

For mixing weights $\boldsymbol{\pi}$, we obtain the exact same result as before in the complete data case

$$q(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi} \mid \alpha_{0,1} + N_1, ..., \alpha_{0,K} + N_K), \quad \text{where } N_k = \sum_i^N r_{ik} \tag{6.42}$$

For the covariance matrices $\boldsymbol{\Sigma}$, we maintain our Inverse-Wishart prior assumption and perform MAP estimation

$$\boldsymbol{\Sigma}_k = \frac{\mathbf{S}_{0k} + \mathbf{S}_k}{N_k + \nu_{0k} + D + 1} \tag{6.43}$$

where

$$\mathbf{S}_k = \sum_i^N r_{ik} \cdot (\mathbb{E}_k[\mathbf{x}_i] - \bar{\mathbf{x}}_k)(\mathbb{E}_k[\mathbf{x}_i] - \bar{\mathbf{x}}_k)^T$$

$$\bar{\mathbf{x}}_k = \frac{1}{N_k} \sum_i^N r_{ik} \cdot \mathbb{E}_k[\mathbf{x}_i]$$

$$N_k = \sum_i^N r_{ik}$$

For the mean parameters $\boldsymbol{\mu}$, the variational posterior takes the form of a Gaussian distribution due to the conjugacy between the NIW prior and the Gaussian data likelihood.

$$q(\boldsymbol{\mu}_k) = \mathcal{N}(\boldsymbol{\mu}_k \mid \mathbf{m}_k, \mathbf{V}_k) \tag{6.44}$$

$$N_k = \sum_i^N r_{ik}$$

$$\mathbf{V}_k = \left(N_k \hat{\boldsymbol{\Sigma}}_k^{-1} + \kappa_0 \hat{\boldsymbol{\Sigma}}_k^{-1}\right)^{-1}$$

$$\mathbf{m}_k = \mathbf{V}_k \left(\kappa_0 \hat{\boldsymbol{\Sigma}}_k^{-1} \mathbf{m}_{0,k} + \hat{\boldsymbol{\Sigma}}_k^{-1} \sum_i^N r_{ik} \mathbb{E}_k[\mathbf{x}_i]\right)$$

Derivation in Appendix 11.3.12

Thus, to compute updates for the Gaussian parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$, we need the following expected sufficient statistics as referenced in Section 6.1.2.

$$\mathbb{E}_k[\mathbf{x}_i] = (\mathbb{E}_k[\mathbf{x}_{ih}]; \mathbf{x}_{io}) = (\mathbf{m}_i^{h|o}; \mathbf{x}_o^{(i)}) \tag{6.45}$$

$$\mathbb{E}_k[\mathbf{x}_i \mathbf{x}_i^T] = \begin{bmatrix} \mathbb{E}_k[\mathbf{x}_{ih} \mathbf{x}_{ih}^T] & \mathbb{E}_k[\mathbf{x}_{ih}] \mathbf{x}_{io}^T \\ \mathbf{x}_{io} \mathbb{E}_k[\mathbf{x}_{ih}]^T & \mathbf{x}_{io} \mathbf{x}_{io}^T \end{bmatrix} \tag{6.46}$$

$$\mathbb{E}_k[\mathbf{x}_{ih} \mathbf{x}_{ih}^T] = \mathbb{E}[\mathbf{x}_{ih}] \mathbb{E}[\mathbf{x}_{ih}]^T + \mathbf{V}_i^{h|o} \tag{6.47}$$

The conditional parameters $\{\mathbf{m_i}^{h|o}, \mathbf{V}_{ik}^{h|o}\}$ are obtained from the variational posterior for the missing data entries $\mathbf{X}_h$ which takes the form of a conditional Gaussian distribution when

conditioned on particular component assignment

$$q(\mathbf{x}_{ih} \mid z_i = k) = \mathcal{N}(\mathbf{x}_{ih} \mid \mathbf{m}_{ik}^{h|o}, \mathbf{V}_{ik}^{h|o}) \tag{6.48}$$

$$\mathbf{V}_{ik}^{h|o} = \hat{\mathbf{\Sigma}}_k^{hh} - \hat{\mathbf{\Sigma}}_k^{ho}(\hat{\mathbf{\Sigma}}_k^{oo})^{-1}\hat{\mathbf{\Sigma}}_k^{oh} \tag{6.49}$$

$$\mathbf{m}_{ik}^{h|o} = \mathbb{E}[\boldsymbol{\mu}_{ik}^h] - \hat{\mathbf{\Sigma}}_k^{ho}(\hat{\mathbf{\Sigma}}_k^{oo})^{-1}(\mathbf{x}_{io} - \mathbb{E}[\boldsymbol{\mu}_{ik}^o]) \tag{6.50}$$

Where $\mathbb{E}[\boldsymbol{\mu}_{ik}^h] = \mathbf{m}_{ik}^h$ and $\mathbb{E}[\boldsymbol{\mu}_{ik}^o] = \mathbf{m}_{ik}^o$ for $\boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\mu}_k \mid \mathbf{m}_k, \mathbf{V}_k)$.

Derivation in Appendix 11.3.13

The update for the variational distribution for the component assignments $\mathbf{z}$ takes the same form as the complete-data case except we use only observed entries for the data-likelihood component

$$q(z_i = k) = \mathbb{E}_{q(\pi_k)}\big[\ln \pi_k\big] + \mathbb{E}_{q(\mu_k, \Sigma_k)}\big[\ln p(\mathbf{x}_{io} \mid \boldsymbol{\mu}_k, \mathbf{\Sigma}_k)\big] + \text{const} \tag{6.51}$$

Full derivation in Appendix 11.3.13

The ELBO for the missing data case for GMMs is given by the following

$$\text{ELBO} = \mathbb{E}_q\Big[\ln p(\mathbf{z}, \boldsymbol{\mu}, \hat{\mathbf{\Sigma}}, \boldsymbol{\pi}, \mathbf{X}_o, \mathbf{X}_h)\Big] - \mathbb{E}_q\Big[\ln q(\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\pi}, \mathbf{X}_h)\Big] \tag{6.52}$$

$$= \mathbb{E}_q[\ln p(\boldsymbol{\pi})] + \sum_k^K \mathbb{E}_q[\ln p(\boldsymbol{\mu}_k, \hat{\mathbf{\Sigma}}_k)] + \sum_i^N \sum_k^K \mathbb{E}_q[z_{ik} \ln \pi_k] + \tag{6.53}$$

$$\sum_i^N \sum_k^K \mathbb{E}_q[z_{ik} \ln p(\mathbf{x}_{io}, \mathbf{x}_{ih} \mid \boldsymbol{\mu}_k, \hat{\mathbf{\Sigma}}_k)] - \mathbb{E}_q[\ln q(\mathbf{z}, \mathbf{X}_h)] \tag{6.54}$$

$$- \sum_k^K \mathbb{E}_q[\ln q(\boldsymbol{\mu}_k)] - \sum_k^K \mathbb{E}_q[\ln q(\boldsymbol{\pi})] \tag{6.55}$$

**GMM VBEM Algorithms**

**Algorithm 9** VBEM Algorithm for GMM with Complete Data

**Input:** Data $\mathbf{X}$, number of clusters $K$

**Initialize:**

Hyperparameters : $\boldsymbol{\alpha_0}, \mathbf{m}_0, \boldsymbol{\kappa}_0, \mathbf{S}_0, \boldsymbol{\nu}_0$

Variational Parameters : $\boldsymbol{\alpha}, \mathbf{m}, \boldsymbol{\kappa}, \mathbf{S}, \boldsymbol{\nu}, \hat{\boldsymbol{\Sigma}}$

**for** $t = 1$ to $T$ **do**

Update $q(\mathbf{z})$ : compute responsibilities $r_{ik}$

Update $q(\boldsymbol{\pi})$ : compute new parameters for each component $k$

$\boldsymbol{\alpha}_k = \alpha_{0k} + \sum_i^N r_{ik}$

Update $q(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ : compute new parameters

$\kappa_k = \kappa_{0k} + N_k$

$\nu_k = \nu_{0k} + N_k$

$\mathbf{m}_k = \frac{1}{\kappa_k}(\kappa_{0k}\mathbf{m}_{0k} + N_k\bar{\mathbf{x}}_k)$

$\mathbf{S}_k = \sum_i^N r_{ik}(\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^T$

Compute ELBO and check for convergence

**end for**

**return** Variational Parameters $\boldsymbol{\alpha}, \mathbf{m}, \boldsymbol{\kappa}, \mathbf{S}, \boldsymbol{\nu}, \hat{\boldsymbol{\Sigma}}$

**Algorithm 10** VBEM Algorithm for GMM with Missing Data

**Input:** Data $\mathbf{X}$ with missing values, number of clusters $K$

**Initialize:**

Hyperparameters : $\boldsymbol{\alpha_0}, \mathbf{m}_0, \boldsymbol{\kappa}_0, \mathbf{S}_0, \boldsymbol{\nu}_0$

Variational Parameters : $\boldsymbol{\alpha}, \mathbf{m}, \boldsymbol{\kappa}, \mathbf{S}, \boldsymbol{\nu}, \hat{\boldsymbol{\Sigma}}$

**for** $t = 1$ to $T$ **do**

**for** $i = 1$ to $N$, $k = 1$ to $K$, $d = 1$ to $D$ **do**

Compute conditional parameters

$\mathbf{m}_{ik}^{h|o} = \mathbb{E}[\boldsymbol{\mu}_{ik}^{h}] - \hat{\boldsymbol{\Sigma}}_k^{ho}(\hat{\boldsymbol{\Sigma}}_k^{oo})^{-1}(\mathbf{x}_{io} - \mathbb{E}[\boldsymbol{\mu}_{ik}^{o}])$

$\mathbf{V}_{ik}^{h|o} = \hat{\boldsymbol{\Sigma}}_k^{hh} - \hat{\boldsymbol{\Sigma}}_k^{ho}(\hat{\boldsymbol{\Sigma}}_k^{oo})^{-1}\hat{\boldsymbol{\Sigma}}_k^{oh}$

Compute Expected Sufficient Stats

$\mathbb{E}_k[\mathbf{x}_i] = (\mathbb{E}_k[\mathbf{x}_{ih}]; \mathbf{x}_{io}) = (\mathbf{m}_i^{h|o}; \mathbf{x}_o^{(i)})$

$\mathbb{E}_k[\mathbf{x}_i\mathbf{x}_i^T] = \begin{bmatrix} \mathbb{E}_k[\mathbf{x}_{ih}\mathbf{x}_{ih}^T] & \mathbf{E}_k[\mathbf{x}_{ih}]\mathbf{x}_{io}^T \\ \mathbf{x}_{io}\mathbf{E}_k[\mathbf{x}_{ih}]^T & \mathbf{x}_{io}\mathbf{x}_{io}^T \end{bmatrix}$

**end for**

Update $q(\mathbf{z})$ : compute responsibilities $r_{ik}$

Update $q(\boldsymbol{\pi})$ : compute new parameters for each component $k$

$\boldsymbol{\alpha}_k = \alpha_{0k} + \sum_i^N r_{ik}$

MAP update for covariances : $\hat{\boldsymbol{\Sigma}}_k = \frac{\mathbf{S}_{0k} + \mathbf{S}_k}{N_k + \nu_{0k} + D + 1}$

Update $q(\boldsymbol{\mu})$ : compute new parameters

$\mathbf{V}_k = \left(N_k\hat{\boldsymbol{\Sigma}}_k^{-1} + \kappa_0\hat{\boldsymbol{\Sigma}}_k^{-1}\right)^{-1}$

$\mathbf{m}_k = \mathbf{V}_k\left(\kappa_0\hat{\boldsymbol{\Sigma}}_k^{-1}\mathbf{m}_{0,k} + \hat{\boldsymbol{\Sigma}}_k^{-1}\sum_i^N r_{ik}\mathbb{E}_k[\mathbf{x}_i]\right)$

Compute ELBO and check for convergence

**end for**

**return** Variational Parameters $\boldsymbol{\alpha}, \mathbf{m}, \boldsymbol{\kappa}, \mathbf{S}, \boldsymbol{\nu}, \boldsymbol{\Sigma}$

## 6.4 EM Algorithm

As discussed in Section 3.5.3, the EM algorithm takes on a frequentist approach that parallels the VBEM algorithm, providing an iterative procedure for computing maximum likelihood (or MAP) estimates in the presence of latent variables. We implement this algorithm with missing data treatment to serve as a non–fully Bayesian benchmark against which our fully Bayesian approaches (Gibbs sampling and VBEM) can be compared.

Unlike VBEM, which maintains variational distributions over model parameters and latent variables, the EM algorithm performs point estimation, optimizing parameters directly via alternating **E-steps** and **M-steps**. The algorithm maximizes the expected complete data log-likelihood, which serves as a lower bound on the marginal likelihood, and is guaranteed

to increase (or leave unchanged) at each iteration. As with VBEM, convergence is typically assessed via changes in the lower bound.

The application of the EM algorithm for complete data in mixture modeling is well established and is not novel. In this section, we specify only the missing-data approaches for the EM algorithm for both BMMs and GMMs, however, derivations for the complete case can be found in [28].

### 6.4.1 Missing Data Approach

There are to types of missing data approaches that this research adopts to benchmark against the fully Bayesian approaches : complete case EM with ad-hoc imputation of missing entries and EM algorithm incorporating missing data as latent variables.

For the ad-hoc imputation strategy, three common techniques are considered: mode, mean, and median imputation. For each feature (column), the corresponding summary statistic is computed using the observed entries and used to impute all missing values in that column. In cases where an entire column is missing, a fallback value of `0` is used to impute all entries in that column. Once imputation is complete, the standard complete-case EM algorithm is applied to the resulting dataset.

In the latent variable EM approach, we incorporate the missing data $\mathbf{X}_h$ as a latent variable in the CDLL.

$$\ln p(\mathbf{X}_h, \mathbf{X}_o, \mathbf{z} \mid \boldsymbol{\phi}, \boldsymbol{\pi}) = \ln p(\mathbf{z} \mid \boldsymbol{\pi}) + \ln p(\mathbf{X}_h, \mathbf{X}_o \mid \mathbf{z}, \boldsymbol{\phi}) \tag{6.56}$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} \ln \pi_k + \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} \ln p(\mathbf{x}_{io}, \mathbf{x}_{ih} \mid \boldsymbol{\phi}_k) \tag{6.57}$$

Thus, the expected CDLL is given by the following

$$Q(\boldsymbol{\phi} \mid \boldsymbol{\phi}^{(t-1)}) = \sum_{n}^{N} \sum_{k}^{K} r_{ik}^{(t-1)} \ln \pi_k + \sum_{n}^{N} \sum_{k}^{K} \mathbb{E}_{p(\mathbf{z}, \mathbf{X}_h \mid \mathbf{X}_o, \boldsymbol{\phi}^{(t-1)})} \Big[ z_{ik} \ln p(\mathbf{x}_{io}, \mathbf{x}_{ih} \mid \boldsymbol{\phi}_k) \Big] \tag{6.58}$$

With this new latent variable, the posterior required for the E-Step must be modified to include the missing entries in order to compute the new expected CDLL. The new posterior for an individual data-point $i$ is

$$p(\mathbf{x}_{ih}, z_i \mid \mathbf{x}_{io}, \boldsymbol{\phi}^{(t-1)}) = \underbrace{p(z_i \mid \mathbf{x}_{io}, \boldsymbol{\phi}^{(t-1)})}_{r_{ik}} \underbrace{p(\mathbf{x}_{ih} \mid \mathbf{x}_{io}, z_i, \boldsymbol{\phi}^{(t-1)})}_{\text{Missing Conditional}} \tag{6.59}$$

Responsibilities are computed using only the observed parts of the data, constituting a marginal likelihood approach

$$r_{ik} = p(z_i = k \mid \boldsymbol{\phi}^{(t-1)}, \mathbf{x}_{io}) \propto p(z_i = k \mid \boldsymbol{\phi}_k^{(t-1)}) p(\mathbf{x}_{io} \mid \boldsymbol{\phi}_k^{(t-1)}, z_i = k) \tag{6.60}$$

The missing conditional distribution gives us the expected sufficient statistics needed to compute the M-Step.

**BMM Case**

In the BMM case with missing data, the expected CDLL used in the EM algorithm takes the form

$$Q(\boldsymbol{\phi} \mid \boldsymbol{\phi}^{(t-1)}) = \sum_{n}^{N} \sum_{k}^{K} r_{ik}^{(t-1)} \left[ \ln \pi_k + \left( \mathbb{E}_k[\mathbf{x}_i]^T \ln \boldsymbol{\theta}_k + (1 - \mathbb{E}_k[\mathbf{x}_i])^T \ln(1 - \boldsymbol{\theta}_k) \right) \right] \tag{6.61}$$

Where $\mathbb{E}_k[\cdot]$ denotes expectation under the conditional distribution $p(\mathbf{x}_{ih} \mid z_i = k, \mathbf{x}_{io}, \boldsymbol{\theta}_k^{(t-1)})$. Thus, in contrast to the complete data case, we need to compute the expected sufficient statistics which for the BMM case is given by the following due to the feature independence property of BMMs

$$\mathbb{E}_k[\mathbf{x}] = \begin{bmatrix} \mathbb{E}_k[\mathbf{x}_h] \\ \mathbf{x}_o \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}_k^{(t-1)} \\ \mathbf{x}_o \end{bmatrix} \tag{6.62}$$

These will be necessary for the M-Step of the EM algorithm with missing data

**E-Step:** The responsibilities for the BMM missing data case remains largely the same with the caveat that we marginalize out the missing dimensions

$$r_{ik} = \frac{\pi_k \prod_{d \in o}^{D} \theta_{kd}^{x_{id}} (1 - \theta_{kd})^{(1 - x_{id})}}{\sum_{j}^{K} \pi_j \prod_{d \in o}^{D} \theta_{jd}^{x_{id}} (1 - \theta_{jd})^{(1 - x_{id})})} \tag{6.63}$$

**M-Step:** The M-step updates the model parameters using the expected sufficient statistics derived from the current posterior estimates.

1. Mixing Weights
$$\pi_k = \frac{\sum_i^N r_{nk}}{N} \tag{6.64}$$

2. Component Parameters
$$\theta_{kd} = \frac{\sum_i^N r_{ik} \mathbb{E}_k[\mathbf{x}_i]}{N_k} \tag{6.65}$$

   where $N_k = \sum_i^N r_{nk}$

**GMM Case**

In the GMM case with missing data, the expected CDLL used in the EM algorithm takes the form [6]

$$Q(\boldsymbol{\phi} \mid \boldsymbol{\phi}^{(t-1)}) = \sum_{n}^{N} \sum_{k}^{K} r_{ik}^{(t-1)} \Bigg[ \ln \pi_k - \frac{1}{2} \ln |\boldsymbol{\Sigma}_k| - \frac{D}{2} \ln 2\pi \tag{6.66}$$

$$- \frac{1}{2} \text{tr} \Big( \boldsymbol{\Sigma}_k^{-1} (\mathbb{E}_k[\mathbf{x}_i \mathbf{x}_i^T] + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T - 2 \boldsymbol{\mu}_k \mathbb{E}_k[\mathbf{x}_i]^T) \Big) \Bigg] \tag{6.67}$$

Where $\mathbb{E}_k[\cdot]$ denotes expectation under the conditional distribution $p(\mathbf{x}_{ih} \mid \mathbf{x}_{io}, z_i = k, \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}^{(t-1)})$. As mentioned in Section 6.1.2, due to the feature dependence property of the multivariate Gaussian, the expected sufficient statistics for the GMM case are the following

$$\mathbb{E}_k[\mathbf{x}_i] = (\mathbb{E}_k[\mathbf{x}_{ih}]; \mathbf{x}_{io}) = (\mathbf{m}_i^{h|o}; \mathbf{x}_o^{(i)}) \tag{6.68}$$

$$\mathbb{E}_k[\mathbf{x}_i \mathbf{x}_i^T] = \begin{bmatrix} \mathbb{E}_k[\mathbf{x}_{ih}\mathbf{x}_{ih}^T] & \mathbb{E}_k[\mathbf{x}_{ih}]\mathbf{x}_{io}^T \\ \mathbf{x}_{io}\mathbb{E}_k[\mathbf{x}_{ih}]^T & \mathbf{x}_{io}\mathbf{x}_{io}^T \end{bmatrix} \tag{6.69}$$

$$\mathbb{E}_k[\mathbf{x}_{ih}\mathbf{x}_{ih}^T] = \mathbb{E}[\mathbf{x}_{ih}]\mathbb{E}[\mathbf{x}_{ih}]^T + \mathbf{V}_i^{h|o} \tag{6.70}$$

where

$$p(\mathbf{x}_{ih} \mid \mathbf{x}_{io}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \mathcal{N}(\mathbf{x}_{ih} \mid \mathbf{m}_{ik}^{h|o}, \mathbf{V}_{ik}^{h|o}) \tag{6.71}$$

**E-Step:** Like in the BMM case, to compute responsibilities, we marginalize the missing dimensions [28]

$$r_{ik} = \frac{\pi_k \mathcal{N}(\mathbf{x}_{io} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j^K \pi_j \mathcal{N}(\mathbf{x}_{io} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{6.72}$$

**M-Step:** The M-step updates the model parameters using the expected sufficient statistics derivation from the current posterior estimates [28]

1. Mixing Weights remain the same as with the BMM case

$$\pi_k = \frac{\sum_i^N r_{nk}}{N} \tag{6.73}$$

2. Component Mean Parameters

$$\boldsymbol{\mu}_k = \frac{\sum_i^N r_{ik}\mathbb{E}_k[\mathbf{x}_i]}{N_k} \tag{6.74}$$

   where $N_k = \sum_i^N r_{nk}$

3. Component Covariance Parameters

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k}\sum_i^N r_{ik} \cdot (\mathbb{E}_k[\mathbf{x}_{ih}\mathbf{x}_{ih}^T] - \boldsymbol{\mu}_k\boldsymbol{\mu}_k^T) \tag{6.75}$$

   where $N_k = \sum_i^N r_{nk}$

**Algorithm 11** EM Algorithm for BMM with Missing Data

**Input:** Data $\mathbf{X}$, number of clusters $K$

**Initialize:**

Variables : $\boldsymbol{\theta}, \boldsymbol{\pi}$

**for** $t = 1$ to $T$ **do**

**E-Step:** Compute Responsibilities and Expected Sufficient Statistics

**for** $i = 1$ to $N$, $k = 1$ to $K$ **do**

$$r_{ik} = \frac{\pi_k \prod_d^D \theta_{kd}^{x_{id}} (1-\theta_{kd})^{(1-x_{id})}}{\sum_j^K \pi_j \prod_d^D \theta_{jd}^{x_{id}} (1-\theta_{jd})^{(1-x_{id})}}$$

$$\mathbb{E}_k[x_{id}] = \begin{cases} \theta_{kd}, & \text{if } d \in h \\ x_{id}, & \text{if } d \in o \end{cases}$$

**end for**

**M-Step:** Update Parameters

**for** $k = 1$ to $K$ **do**

$$\pi_k = \frac{\sum_i^N r_{nk}}{N}$$

$$\theta_k = \frac{\sum_i^N r_{ik} \mathbb{E}_k[\mathbf{x}_i]}{N_k}$$

**end for**

Compute log-likelihood and check for convergence

**end for**

**return** Parameters $\boldsymbol{\theta}, \boldsymbol{\pi}, \mathbf{R} = \{r_{ik} \mid i \in N, k \in K\}$

**GMM EM Algorithms**

**Algorithm 12** EM Algorithm for GMM with Missing Data

**Input:** Data $\mathbf{X}$, number of clusters $K$

**Initialize:**

Variables : $\boldsymbol{\mu}, \boldsymbol{\pi}$

**for** $t = 1$ to $T$ **do**

**E-Step:** Compute Responsibilities And Expected Sufficient Statistics

**for** $i = 1$ to $N$, $k = 1$ to $K$ **do**

$r_{ik} = \frac{\pi_k \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j^K \pi_j \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$

**end for**

**for** $k = 1$ to $K$ **do**

$\mathbb{E}_k[\mathbf{x}_i] = (\mathbb{E}_k[\mathbf{x}_{ih}]; \mathbf{x}_{io}) = (\mathbf{m}_i^{h|o}; \mathbf{x}_o^{(i)})$

$\mathbb{E}_k[\mathbf{x}_i \mathbf{x}_i^T] = \begin{bmatrix} \mathbb{E}_k[\mathbf{x}_{ih}\mathbf{x}_{ih}^T] & \mathbf{E}_k[\mathbf{x}_{ih}]\mathbf{x}_{io}^T \\ \mathbf{x}_{io}\mathbf{E}_k[\mathbf{x}_{ih}]^T & \mathbf{x}_{io}\mathbf{x}_{io}^T \end{bmatrix}$

$\mathbb{E}_k[\mathbf{x}_{ih}\mathbf{x}_{ih}^T] = \mathbf{E}[\mathbf{x}_{ih}]\mathbf{E}[\mathbf{x}_{ih}]^T + \mathbf{V}_i^{h|o}$

**end for**

**M-Step:** Update Parameters

**for** $k = 1$ to $K$ **do**

$\pi_k = \frac{\sum_i^N r_{nk}}{N}$

$\boldsymbol{\mu}_k = \frac{\sum_i^N r_{ik}\mathbb{E}_k[\mathbf{x}_i]}{N_k}$

$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_i^N r_{ik} \cdot (\mathbb{E}_k[\mathbf{x}_{ih}\mathbf{x}_{ih}^T] - \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T)$

**end for**

Compute log-likelihood and check for convergence

**end for**

**return** Parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}, \mathbf{R} = \{r_{ik} \mid i \in N, k \in K\}$

# Chapter 7

# Evaluation

## 7.1 Datasets

A total of six datasets were used to evaluate the performance of the inference algorithms developed in this research, with three datasets allocated to each model type: GMMs and BMMs. For both model classes, two datasets were synthetically generated while one was drawn from real-world data.

The use of synthetic datasets provided several key advantages. First, they enabled precise control over the data-generating process, allowing the evaluation of inference quality against known ground truth parameters and cluster assignments. More importantly, synthetic data can be constructed to follow the same generative assumptions as the models being evaluated. This alignment ensures that model mis-specification is not a confounding factor, thereby allowing for a more focused assessment of the inference algorithms themselves in their ability to recover latent structure and handle missing data effectively.

In contrast, the inclusion of a real-world dataset for each model type provided a complementary benchmark for assessing how well the methods generalize beyond idealized conditions. Real-world data typically introduces noise, dependencies, or structural deviations from the assumed model, making it a more challenging but realistic test of algorithm performance. Together, this dual approach of testing on both synthetic and real-world datasets enables a comprehensive evaluation of the inference procedures across both controlled and practical scenarios. The datasets for each model also cover a range of complexity, allowing assessment of algorithm performance under different levels of data dimensionality and different numbers of components.

### 7.1.1 BMM Datasets

Two synthetic datasets were generated to test the performance of inference for the BMM models : one following the generative process, and another higher dimensional dataset representing a number of basic shapes using binary pixel values.

### Synthetic Dataset Generation

Synthetic datasets for the Bernoulli Mixture Model (BMM) were generated in accordance with the model's generative process. Specifically, the parameters of each component were independently drawn from a shared Beta prior, ensuring consistency with the assumptions used during inference. For each component $k$ and each feature $d$, the corresponding Bernoulli success probability was sampled as:

$$\theta_{kd} \sim \text{Beta}(a, b) \tag{7.1}$$

where we set $a = b = \frac{1}{2}$ as to create more separable component means.

Mixing weights $\boldsymbol{\pi}$ are drawn from a Dirichlet distribution

$$\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\alpha}) \tag{7.2}$$

where we use a concentration parameter of $\boldsymbol{\alpha} = 1$ which matches the specification of the algorithms.

Finally, a sample of size $N = 2500$ is generated with $D = 5$ features and $K = 5$ components

$$\mathbf{X}_k \sim \text{Bern}(\boldsymbol{\theta}_k) \text{ for } k = 1 \text{ to } K \tag{7.3}$$

### Shape Dataset Generation

The second synthetic dataset used for evaluating BMM inference consists of binary images representing three basic geometric shapes: triangle, square, and plus. Each shape is defined over a fixed $10 \times 10$ image grid, as illustrated in Figure 7.1.

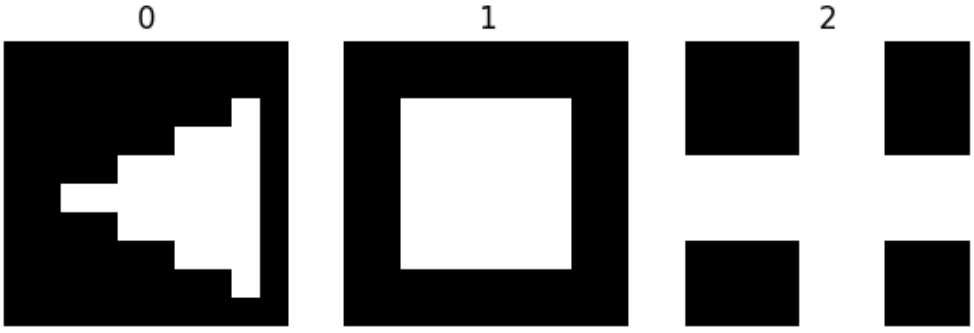



Figure 7.1: Base templates for the triangle, square, and plus shapes used in the synthetic BMM dataset.

To introduce variability within each cluster while maintaining its defining structure, we treat each pixel as a Bernoulli random variable. Pixels belonging to the shape are assigned a high Bernoulli parameter $\theta$, while background pixels are assigned a low value $1 - \theta$. Sampling from this pixel-wise Bernoulli distribution produces noisy variants of each base shape. This process mimics a realistic scenario in which data generated from a latent cluster exhibits within-class variation.

Figure 7.2 shows an example of a generated data-point with noise introduced.

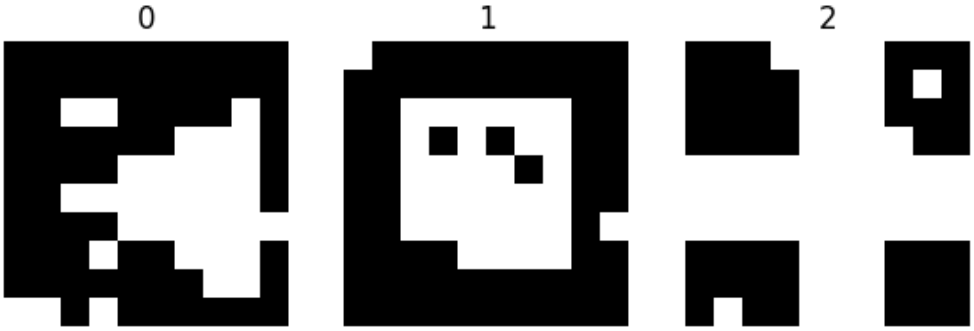



Figure 7.2: Example sample from the shape dataset with pixel-wise Bernoulli noise.

For this dataset, we use a noise parameter of $\theta = 0.95$, resulting in high visual fidelity to the base template while preserving stochasticity. A total of 1500 samples are generated across $K = 3$ components with 500 samples per component.

The empirical component means (i.e., the average of all samples per component) are shown in Figure 7.3. These illustrate the algorithm's ability to recover the dominant structure of each shape through averaging, despite the injected pixel-level noise.

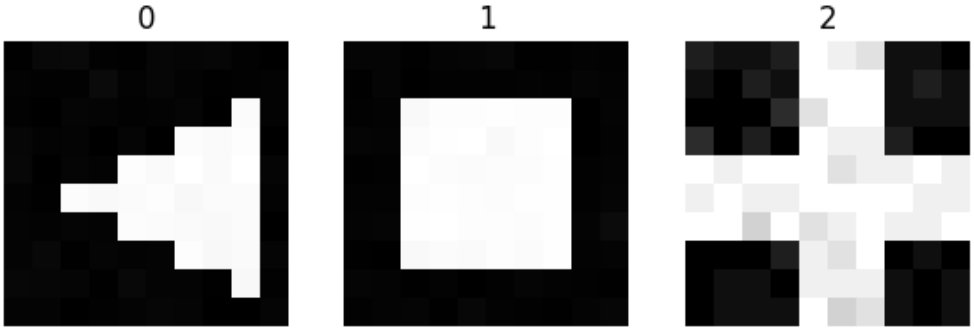



Figure 7.3: Empirical component means computed over samples assigned to each component.

This dataset is designed to assess inference performance in a higher-dimensional binary setting ($D = 100$), where the underlying component distributions are irregular and structured. While the data-generating process still conforms to the assumptions of a BMM in a formal sense, the spatial patterns encoded by the shapes introduce complexity not captured by simple i.i.d. Bernoulli noise, making the inference task more realistic and challenging.

**MP Voting Dataset [31]**

This dataset from the **Public Whip Project**, consists of voting results of British MPs (Members of Parliament) across legislative divisions for the 2024 year. Here, the datapoints correspond to individual MPs, where features are voting results for legislative divisions proposed for the 2024 year. The dataset naturally contains missingness as MPs can and often choose not to vote for divisions. MPs in this dataset are associated with a number of political parties, where for the 2024 year, we have the following breakdown

Table 7.1: MP dataset — party representation

| **Party** | **MPs** |
|---|---|
| Labour | 411 |
| Conservative | 121 |
| Liberal Democrats | 72 |
| Independent | 14 |
| Scottish National Party | 9 |
| Democratic Unionist Party | 5 |
| Reform UK | 5 |
| Plaid Cymru | 4 |
| Green | 4 |

Thus the clustering objective becomes clustering the MP's into their political parties based on their voting behavior. The 2024 dataset has a total missingness of 30.92 percent. A benefit of the MP dataset is the fact that it has very imbalanced component sizes which allows testing the ability of inference algorithms to accurately recover skewed mixing proportions under missing data.

### 7.1.2 GMM Datasets

Two synthetic datasets were generated to test the performance of inference for the GMM models : one following the generative process, and another higher dimensional dataset representing a number of numerical digits using normalized pixel values.

#### Synthetic Dataset Generation

The synthetic dataset for the GMM was generated in accordance with the model's generatie process. The parameters fro each component were independently drawn from a shared Normal-Inverse-Wishart prior, ensuring consistency with the assumptions during inference. For each component $k$, the corresponding Gaussian parameters are sampled as

$$\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \sim \mathcal{NIW}(\nu, \kappa, \mathbf{S}, \mathbf{m}) \tag{7.4}$$

Where

- $\nu = D + 2$ where $D$ is the number of dimensions per data-point
- $\kappa = 0.01$
- $\mathbf{m}$ = zero vector of size $D$
- $\mathbf{S}$ is the Identity matrix of size $D \times D$

These choices math the prior specification used in the Fully Bayesian inference algorithms.

Mixing weights $\boldsymbol{\pi}$ are drawn from a Dirichlet distribution

$$\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\alpha}) \tag{7.5}$$

where we use a concentration parameter of $\boldsymbol{\alpha} = 1$ which matches the specification of the algorithms.

Finally a sample of size $N = 750$ is generated with $D = 3$ features, and $K = 3$ components.

$$\mathbf{X}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad \text{For } k = 1 \text{ to } K \tag{7.6}$$

### Digits Dataset Generation

The second synthetic dataset used for evaluating GMM inference consists of images representing numerical digits from 0 to 4. Each image is defined over a fixed $5 \times 5$ grids as illustrated below

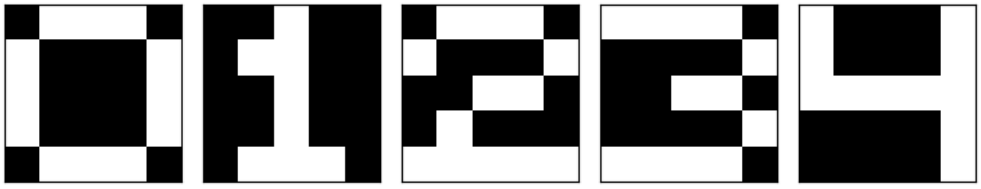


Figure 7.4: Base templates for the digits 0-4 in the synthetic digit GMM dataset

To introduce variability within each cluster while maintaining its defining structure, we introduce Gaussian noise to each pixel between [0,1] with a mean of 0.0 and a standard deviation of 0.1.

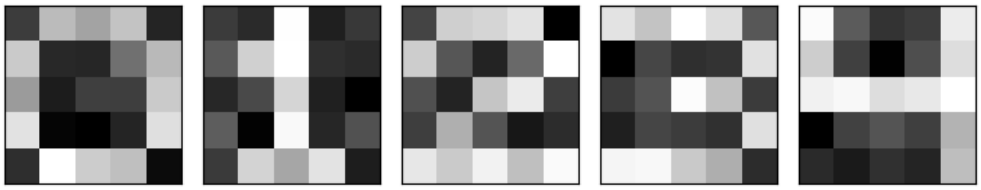


Figure 7.5: Example sample from the digits dataset with pixel-wise Gaussian noise.

For each component, 500 data points are generated resulting in even mixing proportions across components.

Similarly to the shape data set for the BMM case, this data set is designed to evaluate the performance of inference in a higher-dimensional setting ($D = 25$) where the underlying component distributions are irregularly structured.

### Iris Dataset [14]

The Iris dataset is popular multivariate dataset consisting of four continuous measurements for three Irish flower species. The measurements which constitute the features of the dataset are : sepal length, sepal width, petal width, and petal length. There are in total 50 data-points for each species, resulting in a balanced set of 150 total data-points.

Unlike synthetically generated data, which adheres strictly to the generative assumptions of the model, the Iris dataset introduces natural variation and potential model mis-specification. This allows for a more realistic assessment of the algorithms' robustness. It also enables a comparison between model performance on controlled vs. naturally occurring data, helping to evaluate the generalization capability of the inference methods.

### 7.1.3 Summary

| **Name** | BMM Synthetic | BMM Shapes | BMM MP Votes | GMM Synthetic | GMM Digits | GMM Iris |
|---|---|---|---|---|---|---|
| Features | 5 | 100 | 207 | 3 | 25 | 4 |
| Components | 5 | 3 | 9 | 3 | 5 | 3 |
| Data points | 2500 | 1500 | 645 | 750 | 1500 | 150 |

Table 7.2: Summary of Datasets

## 7.2 Experiments

The overall experimental design of this research is to investigate how the performance of each missing data approach changes as the amount of missingness increases. In particular, we simulate missingness in 10 percent increments, ranging from 0 percent (fully observed) to 90 percent missingness. This allows for a systematic evaluation of inference robustness across a wide spectrum of missing data severity. This is done for each dataset with the exception of the MP voting record dataset which naturally incorporates missingness.

Missingness is applied uniformly across the dataset according to the MCAR assumption. Specifically, for a given missingness level $P$, each feature of each data-point is independently set to missing with probability $P$. This ensures that the missingness mechanism is independent of both the observed and unobserved data values, thereby isolating the effect of missingness quantity from the complexities of MAR or MNAR scenarios.

By varying the degree of missingness in controlled increments, the experimental setup enables a direct comparison of different inference methods in terms of their ability to recover latent structure and impute missing values under increasingly challenging conditions. This design also facilitates the identification of performance degradation thresholds, where certain algorithms begin to fail or lose accuracy, thus providing insight into the robustness of each approach.

A secondary objective of the evaluation is to provide some assessment for the generalizability of the each inference method. To this end, each dataset is partitioned into training and test sets with a 80-20 proportion using stratified sampling conditioned on the component labels. The inference algorithms are trained on the training set, and performance metrics are computed on both the training and held-out test data using the learned parameters. This setup enables an evaluation of how well each approach generalizes beyond the data it was trained on, offering insight of its practical utility in downstream tasks involving incomplete data.

The two proposed fully Bayesian inference approaches are the **VBEM** and **Gibbs** Sampler algorithms. We bench mark these two against the following non-fully Bayesian approaches

- **One-Step EM** : The approach outlined in Section 6.4.1 wherein the missing data $\mathbf{X}_h$ is incorporated into the complete data log-likelihood as a latent variable.

- **Imputation EM** : Complete case EM algorithm where missing values are first imputed using the mean of the observed values for each feature. In the case of BMMs, imputed values are rounded to 0 or 1. If a feature is missing for all data-points, we use a fall-back value of 0 for imputation.
- **Complete Case EM** : EM algorithm with complete case analysis wherein we only use fully observed data-points to train the algorithm. In this approach, data-points with missing features are assigned uniform responsibility over the components, constituting a random latent cluster assignment.
- **Imputation K-Means** : We use the K-Means algorithm with random initialization and Euclidean distancing to cluster data-points where missing values are imputed using the mean of the observed values for each feature. For BMMs, imputed values are rounded to 0 or 1. If a feature is missing for all data-points, we use a fall-back value of 0 for imputation.
- **Complete Case K-Means** : K-Means algorithm wherein we only use fully observed data-points to train the algorithm. Data-points with missing features are held out of the training process, and given uniformly random cluster assignments.

### 7.2.1 Evaluation Metrics

To comprehensively assess the performance of each inference algorithm, we employ evaluation metrics that capture both the quality of the clustering results and the overall model fit. This dual approach ensures that the evaluation reflects not only the utility of the clustering results but also the internal consistency of the learned model with respect to the data-generating process.

#### Adjusted Random Index

Adjusted Random Index (ARI) provides a measure of clustering quality by computing the similarity between two clusterings. To do so, it considers all pairs of samples and counts pairs that are assigned in the same or different clusters in the predicted and true clusterings [1]. ARI is a chance-adjusted Rand Index such that a completely random cluster assignment produces an ARI of 0. ARI has a maximum value of 1, which indicates perfect alignment, and a minimum value of -1, which indicates that the clustering is worse than what would be expected by random chance [1]. ARI serves as the primary metric to assess quality of each algorithm in inferring latent structures in the data.

#### Log-Likelihood

To evaluate how well the model inferred by each algorithm fits the underlying data distribution, we compute the average log-likelihood of the learned parameters on the ground truth, fully observed version of the dataset used for training. This evaluation isolates the quality of the inferred model parameters by applying them to data unaffected by missingness, allowing for a fair comparison across different levels of missing data.

For the Gibbs Sampling and standard EM algorithms, the log-likelihood can be computed directly using the the PDF of either the GMM or BMM since these two algorithms directly

learn the model parameters. For Gibbs sampling, we use the mean of parameters across samples after aligning samples using the Hungarian method.

- **BMM:** Given component parameters $\boldsymbol{\theta}$ and mixing weights $\boldsymbol{\pi}$, the log-likelihood is

$$\mathcal{L} = \frac{1}{N}\sum_{i}^{N} \log\left[\sum_{k}^{K} \pi_k \prod_{d}^{D} \theta_{kd}^{x_{id}}(1-\theta_{kd})^{(1-x_{id})})\right] \tag{7.7}$$

- **GMM:** Given component parameters $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ and mixing weights $\boldsymbol{\pi}$, the log-likelihood is

$$\mathcal{L} = \frac{1}{N}\sum_{i}^{N} \log\left[\sum_{k}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\theta}_k, \boldsymbol{\Sigma}_k)\right] \tag{7.8}$$

The VBEM algorithm on the other hand learns the parameters of the variational posteriors of the model parameters. Consequently, we evaluate the expected log-likelihood using these parameters.

- **BMM:** Given Beta posterior parameters $\{\mathbf{a}, \mathbf{b}\}$, Dirichlet posterior parameter $\boldsymbol{\alpha}$, the expected log-likelihood is

$$\begin{aligned}\mathcal{L} =& \frac{1}{N}\sum_{i}^{N}\sum_{k}^{K}\left[\mathbb{E}_{q(\boldsymbol{\pi})}\Big[\ln \pi_k\Big] + \mathbb{E}_{q(\boldsymbol{\theta})}\Big[\sum_{d}^{D} x_{id}\ln\theta_{kd} + (1-x_{id})\ln(1-\theta_{kd})\Big]\right] \\ =& \frac{1}{N}\sum_{i}^{N}\sum_{k}^{K} r_{ik}\Bigg[\psi(\alpha_k) - \psi\Big(\sum_{k'}\alpha_{k'}\Big) + \\ & \sum_{d}^{D}\Big[x_{id}\big(\psi(a_{k,d}) - \psi(a_{k,d}+b_{k,d})\big) + (1-x_{id})\big(\psi(b_{k,d}) - \psi(a_{k,d}+b_{k,d})\big)\Big]\Bigg]\end{aligned} \tag{7.9}$$

- **GMM:** Given the NIW posterior parameters $\{\mathbf{m}, \mathbf{S}, \boldsymbol{\kappa}, \boldsymbol{\nu}\}$ and the Dirichlet posterior parameter $\boldsymbol{\alpha}$, the expected log-likelihood is

$$\begin{aligned}\mathcal{L} =& \frac{1}{N}\sum_{i}^{N}\sum_{k}^{K}\left[\mathbb{E}_{q(\boldsymbol{\pi})}\Big[\ln \pi_k\Big] + \mathbb{E}_{q(\boldsymbol{\mu},\boldsymbol{\Sigma})}\Big[\ln \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\Big]\right] \\ =& \frac{1}{N}\sum_{i}^{N}\sum_{k}^{K} r_{ik}\Bigg[\psi(\alpha_k) - \psi\Big(\sum_{k'}\alpha_{k'}\Big) + \\ & -\frac{1}{2}\Big[D\ln 2\pi - \ln|\mathbf{W}_k| - \sum_{i}^{D}\psi\left(\frac{\nu_k+1-i}{2}\right) \\ & + \nu_k(\mathbf{x}_i - \mathbf{m}_k)^T\mathbf{W}_k^{-1}(\mathbf{x}_i - \mathbf{m}_k) + \frac{D}{\kappa_k}\Big]\Bigg]\end{aligned} \tag{7.10}$$

We naturally omitted K-Means from this metric as it is not probablistic.

#### Root Mean Square Error (RMSE) on Missing Data Imputation

In the case of missing data, we additional compute the accuracy of imputing missing entries using the posterior predictive distribution over the learned parameters. This is evaluated using the RMSE between the imputed entries and the ground truth complete data. This is given by the following

$$\text{RMSE} = \sqrt{\frac{1}{|h|}\sum_{i}^{N}\sum_{d\in h}(x_{id} - \hat{x}_{id})^2} \tag{7.11}$$

where $|h|$ denotes the total count of missing entries, $x_{id}$ denotes the value at dimension $d$ for observation $i$ of the ground truth complete data, and $\hat{x}_{id}$ denotes the corresponding entry for the imputed data. We have chosen omit evaluating K-Means under this metric since the K-Means algorithm does not have a principled mechanism for missing data imputation.

### 7.2.2 Algorithm Setup

In this section, we discuss the configuration of each algorithm including aspects such as initialization strategy and early stopping criteria.

#### Gibbs Sampler

For the Gibbs sampling approach, the algorithm is run with 6000 iterations with a burn-in of 2000 samples. The procedure starts by randomly initializing latent assignments $\mathbf{z}$, and then proceeds with the remainder of the iterative sampling procedure. Clustering quality is evaluated on the MAP estimate of latent assignments, where the remaining performance metrics are evaluated using the mean of the parameters of the resulting empirical distribution after aligning samples using the Hungarian algorithm.

#### VBEM and EM Algorithm

For the VBEM and EM approaches, the algorithms are run with a convergence tolerance of 0.001 and a maximum of 200 iterations in the case that convergence is not met. The algorithms are initialized by first randomly initializing latent assignments $\mathbf{z}$, then followed by the iterative variational updates. Both algorithms are run with 10 random restarts to account for sub-optimal initializations and the risk of convergence to local optima. Summarizing performance metrics are taken across the 10 runs where we report the mean and standard deviation of the metrics.

#### K-Means

For all K-Means approaches, we utilize random centroid initialization and therefore are run with 10 random restarts to account for sub-optimal initializations. Summarizing performance metrics are taken across the 10 runs where we report the mean and standard deviation of the metrics.

# Chapter 8

# Results

## 8.1 Clustering Performance (ARI)

### 8.1.1 BMM Datasets

#### BMM Synthetic Dataset

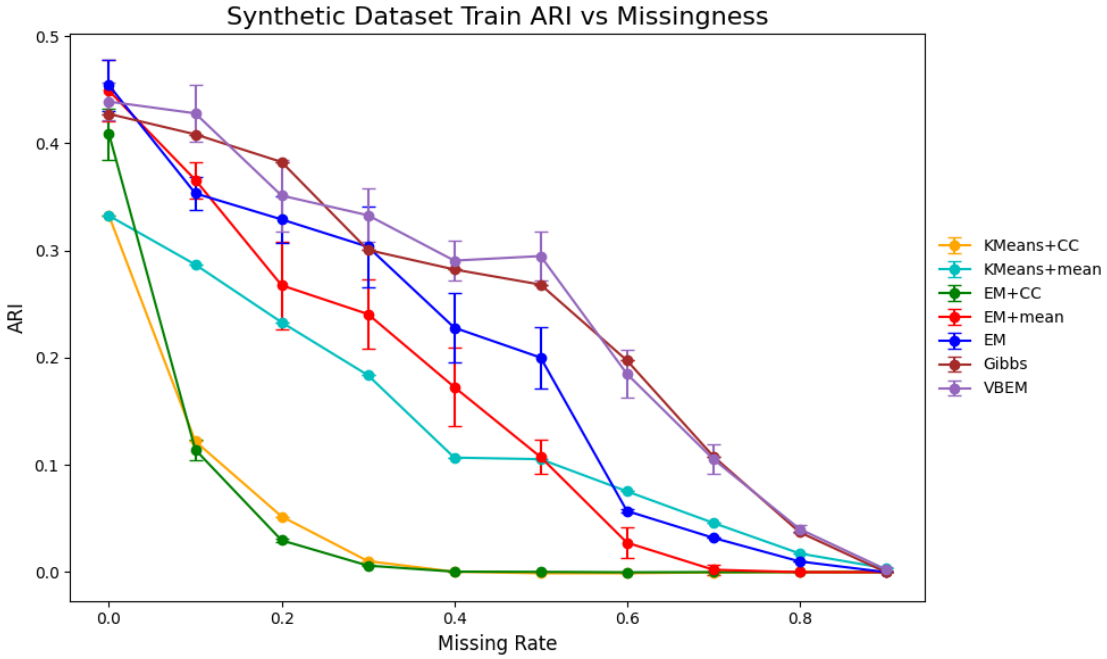



Figure 8.1: Clustering Performance (ARI) Using BMM on the synthetic dataset training set. Note that the `EM+CC` curve completely overlaps the `KMeans+CC` curve, indicating identical performance across all levels of missingness.

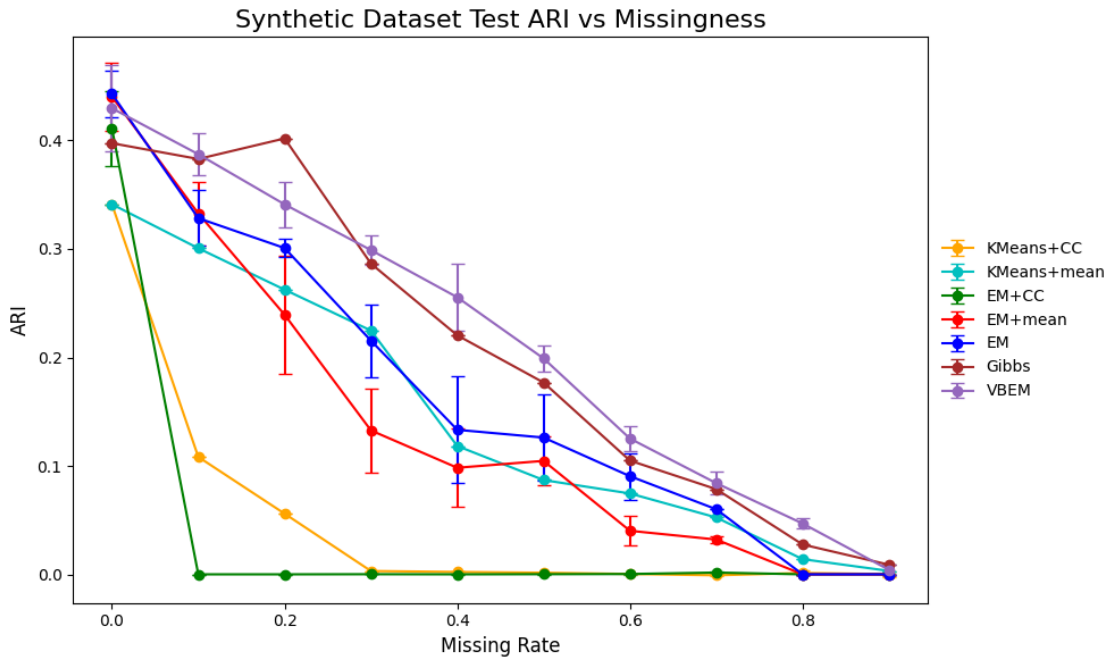



Figure 8.2: Clustering performance (ARI) of BMM on the held-out test set of the synthetic dataset. Note that the `EM+CC` curve completely overlaps the `KMeans+CC` curve.

### Shapes Dataset

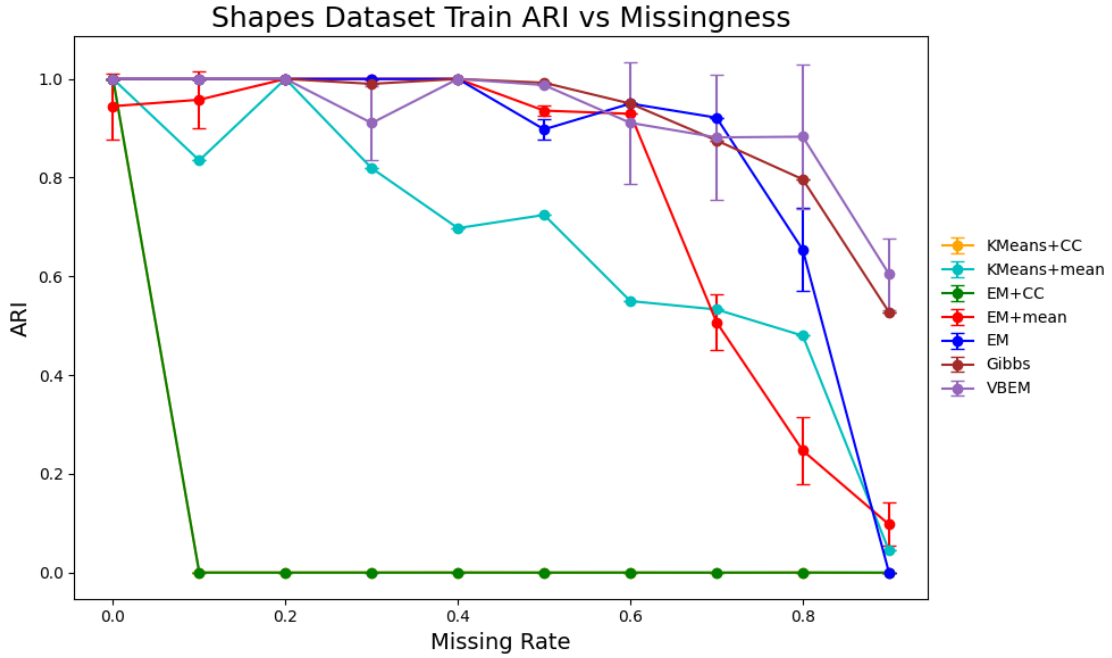



Figure 8.3: Clustering Performance (ARI) Using BMM on the shapes dataset training set. Note that the `EM+CC` curve completely overlaps the `KMeans+CC` curve.

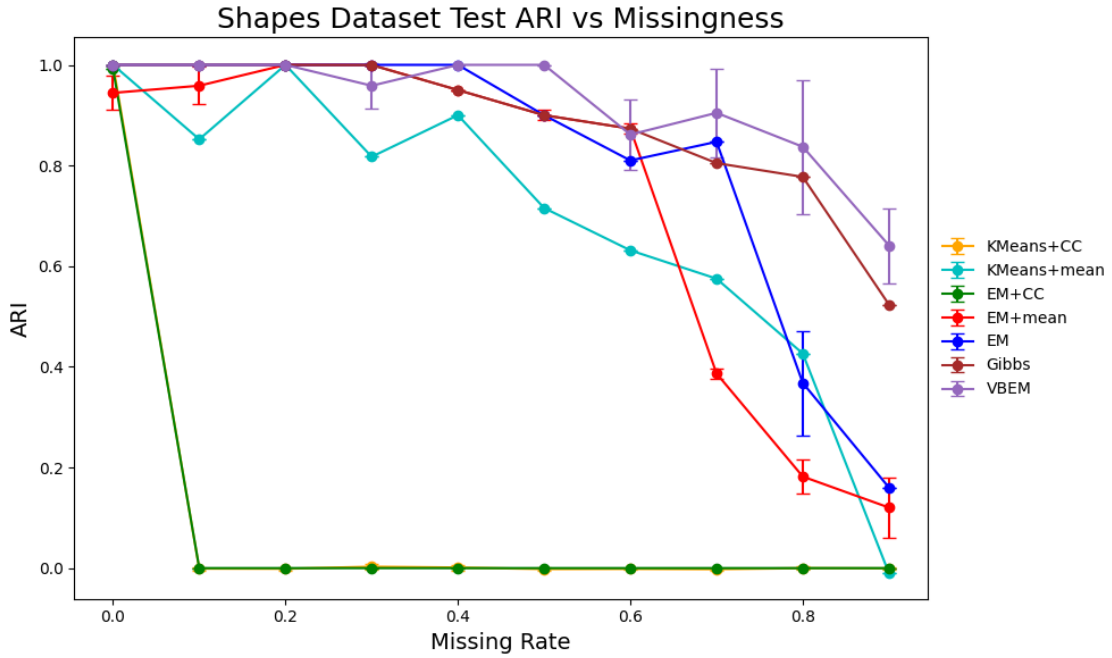



Figure 8.4: Clustering performance (ARI) of BMM on the held-out test set of the shapes dataset. Note that the `EM+CC` curve completely overlaps the `KMeans+CC` curve.

### MP Voting Dataset

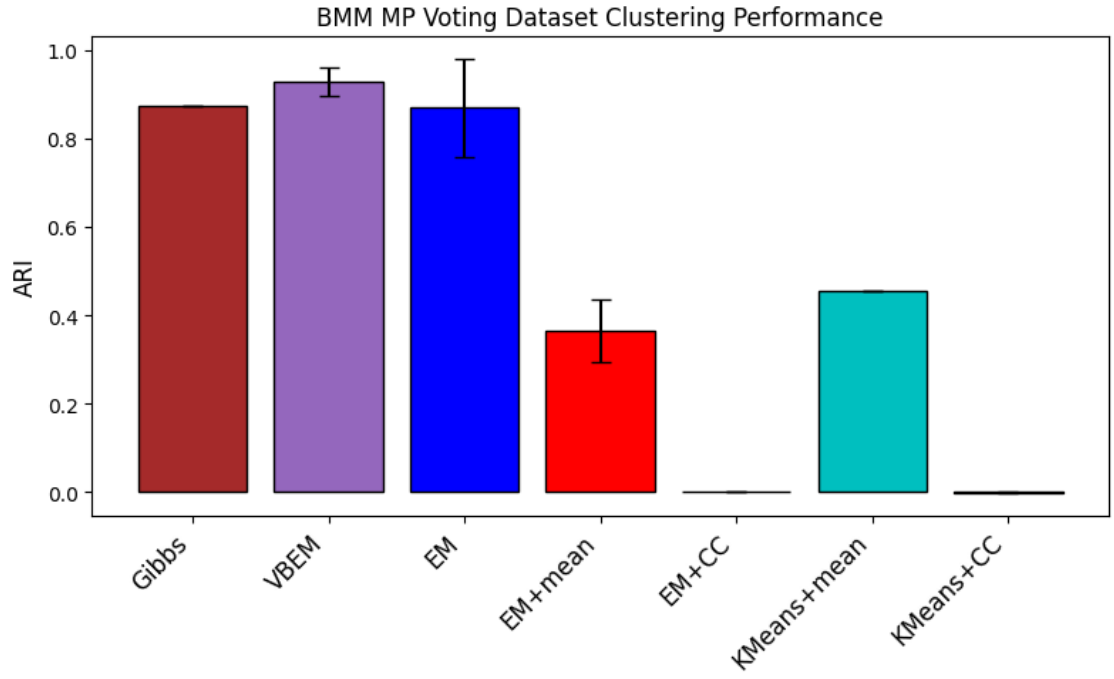



Figure 8.5: Clustering Performance (ARI) using BMM on the MP Voting Dataset.

### 8.1.2 GMM Datasets

#### GMM Synthetic Dataset

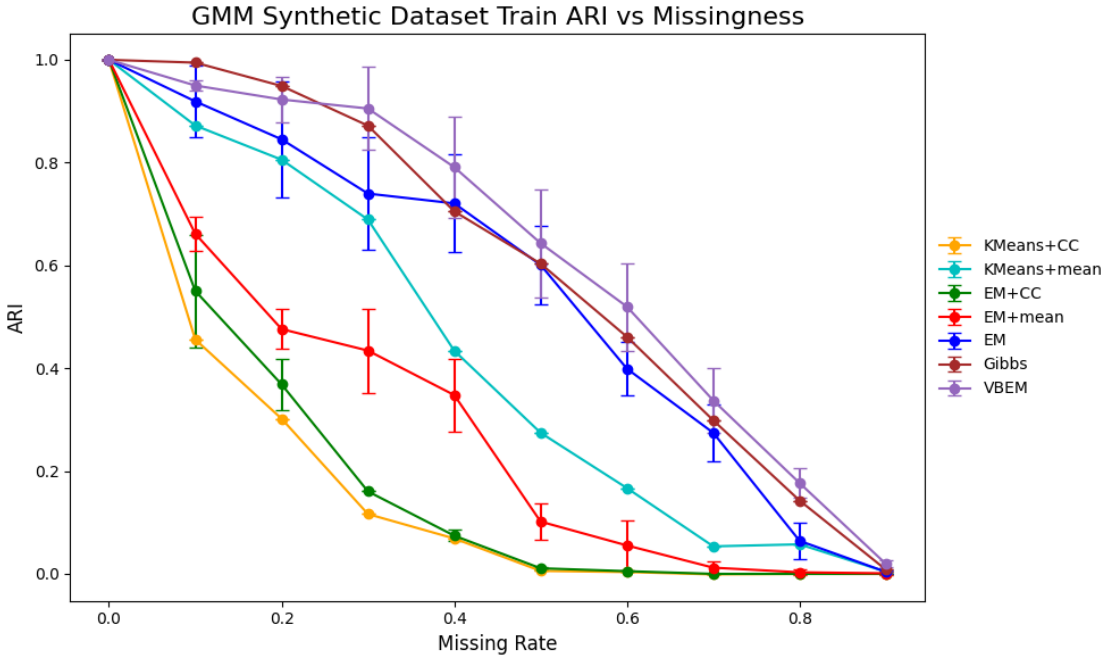



Figure 8.6: Clustering Performance (ARI) of GMM on the synthetic dataset training set.

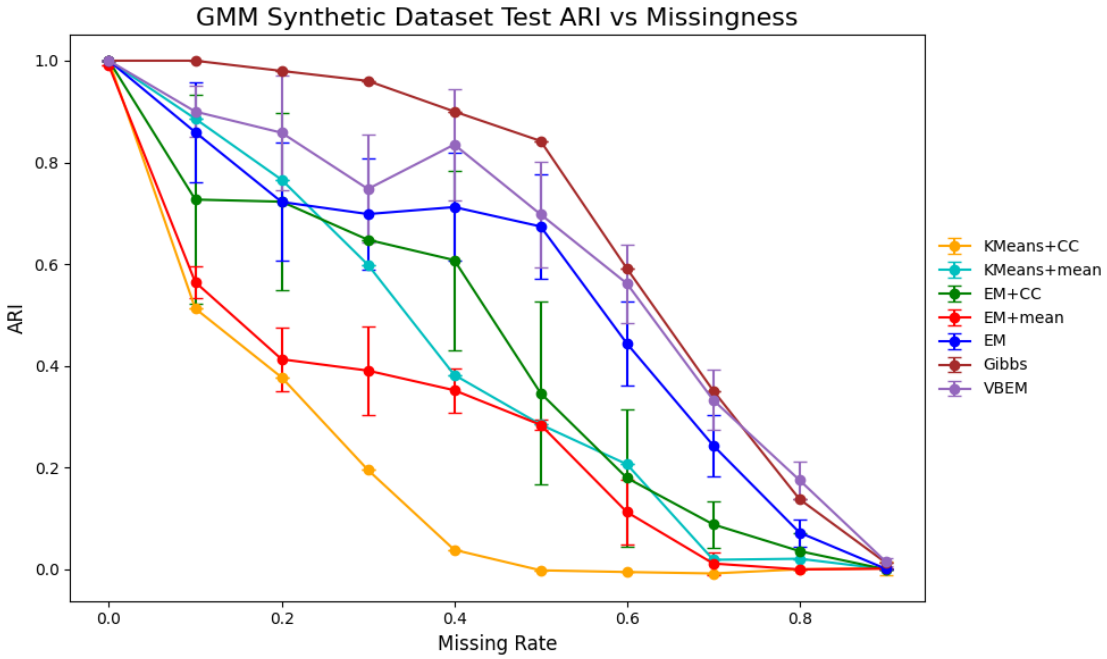



Figure 8.7: Clustering performance (ARI) of GMM on the held-out test set of the synthetic dataset.

### GMM Digits Dataset

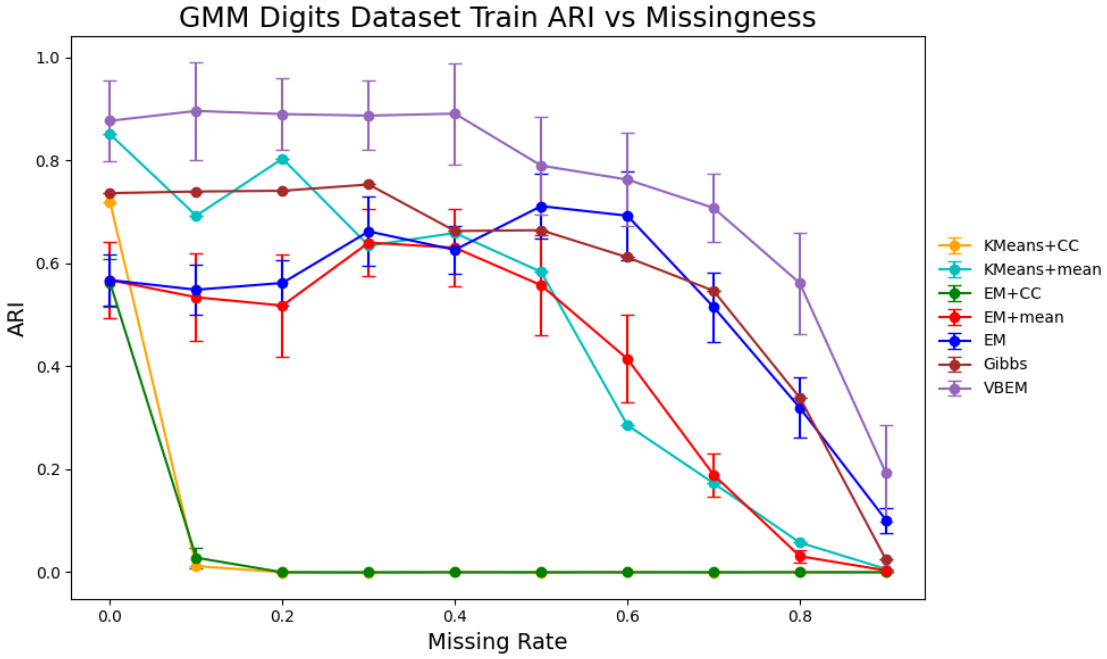



Figure 8.8: Clustering Performance (ARI) of GMM on the digits dataset training set.

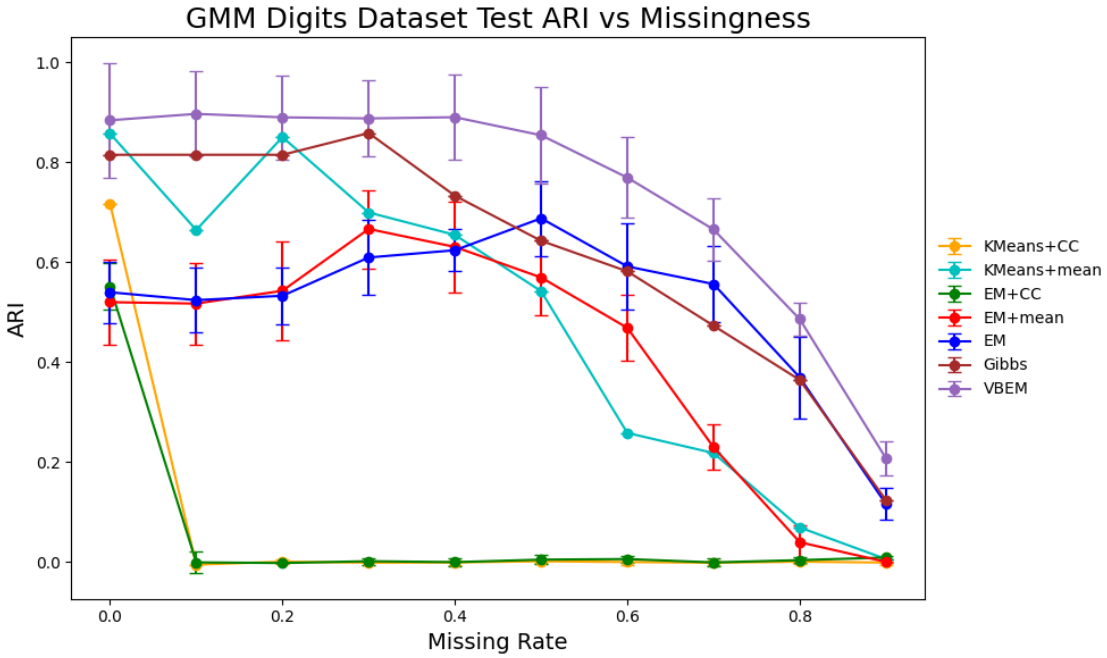



Figure 8.9: Clustering performance (ARI) of GMM on the held-out test set of the digits dataset.

### GMM Iris Dataset

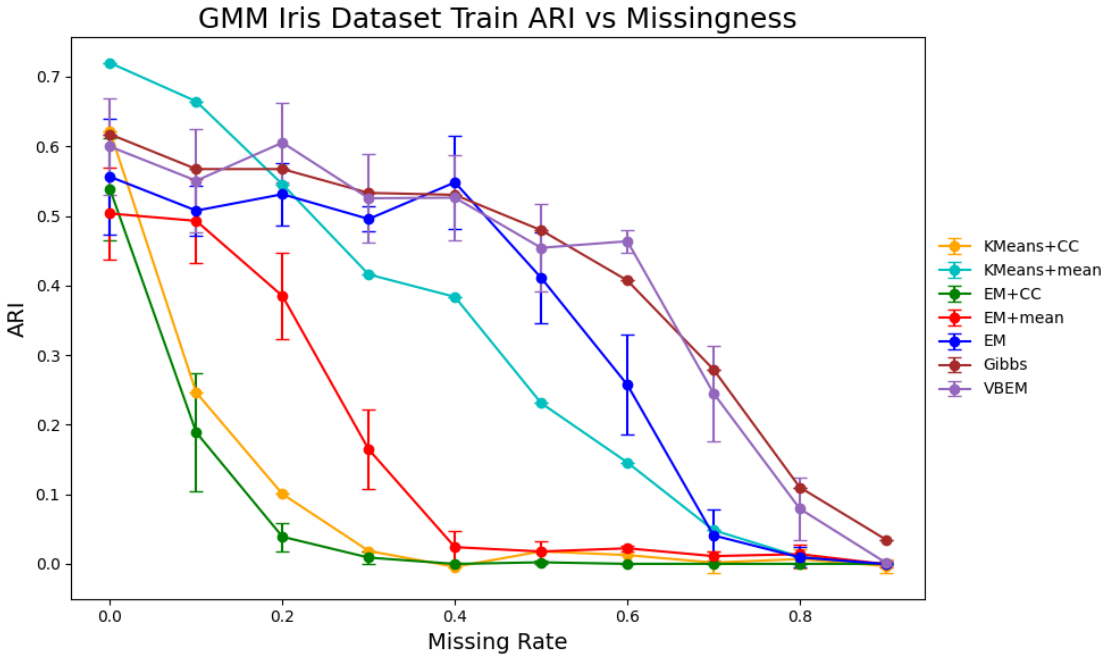



Figure 8.10: Clustering Performance (ARI) of GMM on the Iris dataset training set.

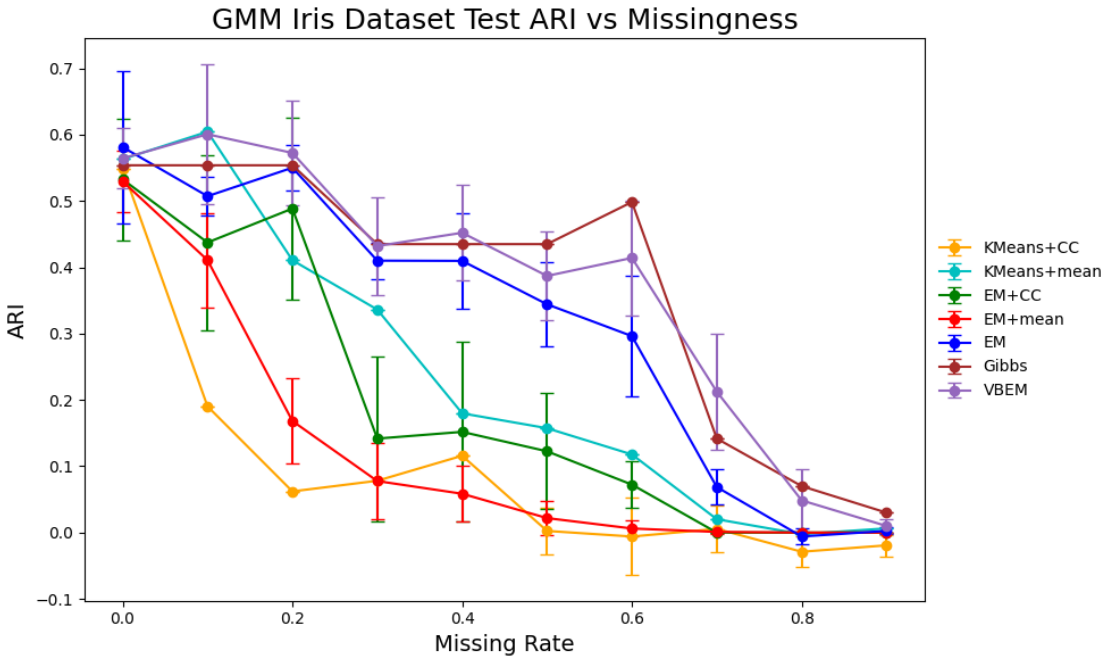



Figure 8.11: Clustering performance (ARI) of GMM on the held-out test set of the Iris dataset.

## Summary

### 8.1.3 Clustering Performance

From the results, fully Bayesian approaches consistently achieve the highest ARI scores across all datasets, with substantial gains over both ad-hoc imputation and complete-case

methods. They are closely followed by the one-step EM algorithm, underscoring the value of statistically principled approaches to handling missing data. In many datasets, the gap between fully Bayesian and EM methods only becomes apparent at higher levels of missingness, as in the BMM shapes dataset. We can also see that the Gibbs sampling approach tends to under perform in the higher dimensional datasets such as in the BMM shapes and MP voting datasets, as well as the GMM digits dataset (at higher levels of missingness). In these cases, Gibbs consistently trails VBEM, and in the case of the MP voting dataset and the GMM digits datasets, it performs similarly to or even worse than the one-step EM approach. This pattern suggests that for more complex datasets, Gibbs may require significantly more iterations to fully converge.

A noticeable trend is that the `KMeans+mean` approach often out-performs the `EM+mean` approach suggesting that ad-hoc mean imputation may be more detrimental within probabilistic frameworks than in deterministic, distance-based methods.

Complete-case algorithms perform worst across all datasets, often dropping to near-zero ARI values even once modest levels of missingness are introduced. This effect is especially severe in higher-dimensional datasets such as BMM shapes and GMM digits, where MCAR missingness leads to a rapid loss of complete rows. In contrast, these methods perform comparatively better on lower-dimensional datasets like GMM synthetic and GMM Iris (three features), where the probability of a row being dropped is smaller.

The clustering performance of the fully Bayesian methods on real-world datasets remains competitive with their performance on synthetic datasets, demonstrating their effectiveness beyond idealized conditions. Overall, fully Bayesian inference offers the most robust clustering performance under missing data, with benefits that become increasingly pronounced as missingness grows.

## 8.2 Model Fit (Log-Likelihood)

### 8.2.1 BMM Datasets

#### BMM Synthetic Dataset

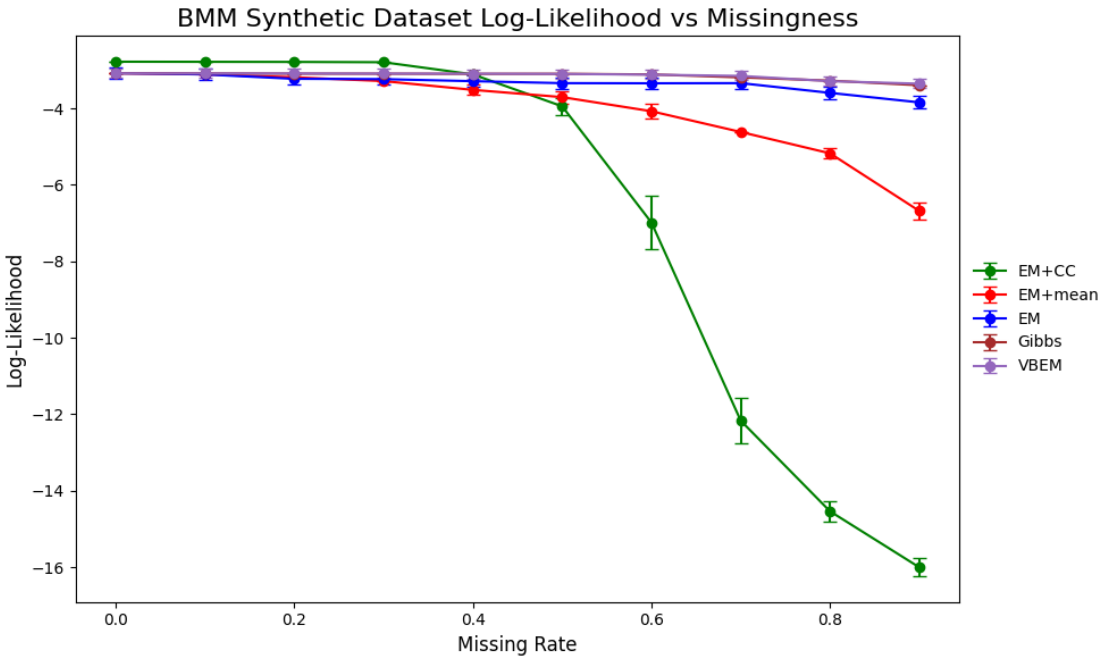



Figure 8.12: Model fit (log-likelihood) on complete training data on BMM synthetic dataset. Note that `VBEM` line overlaps with the `Gibbs` line.

#### BMM Shapes Dataset

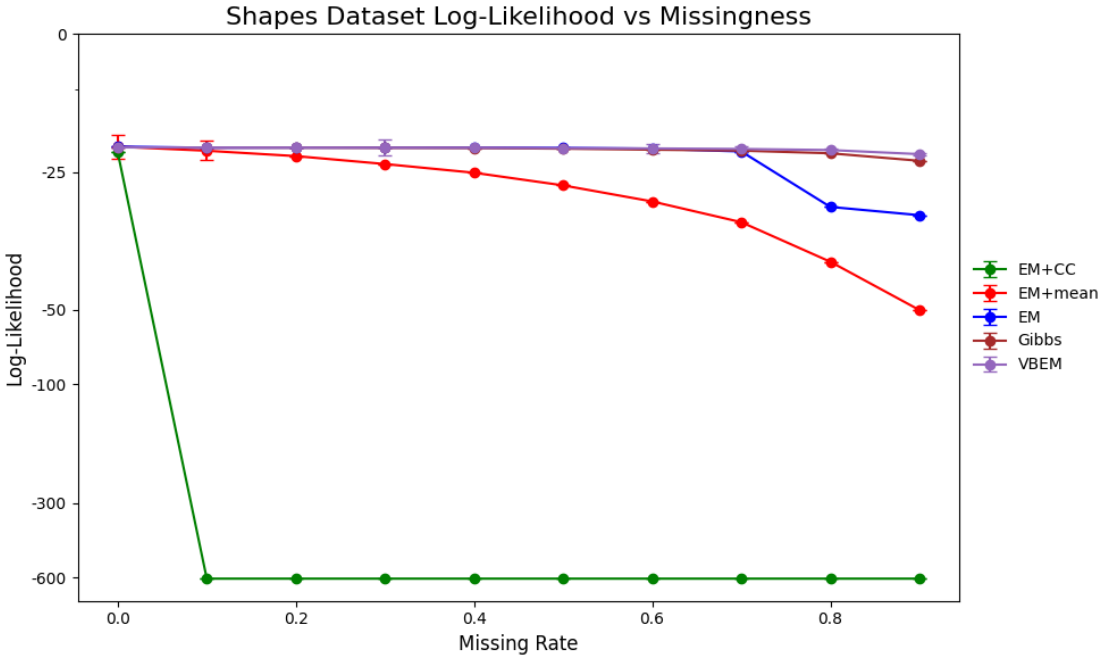



Figure 8.13: Model fit (log-likelihood) on complete training data on BMM shapes dataset. Note that a log-scale y-axis has been used and that `VBEM` line overlaps with the `Gibbs` line.

### 8.2.2 GMM Datasets

#### GMM Synthetic Dataset

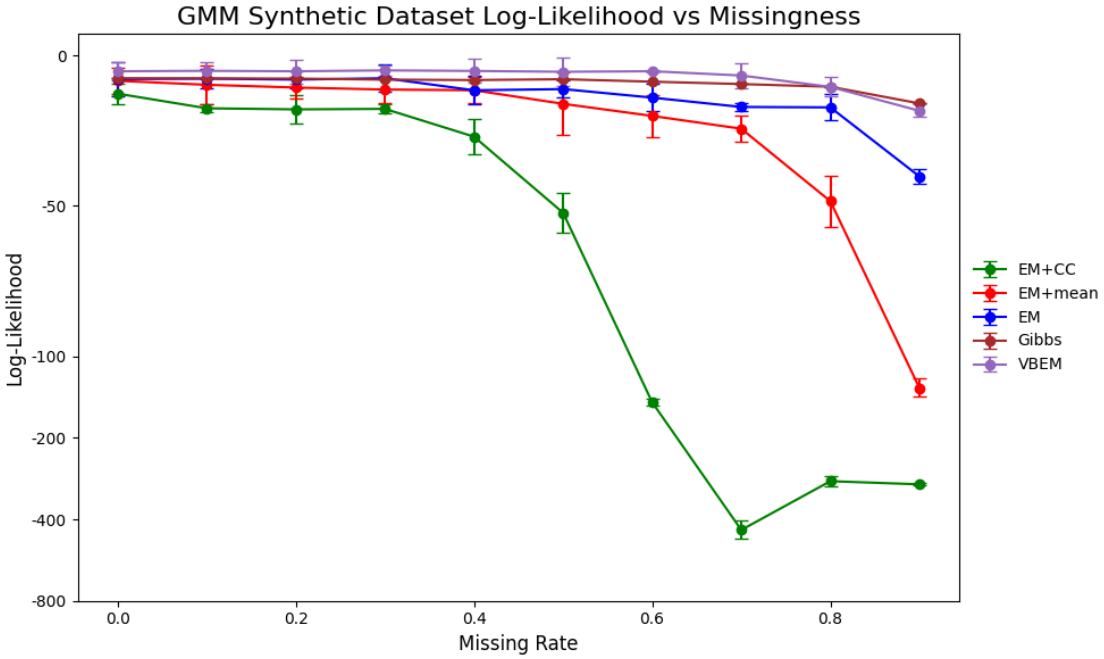



Figure 8.14: Model fit (log-likelihood) on complete training data on GMM synthetic dataset. Note that a log-scale y-axis has been used that `VBEM` line overlaps with the `Gibbs` line.

#### GMM Iris Dataset

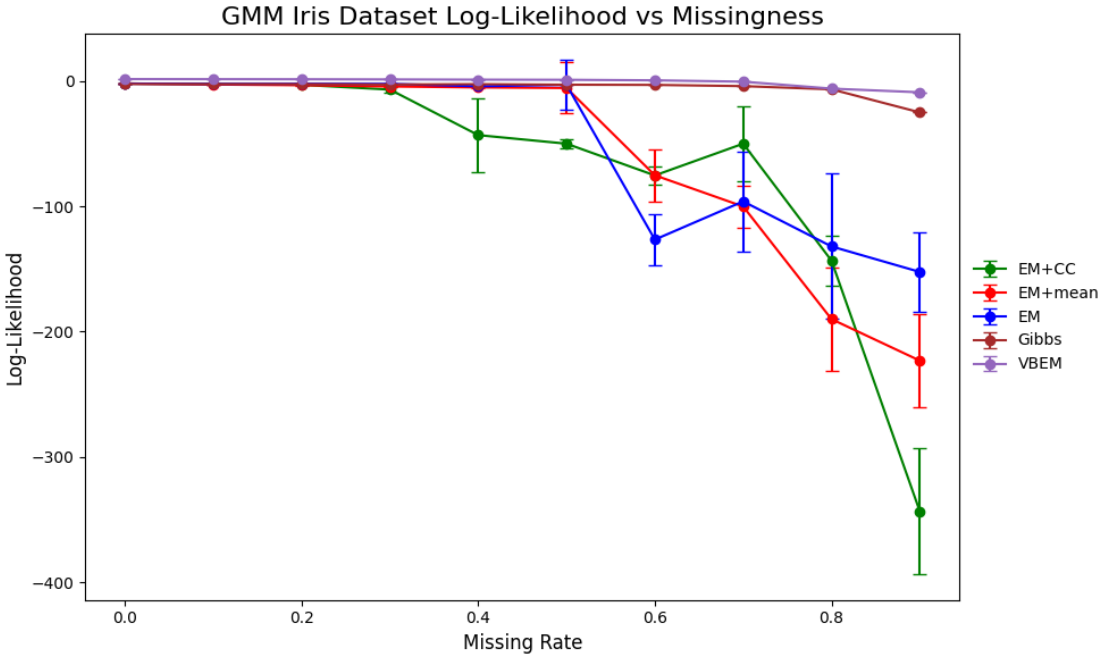



Figure 8.15: Model fit (log-likelihood) on complete training data on GMM Iris dataset. Note that `VBEM` line overlaps with the `Gibbs` line.

### GMM Digits Dataset

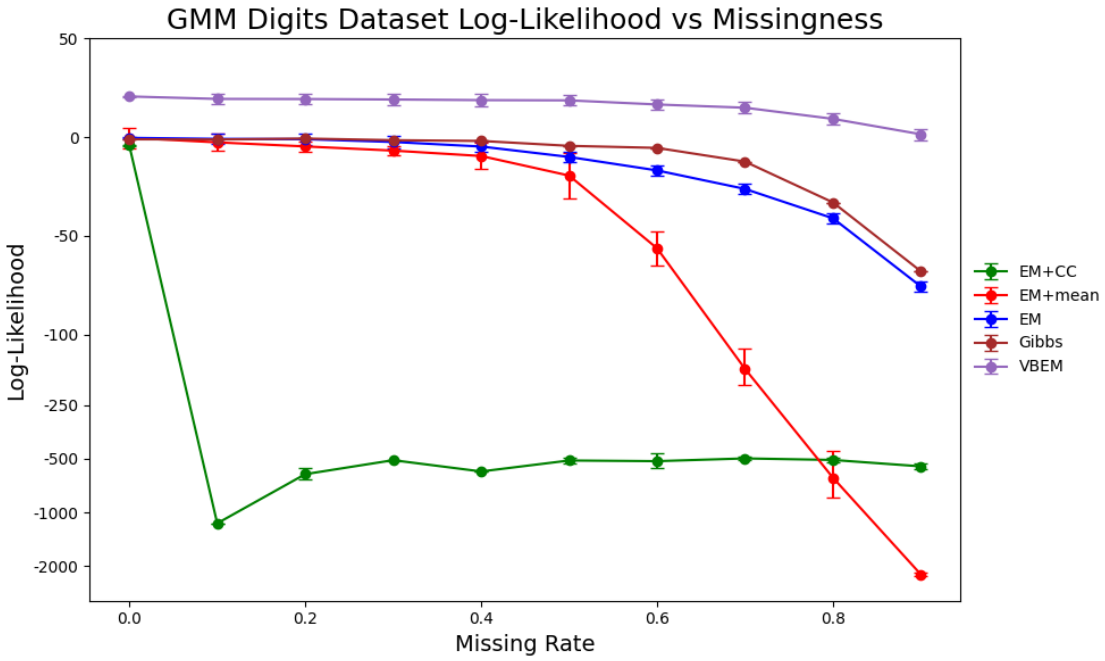



Figure 8.16: Model fit (log-likelihood) on complete training data on GMM digits dataset. Note that a log-scale y-axis has been used.

### GMM Iris Dataset

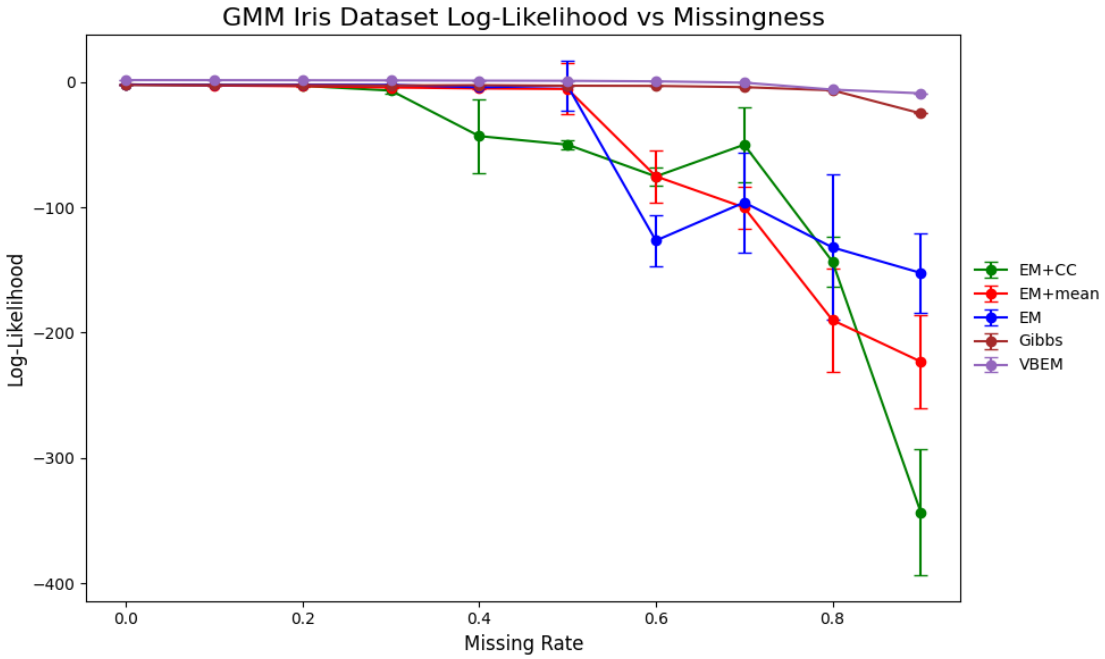



Figure 8.17: Model fit (log-likelihood) on complete training data on GMM Iris dataset.

## Summary

Across all datasets, the fully Bayesian methods (Gibbs and VBEM) consistently achieve the highest log-likelihood values, demonstrating their superior ability to recover model parame-

ters that generalize to the fully observed ground truth data. This advantage is particularly evident at high missingness levels ($\geq$ 70 percent), where probabilistic inference over both parameters and missing entries helps preserve model fidelity while other methods degrade sharply.

The one-step EM approach generally follows as the next best performer, maintaining competitive log-likelihoods up to moderate missingness (roughly 50-60 percent), but in many cases diverging from the Bayesian methods as missingness grows. Mean imputation paired with EM (`EM+mean`) performs worse, where parameter estimates become increasingly biased at higher missing rates.

The complete-case EM algorithm (`EM+CC`) exhibits consistently and substantially lower log-likelihood than all other approaches. A clear pattern emerges when comparing higher-dimensional datasets to lower-dimensional ones. In the BMM Shapes and GMM Digits datasets, log-likelihood drops sharply as soon as 10 percent missingness is introduced, then levels off as missingness increases. This behavior suggests that the number of complete cases collapses to near zero almost immediately, creating a worst-case scenario for complete-case analysis. Once this point is reached, further increases in missingness have little additional effect, as the probability of observing a fully complete row is already negligible. This pattern matches that of the clustering performance results where for these two datasets, ARI similarly drops to near-zero levels and stabilizes. Notably, in the GMM Digits dataset, the log-likelihood of the `EM+mean` imputation approach falls below that of `EM+CC`, suggesting that ad-hoc imputation can introduce sufficient bias to degrade performance below that of random chance.

Overall, these results reinforce the pattern seen in ARI. Fully Bayesian inference offers the most robust and stable model fit under missing data, where benefits become more clear at higher missingness levels.

## 8.3 Imputation Performance (RMSE)

### 8.3.1 BMM Datasets

#### BMM Synthetic Dataset

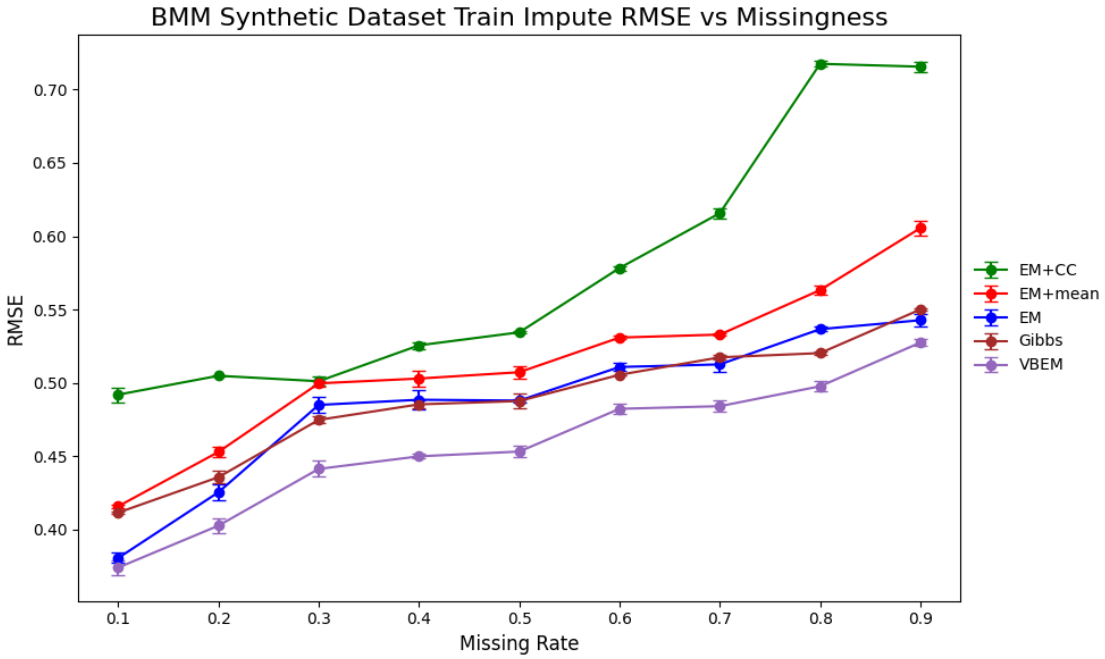



Figure 8.18: Imputation performance (RMSE) on **train** split of BMM synthetic dataset.

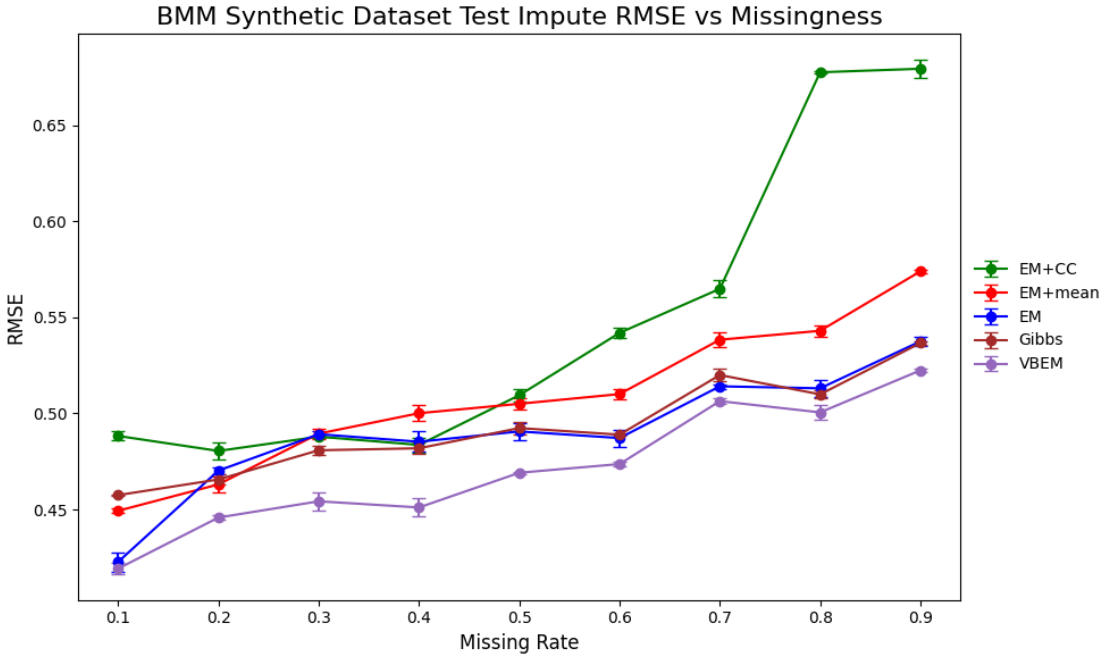



Figure 8.19: Imputation performance (RMSE) on held-out **test** split of BMM synthetic dataset.

### Shapes Dataset

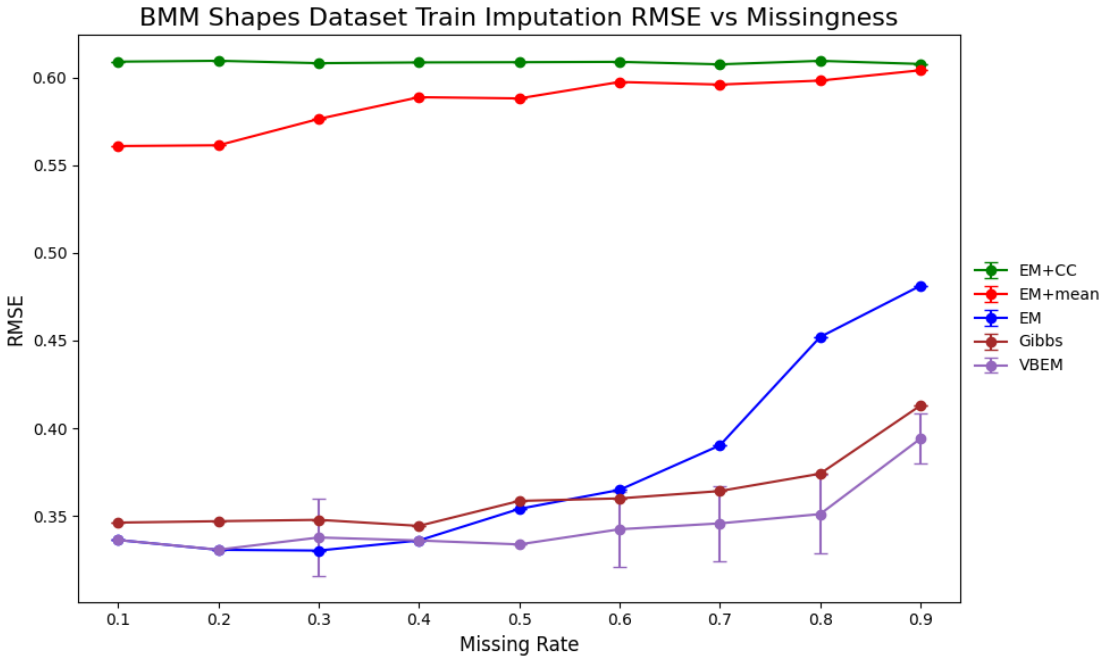



Figure 8.20: Imputation performance (RMSE) on **train** split of BMM shapes dataset.

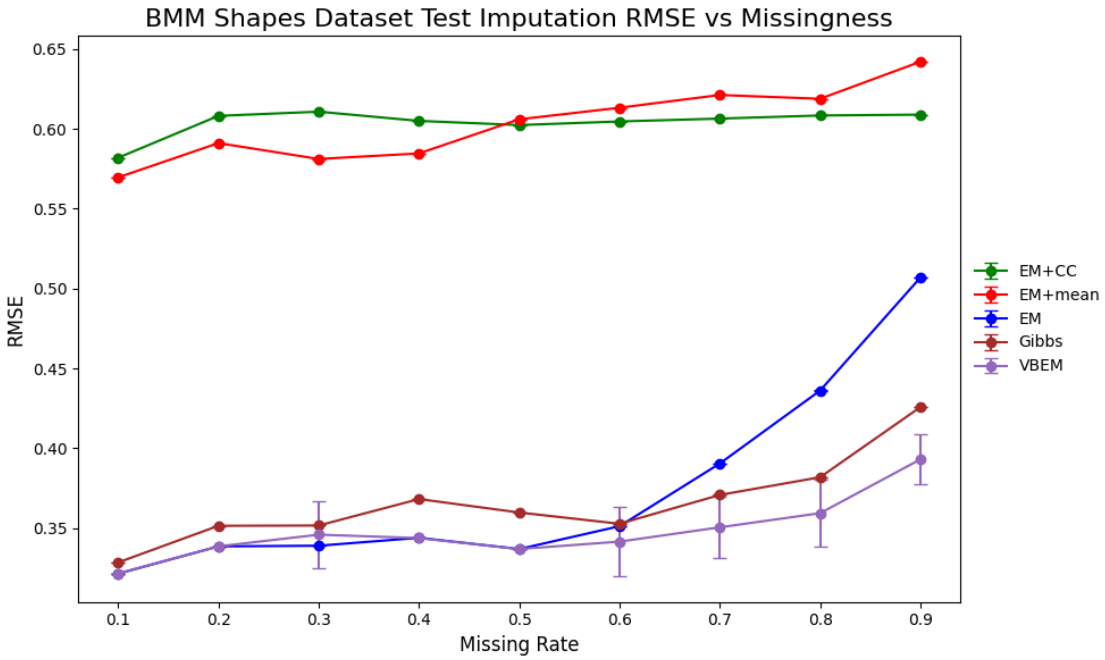



Figure 8.21: Imputation performance (RMSE) on held-out **test** split of BMM shapes dataset.

Using the Shapes dataset, we can visualize how imputation quality varies across algorithms and levels of missingness. In this experiment, each algorithm is trained on data with varying degrees of missingness, after which a candidate test sample is subjected to the same missingness pattern and reconstructed using the trained models. We present qualitative reconstructions for two representative scenarios: one with 30 percent missingness and another

with 80 percent missingness, illustrating how each method performs under mild and severe missing data conditions.

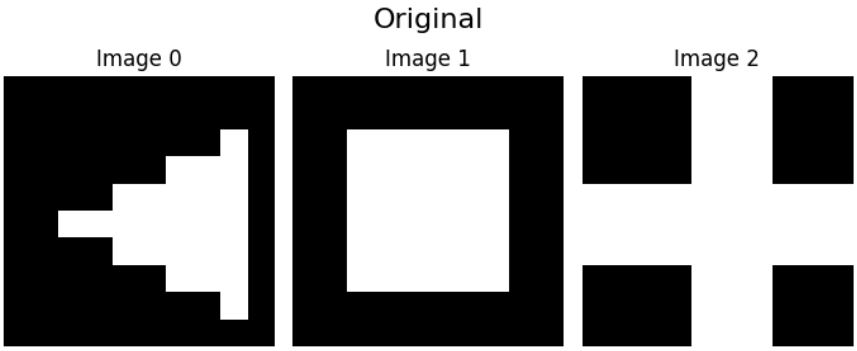



Figure 8.22: The original sample before applying missingness

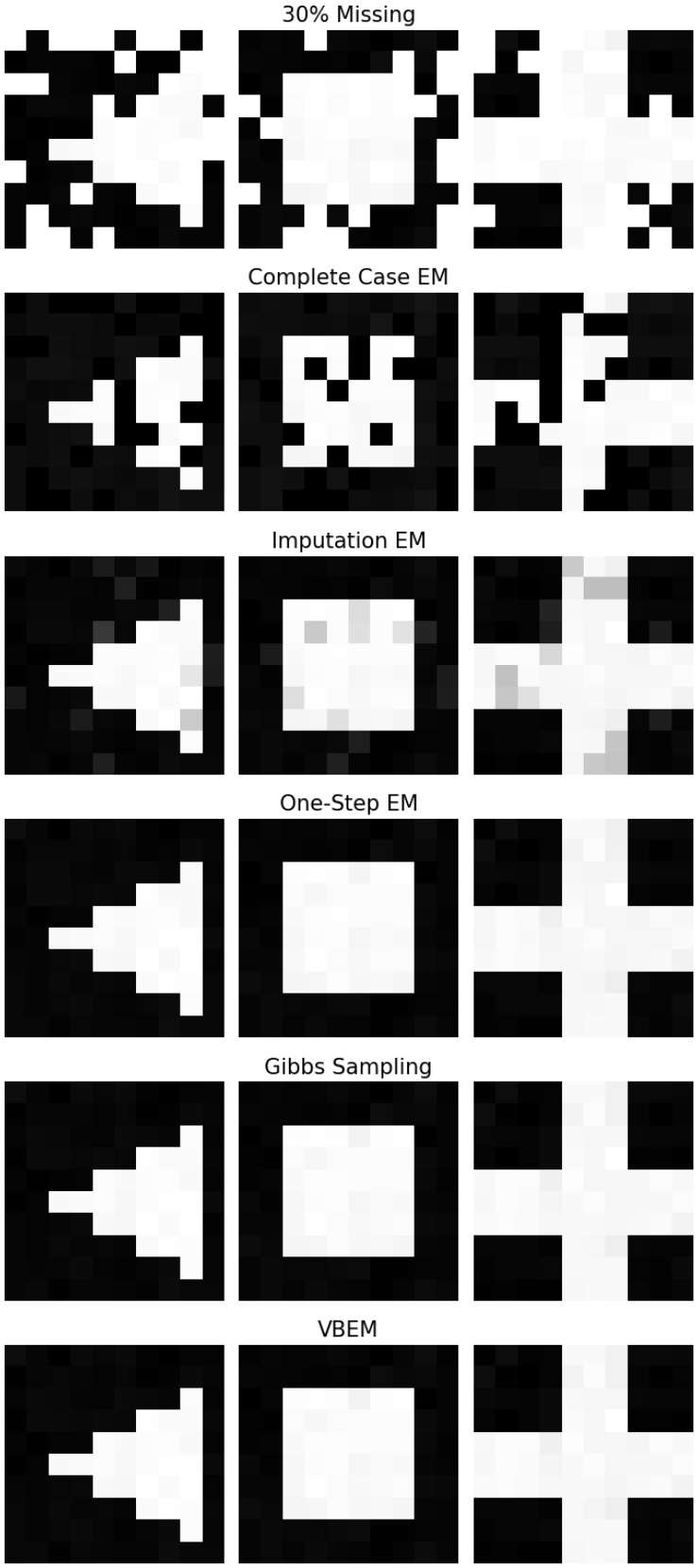



Figure 8.23: Top row shows the result of applying **30** percent MCAR missingness to the candidate sample. The lower 5 images show the imputation results from each algorithm with a correponding label on top of each image.

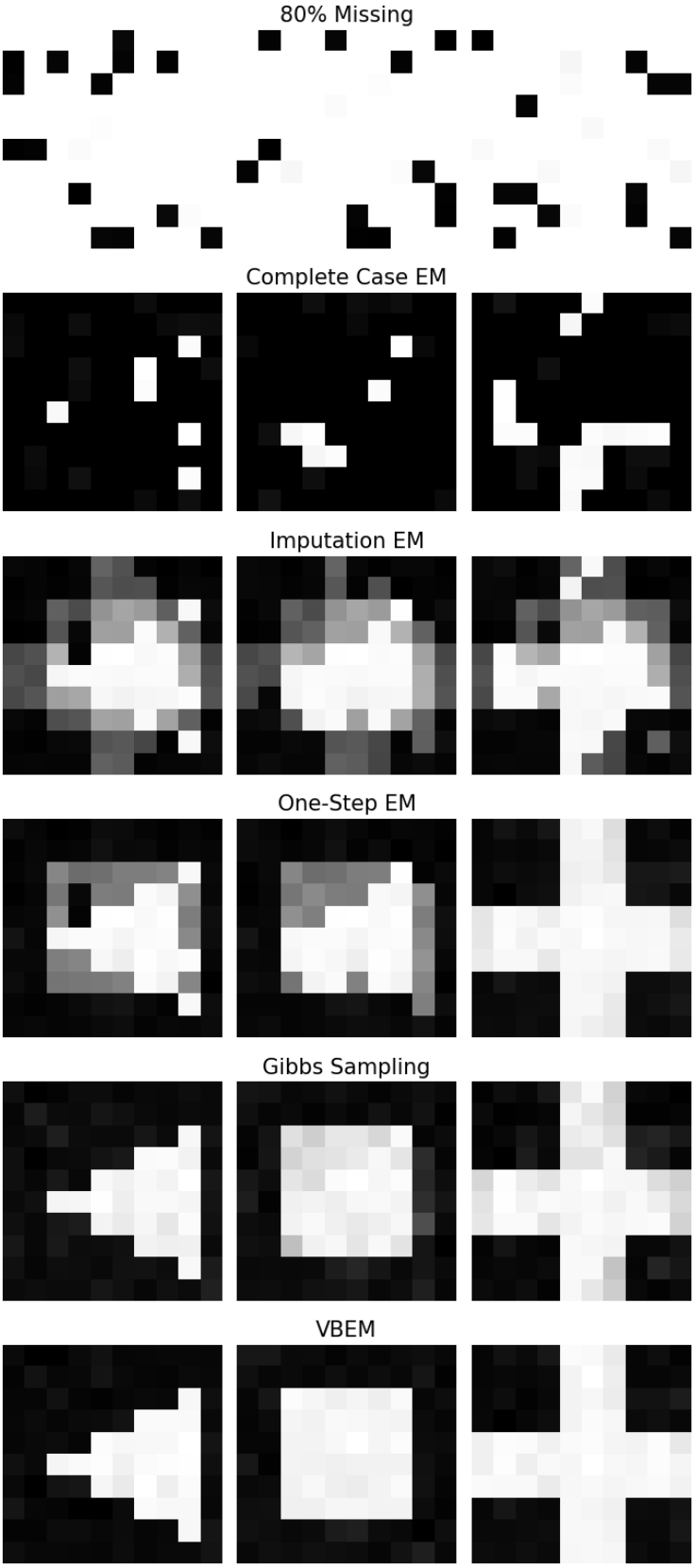



Figure 8.24: Top row shows the result of applying **80** percent MCAR missingness to the candidate sample. The lower 5 images show the imputation results from each algorithm with a correponding label on top of each image.

### 8.3.2 GMM Datasets

#### GMM Synthetic Dataset

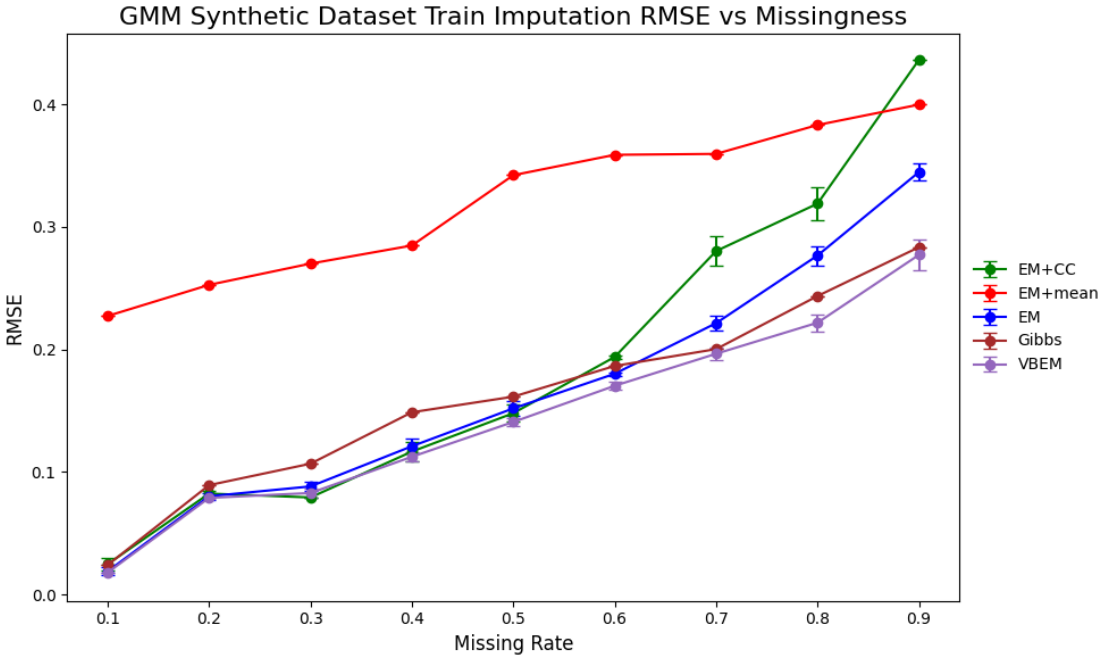



Figure 8.25: Imputation performance (RMSE) on **train** split of GMM Synthetic dataset.

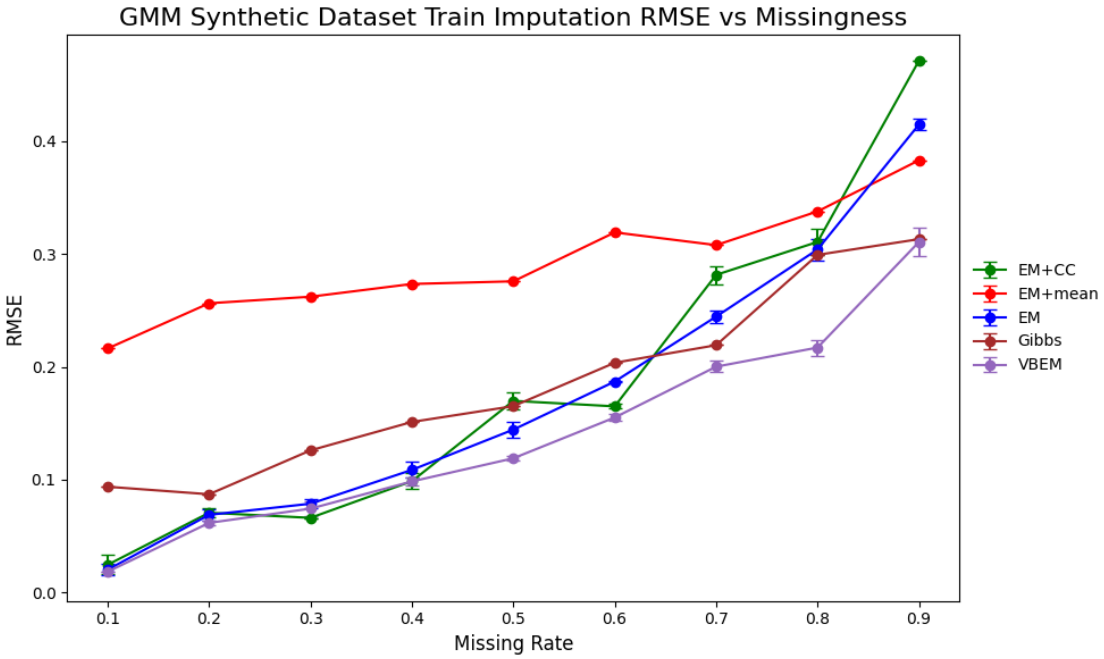



Figure 8.26: Imputation performance (RMSE) on held-out **test** split of GMM Synthetic dataset.

### GMM Digits Dataset

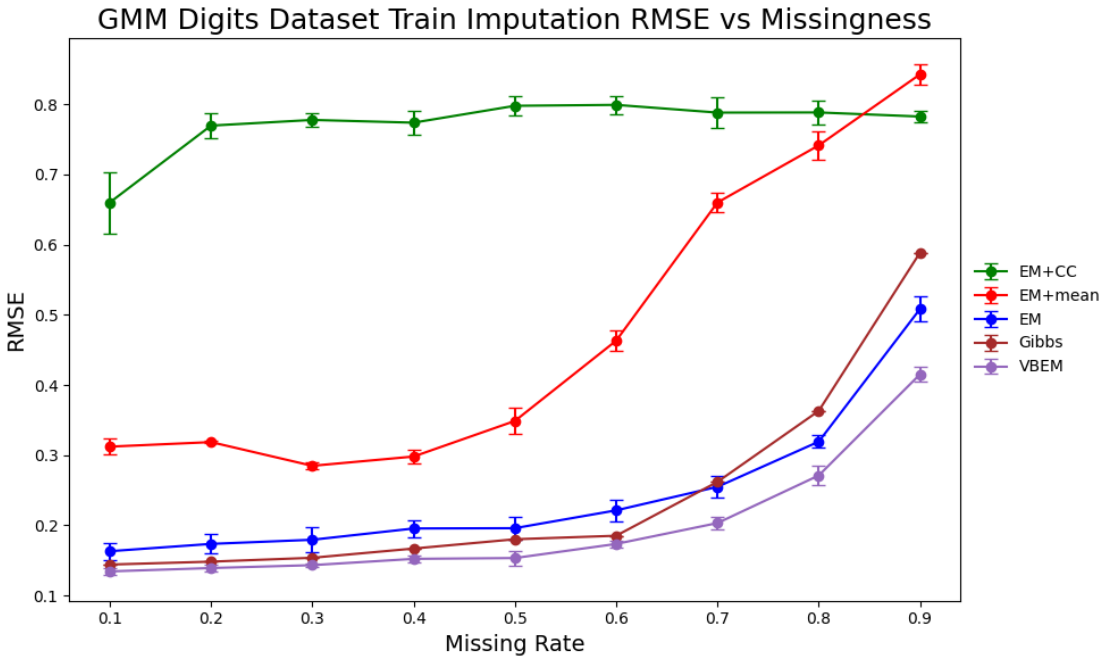



Figure 8.27: Imputation performance (RMSE) on **train** split of GMM digits dataset.

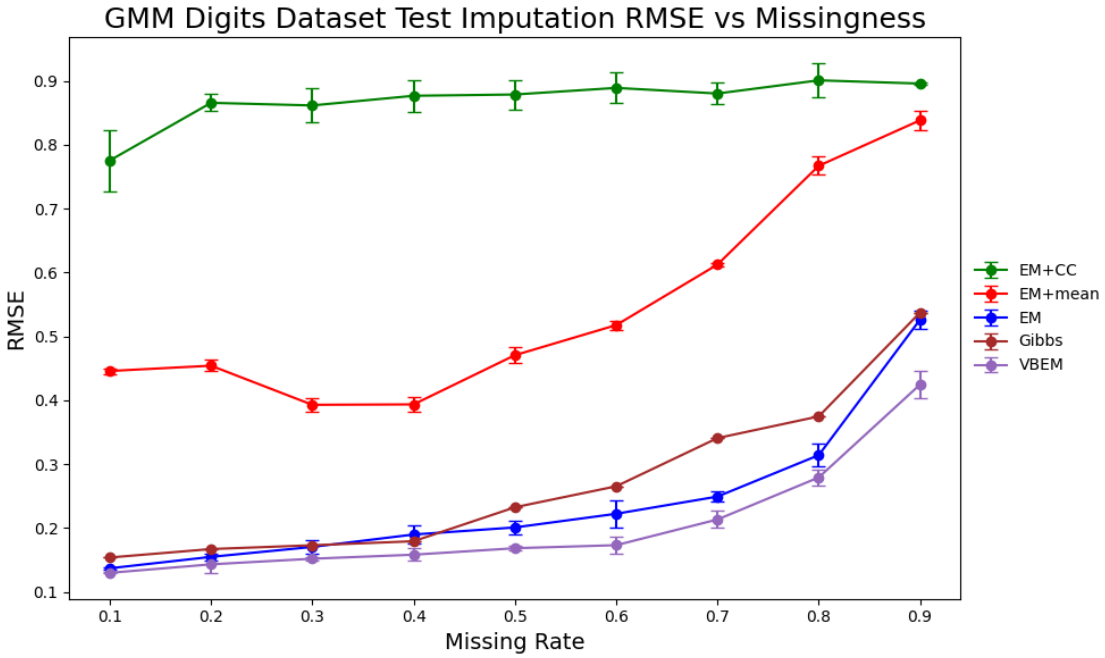



Figure 8.28: Imputation performance (RMSE) on held-out **test** split of GMM digits dataset.

Similar to the BMM shapes dataset, we can visualize the imputation quality. Below we conduct the same experiment where in we present qualitative reconstructions for two representative scenarios : one with 30 percent missingness, and another with 80 percent missingness.

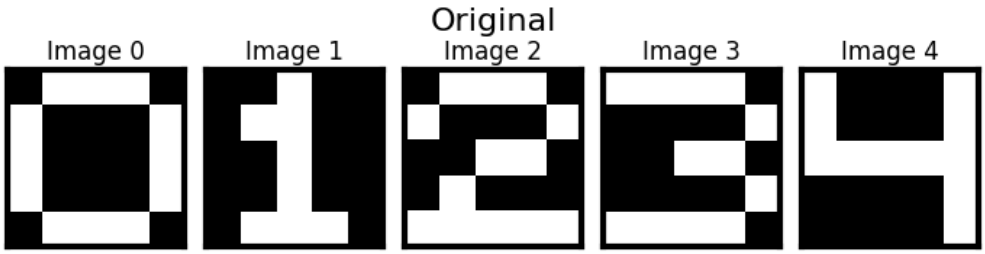



Figure 8.29: The original candidate sample before applying missingness

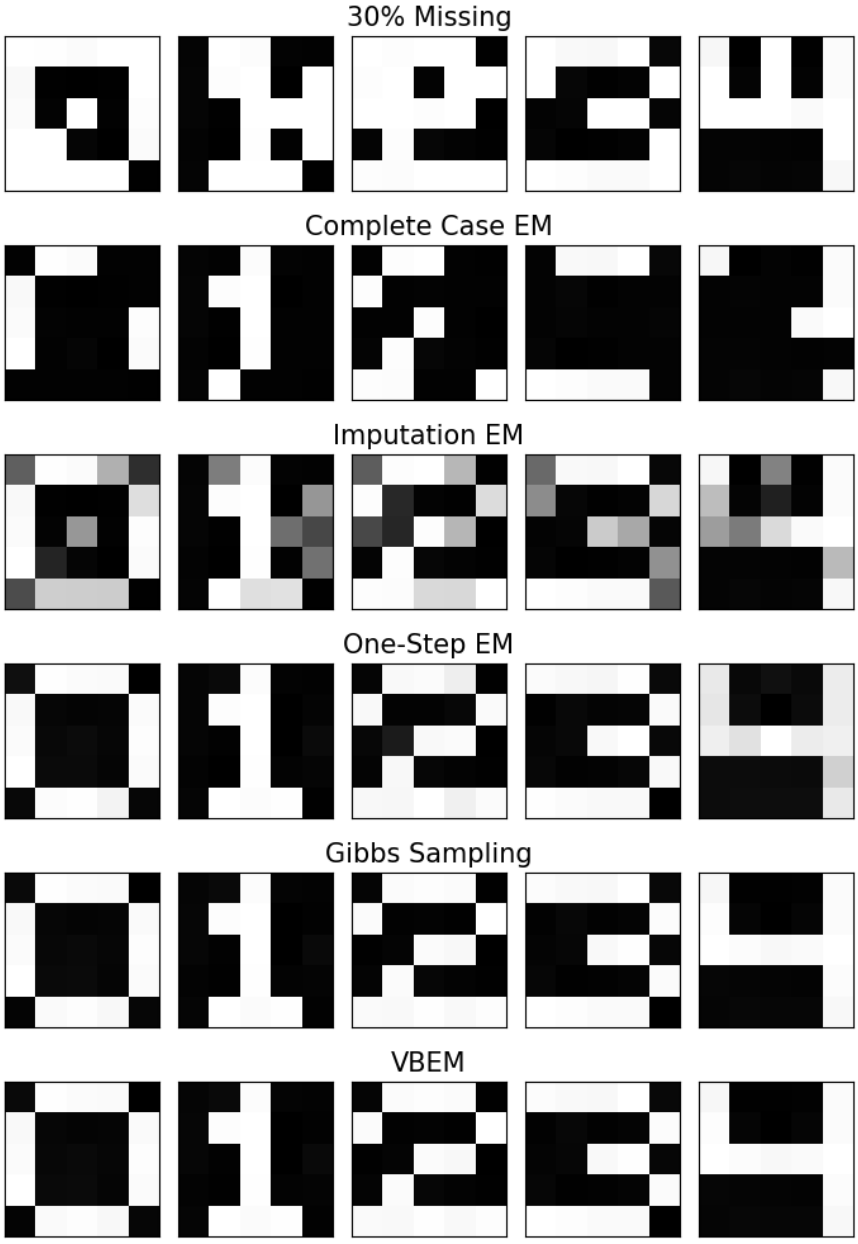



Figure 8.30: Top row shows the result of applying 30 percent MCAR missingness to the candidate sample. The lower 5 images show the imputation results from each algorithm with a corresponding label on top of each image.

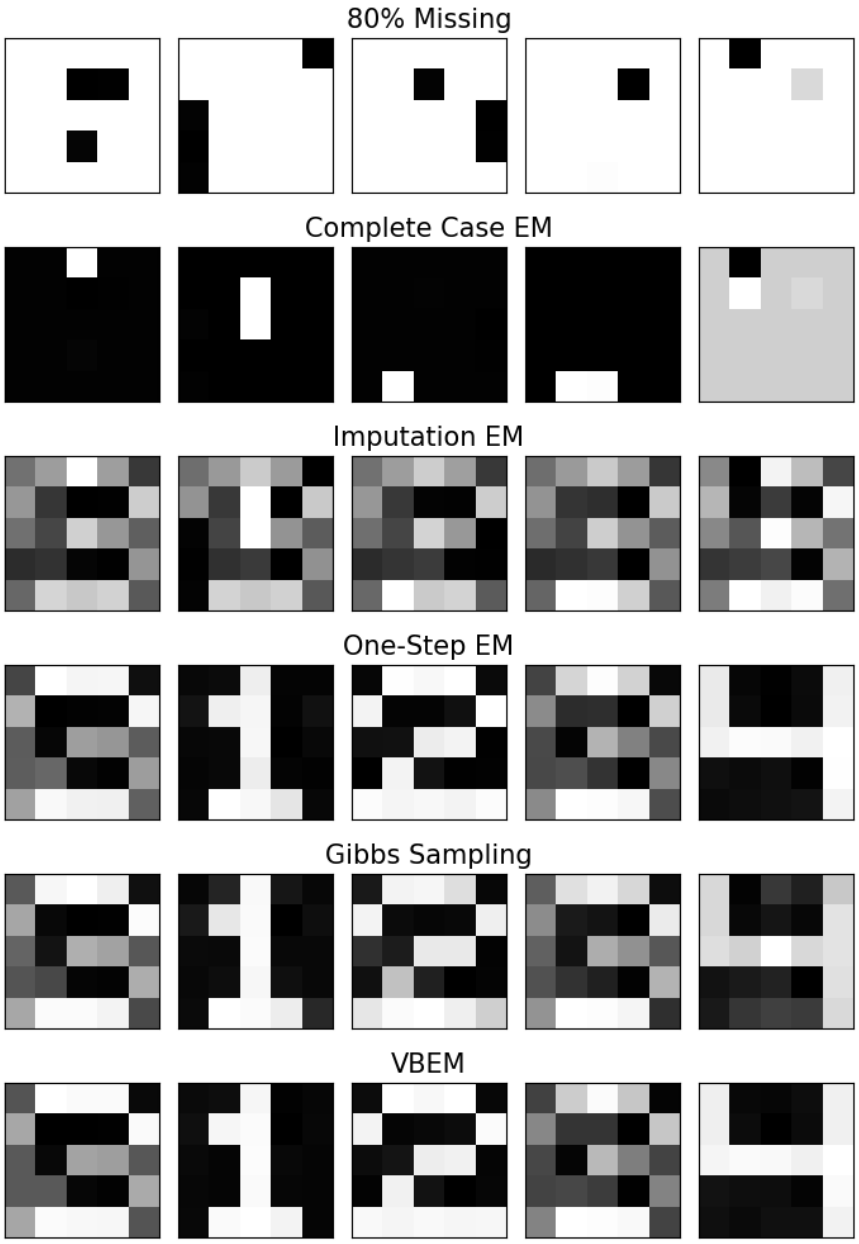



Figure 8.31: Top row shows the result of applying 80 percent MCAR missingness to the candidate sample. The lower 5 images show the imputation results from each algorithm with a corresponding label on top of each image.

### GMM Iris Dataset

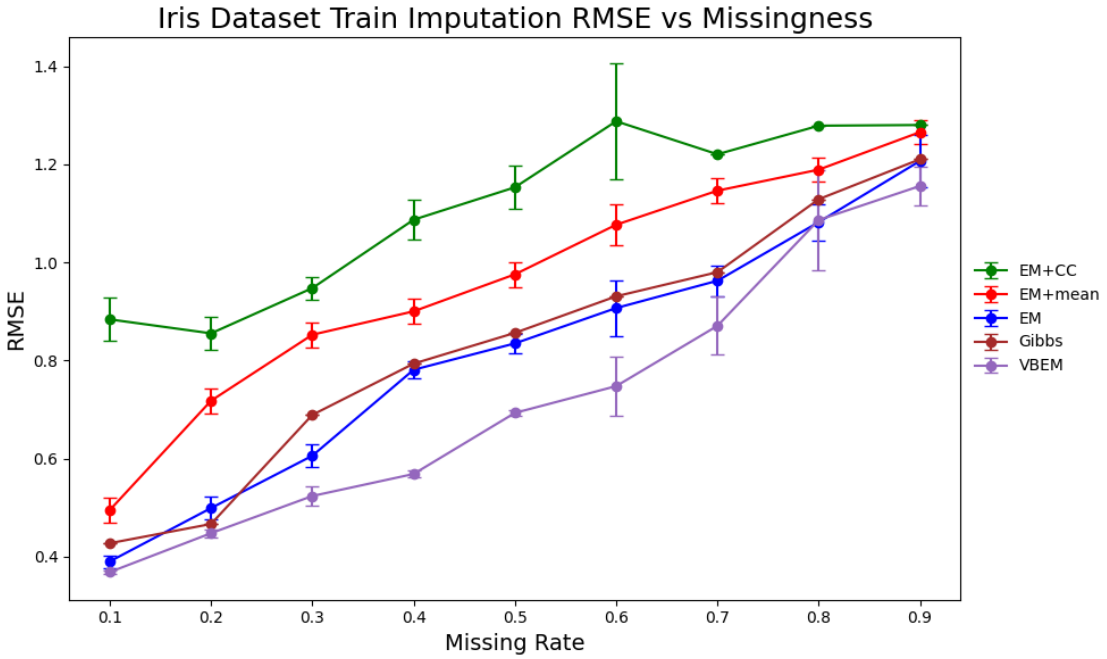



Figure 8.32: Imputation performance (RMSE) on **train** split of GMM Iris dataset.

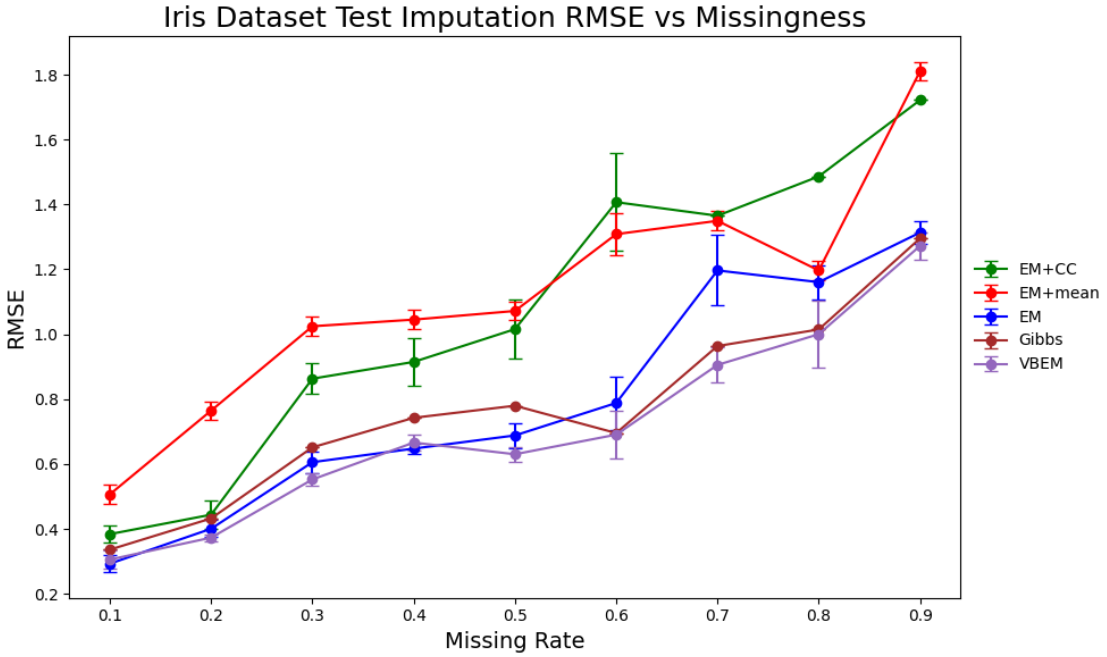



Figure 8.33: Imputation performance (RMSE) on held-out **test** split of GMM Iris dataset.

## Summary

Across all datasets, the complete-case EM approach (`EM+CC`) consistently yields the highest RMSE, often by a substantial margin, which confirms its poor performance under missing data. The degradation is particularly significant for high-dimensional datasets (e.g., GMM Digits, BMM Shapes), where the probability of a fully complete case is low, even at moderate

missingness levels. This effect mirrors the earlier log-likelihood results where once complete cases vanish, further missingness produces minimal additional degradation in performance.

The `EM+mean` approach generally performs worse than the fully Bayesian (`Gibbs, VBEM`) and one-step EM (`EM`) methods but better than `EM+CC`, except in specific high-dimensional cases (notably GMM Digits) where mean imputation introduces biases to reconstructions sufficiently to rival or even under-perform compared to complete-case analysis.

For the lower-dimensional datasets (Iris, BMM Synthetic), differences between EM, Gibbs, and VBEM are more modest, with VBEM often achieving the lowest RMSE, especially in the mid-to-high missingness range. Similar to the results of clustering performance and model log-likelihood, the imputation performance of Gibbs degrades for higher-dimensional datasets (e.g shapes and digits datasets) in the higher missingness ranges, likely due to insufficient samples to fully explore the posterior probability space with respect to the complexity of the data.

In general, VBEM appears to be the top performer consistently across the datasets, showing low absolute RMSE values and slow growth in response to increasing missingness. This is followed closely by the Gibbs sampling approach which out-performs the one-step EM approach with the exception of the complex higher-dimensional datasets for higher levels of missingness. These results are also confirmed by the qualitative results for the BMM shapes and GMM digits data sets in Figures 8.24 and 8.31. In both cases (shapes and digits) where 80 percent missingness is introduced to the candidate sample, we can start to see a noticeable difference in imputation quality between the Gibbs result and VBEM. In the case of the digits imputation experiment, the Gibbs imputation result appears noisier than that of the one-step EM result in Figure8.31.

A notable trend across datasets is that RMSE growth with missingness is not always monotonic in the non-Bayesian approaches. For example, `EM+mean` sometimes shows flat or erratic behavior at low missingness before worsening sharply. Bayesian methods tend to show smoother and more gradual increases in RMSE, reflecting their stability under missing data.

## 8.4 Summarizing Analysis

Across all evaluation metrics, clustering performance (ARI), model fit (log-likelihood), and imputation quality (RMSE), the fully Bayesian approaches (`Gibbs` and `VBEM`) consistently demonstrate superior performance under missing data. Their advantages are most pronounced in high-dimensional datasets, where the probability of complete cases rapidly approaches zero and uncertainty in missing values plays a critical role. VBEM often edges out Gibbs in these settings. This result is largely dependent on the number of iterations the Gibbs sampling algorithm is executed with in these experiments.

The one-step EM algorithm (`EM`) generally performs well at lower and moderate levels of missingness, sometimes matching the fully Bayesian methods in simpler, low-dimensional datasets. However, its performance declines more sharply as missingness increases, highlighting the loss in inference fidelity from maximum likelihood estimation as opposed to

fully Bayesian modelling. Ad-hoc mean imputation (`EM+mean`, `KMeans+mean`) performs reasonably in low-dimensional cases but introduces bias and instability in higher-dimensional settings, occasionally under-performing even complete-case analysis.

Complete-case approaches (`EM+CC`, `KMeans+CC`) consistently rank lowest across all metrics, with severe degradation at even the lowest missingness levels. This effect is most visible in high-dimensional datasets, where the number of complete rows becomes negligible almost immediately, rendering these methods unusable in practice. Interestingly, the results on the GMM digit dataset show that the EM algorithm with mean imputation at high missingness drops below complete-case EM in performance, suggesting that bias the introduced from ad-hoc imputation may degrade performance below that of random chance. This sheds light on the potential risks of naive imputation strategies.

The consistency of trends across ARI, log-likelihood, and RMSE suggests a strong relationship between model fit, predictive clustering accuracy, and the ability to reconstruct missing data accurately. Fully Bayesian methods, by integrating over posterior uncertainty, maintain stability and accuracy where other approaches collapse. This makes the fully Bayesian the most robust choice for inference with missing data, particularly as dimensionality and missingness increase.

The results also show similar performance across clustering performance and imputation accuracy metrics between the training and test sets, suggesting that the models generalize well to unseen data, with no clear evidence of overfitting. We provide test-vs-train clustering performance trace-plots for the fully Bayesian VBEM and Gibbs sampling algorithms for each dataset in appendix Section 11.2. Notably from these figures, the training and test performance trajectories track very closely, with differences in ARI being small and stable. While K-fold cross-validation would provide a stronger assessment of generalizability, it was computationally prohibitive within the time frame of this study.

# Chapter 9

# MNAR Exploratory Extension

A potential solution to handle a restricted MNAR assumption has been explored wherein the nature of missingness of a data-point is dependent on the latent component to which the data-point belongs to. As this extension is not a part of the main focus of this research, evaluation and implemention is limited and cover only a limited number of cases under this MNAR assumption. However, the results of the evaluation show potential promise for a more flexible solution capable of handling both MNAR missingness and MCAR missingness.

## 9.1 MNAR Generative Model

In this generative approach, we treat the missingness pattern of the data as an additional observed variable alongside the primary data. Formally, Let $\mathbf{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_N\}$ be a binary mask for $\mathbf{X}$, where $m_{id} = 1$ if $x_{id}$ is observed and $m_{id} = 0$ if $x_{id}$ is missing. Hence we can define a multivariate Bernoulli distribution over missing masks in a similar manner to the BMM case. Conditional on the latent assignment $z_i = k$, we model the mask with independent Bernoulli variables parameterized by $\boldsymbol{\gamma}_k = (\gamma_{k1}, \ldots, \gamma_{kD})$:

$$p(\mathbf{m}^{(i)} \mid z^{(i)} = k, \boldsymbol{\gamma}_k) = \prod_{d=1}^{D} \text{Bern}\Big(m_d^{(i)} \mid \gamma_{kd}\Big), \quad \gamma_{kd} \in [0, 1].$$

(We place Beta priors on $\gamma_{kd}$ in the usual way.)

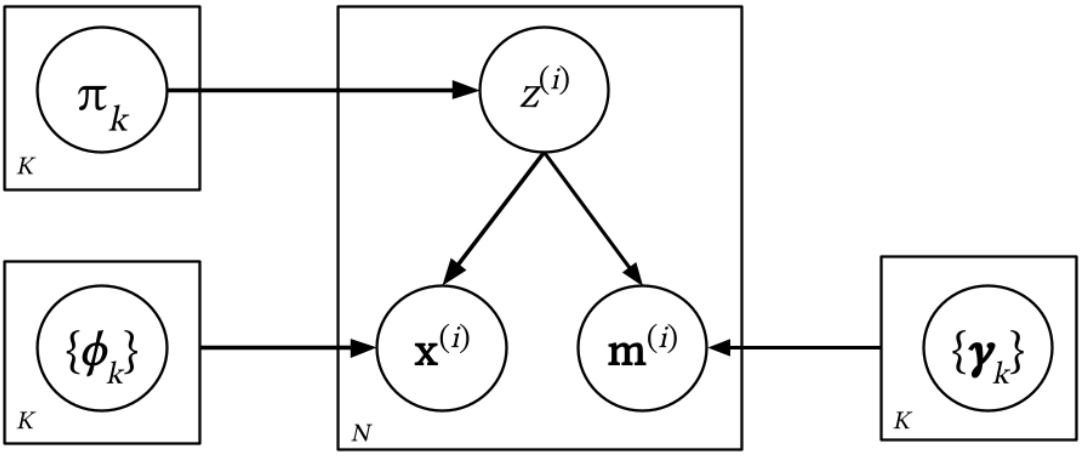



Figure 9.1: Bayesian network for MNAR generative model

Marginalizing over the latent assignment, the complete mixture model is given by:

$$p(\mathbf{m}^{(i)} \mid \boldsymbol{\pi}, \boldsymbol{\gamma}) = \sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} \text{Bern}(m_d^{(i)} \mid \gamma_{kd}) \tag{9.1}$$

With this, the observed data likelihood over all data points $\mathbf{M}$ factorizes as follows

$$p(\mathbf{M} \mid \boldsymbol{\gamma}, \boldsymbol{\pi}) = \prod_{i}^{N} \sum_{k}^{K} \pi_k \prod_{d}^{D} \text{Bern}(m_d^{(i)} \mid \gamma_{kd}) \tag{9.2}$$

If we jointly infer missing data $\mathbf{X}_h$ (with $\mathbf{X}_o$ observed), the augmented posterior is

$$p(\boldsymbol{\pi}, \boldsymbol{\phi}, \boldsymbol{\gamma}, \mathbf{z}, \mathbf{X}_h \mid \mathbf{X}_o, \mathbf{M}) \propto p(\mathbf{X}_o, \mathbf{X}_h \mid \boldsymbol{\phi}, \mathbf{z})\, p(\mathbf{M} \mid \boldsymbol{\gamma}, \mathbf{z})\, p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi})\, p(\boldsymbol{\phi})\, p(\boldsymbol{\gamma}).$$

This effectively adds a second source of evidence ($\mathbf{M}$) for the latent assignments, via $p(\mathbf{M} \mid z_i, \boldsymbol{\gamma})$ in the updates. Under MCAR this factor is constant and drops out. Under the the MNAR assumption, it can better inform responsibilities and improve clustering performance.

## 9.2 Implementation

Due to time constraints, this research only explores the implementation of this generative approach using Gibbs sampling, although implementation for VBEM should be straightforward. In this section, we outline the details of the algorithm for both BMMs and GMMs.

### 9.2.1 BMM Gibbs Sampling for MNAR

In the BMM approach with missing data, the full joint distribution factors into the following

$$\begin{aligned} p(\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\pi}, \mathbf{X}_h) \propto\ & p(\mathbf{X}_o \mid \boldsymbol{\theta}, \mathbf{z})\, p(\mathbf{X}_h \mid \boldsymbol{\theta}, \mathbf{z})\, p(\mathbf{M} \mid \boldsymbol{\gamma}, \mathbf{z}) \\ & \times p(\boldsymbol{\theta} \mid \mathbf{a}_0, \mathbf{b}_0)\, p(\boldsymbol{\gamma} \mid \mathbf{u}_0, \mathbf{v}_0)\, p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}_0) \end{aligned} \tag{9.3}$$

Leveraging conjugacy, we can define the following closed-form full conditionals to sample from

1. Mixing weights $\boldsymbol{\pi}$

   $$p(\boldsymbol{\pi} \mid \mathbf{z}) = \text{Dir}(\boldsymbol{\pi} \mid \alpha_{0,1} + N_1, \cdots, \alpha_{0,K} + N_K)$$

   Where $N_k = \sum_{i=1}^{N} \mathbb{1}(z_i = k)$

2. Bernoulli bias over data $\mathbf{X}$ for each component $k$ and dimension $d$ $\theta_{k,d}$

   $$p(\theta_{kd} \mid \boldsymbol{X}, \mathbf{z}) = \text{Beta}(\theta_{kd} \mid a_{0d} + N_{kd}^{(1)}, b_{0d} + N_{kd}^{(0)})$$

   where

   - $N_{kd}^{(1)} = \sum_{i}^{N} \mathbb{1}(x_{id} = 1, m_{id} = 1, z_i = k)$
   - $N_{kd}^{(0)} = \sum_{i}^{N} \mathbb{1}(x_{id} = 0, m_{id} = 1, z_i = k)$

3. Bernoulli bias $\gamma_{k,d}$ over missing masks $\mathbf{M}$ for each component $k$ and dimension $d$

$$p(\gamma_{kd} \mid \boldsymbol{M}, \mathbf{z}) = \text{Beta}(\gamma_{kd} \mid u_{0d} + N_{kd}^{(1)}, v_{0d} + N_{kd}^{(0)})$$

where

- $N_{kd}^{(1)} = \sum_i^N \mathbb{1}(m_{id} = 1, z_i = k)$
- $N_{kd}^{(0)} = \sum_i^N \mathbb{1}(m_{id} = 0, z_i = k)$

4. Latent component assignments for each for each data-point $i$ and component $k$

$$p(z_i = k \mid \mathbf{x}_i, \mathbf{m}_i, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\pi}) = \pi_k \prod_d^D \left[\theta_{kd}^{x_{id}}(1-\theta_{kd})^{(1-x_{id})} \times \gamma_{kd}^{m_{id}}(1-\gamma_{kd})^{(1-m_{id})}\right]$$

Normalizing over all component gives the sampling update

$$r_{ik} = \text{Cat}(z_i = k \mid \mathbf{x}_i, \boldsymbol{\theta}, \boldsymbol{\pi}) = \frac{\pi_k \prod_d^D \left[\theta_{kd}^{x_{id}}(1-\theta_{kd})^{(1-x_{id})} \times \gamma_{kd}^{m_{id}}(1-\gamma_{kd})^{(1-m_{id})}\right]}{\sum_j^K \pi_j \left[\prod_d^D \theta_{jd}^{x_{id}}(1-\theta_{jd})^{(1-x_{id})} \times \gamma_{jd}^{m_{id}}(1-\gamma_{jd})^{(1-m_{id})}\right]}$$

With full conditionals for all variables in the joint distribution, we perform the following sampling steps at each iteration $t$ of the Gibbs sampler:

$$\text{Sample } \boldsymbol{\pi}^{(t)} \sim \text{Dir}\left(\alpha_{0,1} + N_1^{(t-1)}, \ldots, \alpha_{0,K} + N_K^{(t-1)}\right)$$

$$\text{Sample } \theta_{kd}^{(t)} \sim \text{Beta}\left(a_{0,k} + N_{kd}^{(1)},\ b_{0,k} + N_{kd}^{(0)}\right), \quad \text{for all } k \in [K],\ d \in [D]$$

$$\text{Sample } \gamma_{kd}^{(t)} \sim \text{Beta}\left(u_{0,k} + N_{kd}^{(1)},\ v_{0,k} + N_{kd}^{(0)}\right), \quad \text{for all } k \in [K],\ d \in [D]$$

$$\text{Sample } z_i^{(t)} \sim \text{Categorical}\left(p(z_i = k \mid \mathbf{x}_i, \mathbf{m}_i, \boldsymbol{\theta}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\pi}^{(t)})\right), \quad \text{for all } i \in [N]$$

Note that this constitutes a partially collapsed Gibbs sampling regime where we marginalize out the missing entries of the data $\mathbf{X}$.

### 9.2.2 GMM Gibbs Sampling for MNAR

In the GMM approach with missing data, the full joint distribution factors into the following

$$p(\mathbf{z}, \boldsymbol{\phi}, \boldsymbol{\gamma}, \mathbf{X}_h) = p(\mathbf{X}_o \mid \boldsymbol{\phi}, \mathbf{z})\, p(\mathbf{X}_h \mid \mathbf{X}_h, \boldsymbol{\phi}, \mathbf{z})\, p(\mathbf{M} \mid \boldsymbol{\gamma}, \mathbf{z}) \tag{9.4}$$

$$\times\, p(\boldsymbol{\phi} \mid \mathbf{m}_0, \boldsymbol{\kappa}_0, \mathbf{S}_0, \boldsymbol{\nu}_0)\, p(\boldsymbol{\gamma} \mid \mathbf{u}_0, \mathbf{v}_0)\, p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}_0) \tag{9.5}$$

where $\boldsymbol{\phi} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$.

Leveraging conjugacy, we can define the following closed-form full conditionals to sample from

1. Mixing weights $\boldsymbol{\pi}$

$$p(\boldsymbol{\pi} \mid \mathbf{z}) = \text{Dir}(\boldsymbol{\pi} \mid \alpha_{0,1} + N_1, \cdots, \alpha_{0,K} + N_K)$$

Where $N_k = \sum_{i=1}^N \mathbb{1}(z_i = k)$

2. Gaussian parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma} \mid \mathbf{X}, \mathbf{z}) = \prod_{k}^{K} \text{NIW}(\mu_k, \Sigma_k | \mathbf{m}_k, \kappa_k, \nu_k, \mathbf{S}_k) \tag{9.6}$$

$$\mathbf{m}_k = \frac{\kappa_{0_k} m_{0_k} + N_k \bar{\mathbf{x}}_k}{\kappa_{0_k} + N_k}$$

$$\kappa_k = \kappa_0 + N_k$$

$$\nu_k = \nu_0 + N_k$$

$$\mathbf{S}_k = \mathbf{S}_{0_k} + \mathbf{S}_{\bar{\mathbf{x}}_k} + \frac{\kappa_{0_k} N_k}{\kappa_{0_k} + N_k} (\bar{\mathbf{x}}_k - m_{0_k})(\bar{\mathbf{x}}_k - m_{0_k})^T$$

3. Bernoulli bias over missing masks $\mathbf{M}$ for each component $k$ and dimension $d$ $\gamma_{k,d}$

$$p(\gamma_{kd} \mid \boldsymbol{M}, \mathbf{z}) = \text{Beta}(\gamma_{kd} \mid u_{0d} + N_{kd}^{(1)}, v_{0d} + N_{kd}^{(0)})$$

where

- $N_{kd}^{(1)} = \sum_i^N \mathbb{1}(m_{id} = 1, z_i = k)$
- $N_{kd}^{(0)} = \sum_i^N \mathbb{1}(m_{id} = 0, z_i = k)$

4. Latent component assignments for each for each data-point $i$ and component $k$

$$p(z_i = k \mid \mathbf{x}_i, \mathbf{m}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma}, \boldsymbol{\pi}) = \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \times \prod_{d}^{D} \left[ \gamma_{kd}^{m_{id}} (1 - \gamma_{kd})^{(1 - m_{id})} \right]$$

Normalizing over all component gives the sampling update

$$r_{ik} = \text{Cat}(z_i = k \mid \mathbf{x}_i, \mathbf{m}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\gamma}, \boldsymbol{\pi}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \times \prod_d^D \left[ \gamma_{kd}^{m_{id}} (1 - \gamma_{kd})^{(1 - m_{id})} \right]}{\sum_j^K \left[ \pi_j \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \times \prod_d^D \gamma_{jd}^{m_{id}} (1 - \gamma_{jd})^{(1 - m_{id})} \right]}$$

With full conditionals for all of the variables in the joint distribution, we perform the following sampling steps at each iteration $t$ of the Gibbs sampler:

Sample $\mathbf{X}_h \sim p(\mathbf{X}_h \mid \mathbf{X}_o, \mathbf{z}^{(t-1)}, \{\boldsymbol{\mu}_k^{(t-1)}, \boldsymbol{\Sigma}_k^{(t-1)}\}_k^K)$

Sample $\boldsymbol{\pi}^{(t)} \sim \text{Dir}\left(\alpha_{0,1} + N_1^{(t-1)}, \ldots, \alpha_{0,K} + N_K^{(t-1)}\right)$

Sample $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \sim \text{NIW}\left(\mathbf{m}_k, \kappa_k, \nu_k, \mathbf{S}_k\right), \quad \text{for all } k \in [K]$

Sample $\gamma_{kd}^{(t)} \sim \text{Beta}\left(u_{0,k} + N_{kd}^{(1)},\ v_{0,k} + N_{kd}^{(0)}\right), \quad \text{for all } k \in [K],\ d \in [D]$

Sample $z_i^{(t)} \sim \text{Categorical}\left(p(z_i = k \mid \mathbf{x}_i, \mathbf{m}_i, \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\pi}^{(t)})\right), \quad \text{for all } i \in [N]$

## 9.3 Evaluation

### 9.3.1 Experiment

We evaluate the model on the MP voting dataset and on synthetic datasets generated under MNAR mechanisms for both BMMs and GMMs. For the synthetic studies we use two cluster-dependent MNAR regimes:

1. **Uniform MNAR.** Each component $k$ has a single observation probability $\gamma_k \in [0, 1]$ applied uniformly across features:

   $$m_{id} \mid z_i = k \sim \text{Bernoulli}(\gamma_k) \quad \text{for all } d.$$

   (Missing rate $= 1 - \gamma_k$.) We use evenly-spaced missingness from 0 to 90 percent missingness for each component $k$.

2. **Pattern MNAR.** Each component $k$ has a feature-specific observation vector $\boldsymbol{\gamma}_k = (\gamma_{k1}, \ldots, \gamma_{kD})$:

   $$m_{id} \mid z_i = k \sim \text{Bernoulli}(\gamma_{kd}).$$

   In this approach, we hard-code missingness templates as Bernoulli bias vectors consisting of values of 0.95 and 0.05.

These two settings let us test the MNAR generative model under both a simple (uniform) and a structured (feature-patterned) missingness scenario.

In our testing, we specifically look at clustering performance (ARI) on both training sets and held-out test sets. We additionally test the approach against MCAR data to test if the MNAR generative model can flexibly adapt to MCAR settings as well.

The datasets used are as follows

| **Name** | BMM Synthetic Uniform | BMM Synthetic Pattern | BMM MP Votes | BMM Synthetic MCAR | GMM Synthetic Uniform | GMM Synthetic Pattern | GMM Synthetic MCAR |
|---|---|---|---|---|---|---|---|
| Features | 5 | 5 | 207 | 5 | 3 | 3 | 3 |
| Components | 3 | 3 | 9 | 3 | 3 | 3 | 3 |
| Data points | 2500 | 2500 | 645 | 2500 | 1500 | 1500 | 1500 |

Table 9.1: Summary of Datasets

## 9.4 Results

### 9.4.1 BMM MNAR Results

| | BMM MNAR | BMM MCAR | Δ (MNAR–MCAR) |
|---|---|---|---|
| *Train ARI* | | | |
| BMM Uniform | **0.1989** | 0.1973 | 0.0016 |
| BMM Pattern | **0.3884** | 0.3552 | 0.0332 |
| MP Voting | **0.8623** | 0.7975 | 0.0648 |
| *Test ARI* | | | |
| BMM Uniform | **0.3253** | 0.2009 | 0.1244 |
| BMM Pattern | **0.4126** | 0.3923 | 0.0203 |

Table 9.2: Clustering performance (Adjusted Rand Index) for BMM datasets.

The results show that for the MNAR datasets, the MNAR Gibbs sampling model consistently out performs the standard MCAR variant for both training and held-out test splits. Gains are modest for the uniform missingness on train (+0.0016 ARI) but substantial on test (+0.1244). The pattern MNAR dataset shows consistent improvements (+0.0332 train, +0.0203 test). For the MP voting data, the MNAR model improves train ARI by +0.0648, suggesting that a cluster-dependent missingness may be present.

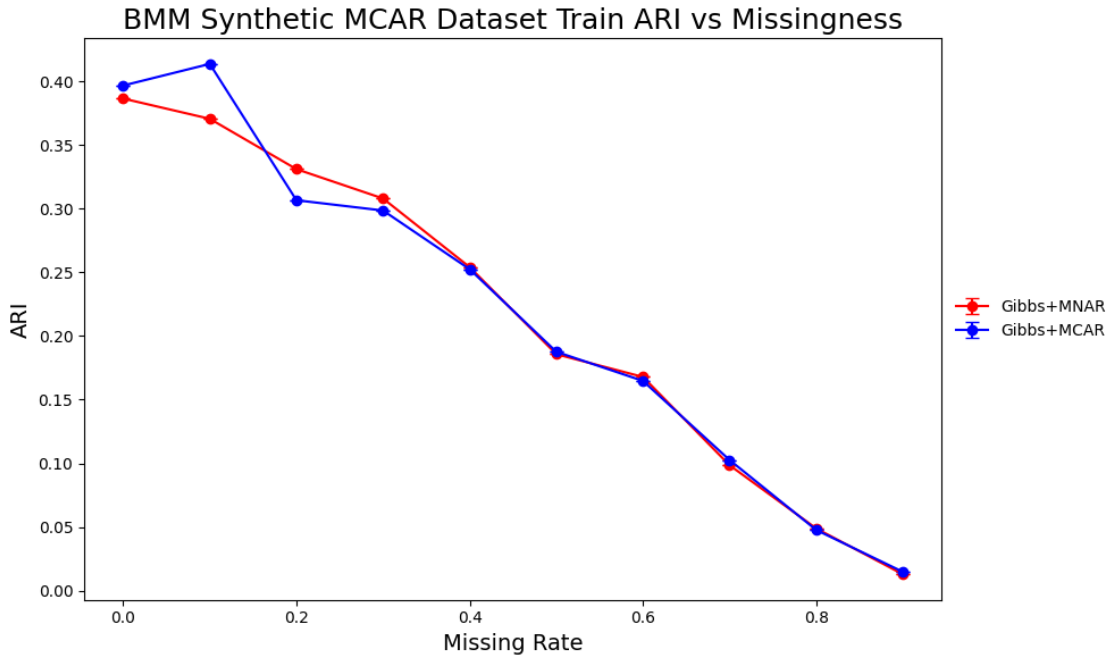



Figure 9.2: Clustering performance of MNAR Gibbs sampling model against standard MCAR Gibbs sampling model in synthetic BMM MCAR dataset. This suggests that for the BMM case, the MNAR model does not degrade in clustering performance for MCAR data

### 9.4.2 GMM MNAR Results

| | GMM MNAR | GMM MCAR | Δ (MNAR–MCAR) |
|---|---|---|---|
| *Train ARI* | | | |
| GMM Uniform | **0.7661** | 0.4490 | 0.3171 |
| GMM Pattern | **0.9652** | 0.4628 | 0.5024 |
| *Test ARI* | | | |
| GMM Uniform | **0.9216** | 0.4959 | 0.4257 |
| GMM Pattern | **1.0** | 0.5490 | 0.4510 |

Table 9.3: Clustering performance (Adjusted Rand Index) for GMM datasets.

The results on the GMM datasets confirm that MNAR Gibbs sampling model offers substantial benefits in clustering performance compared to the standard MCAR model. In the Gaussian case, the differences in ARI are more apparent, particularly for the uniform MNAR missing setting. Again, the pattern-based MNAR setting produces starker improvements in ARI compared to the uniform case.

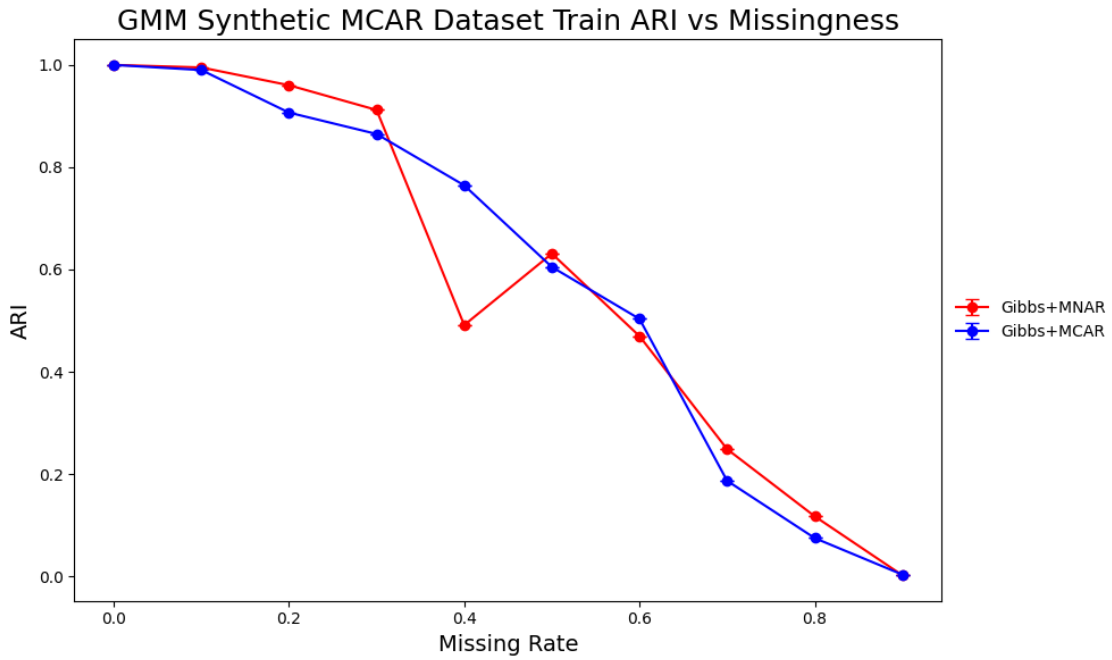



Figure 9.3: Clustering performance of MNAR Gibbs sampling model against standard MCAR Gibbs sampling model in synthetic GMM MCAR dataset. This suggests that for the GMM case, the MNAR model does not degrade in clustering performance for MCAR data

# Chapter 10

# Conclusion

This work investigated fully Bayesian, one–step inference for mixture models in the presence of missing data. Framing clustering through a generative perspective, we treated missing entries as latent variables and performed joint inference over assignments, parameters, and (when applicable) missingness mechanisms. We implemented Gibbs sampling and Variational Bayes EM (VBEM) for Bernoulli and Gaussian mixture models, and benchmarked them against maximum–likelihood and ad-hoc two–step baselines across a range of missingness levels and dimensionalities.

Across all datasets and metrics (clustering (ARI), model fit (log-likelihood), and imputation accuracy (RMSE)) the fully Bayesian approaches consistently outperformed EM, imputation, and complete-case analysis baselines. The advantages were most pronounced as missingness and dimensionality increased, where complete-case methods quickly became unusable and ad-hoc imputations introduced bias and instability. VBEM often edged out Gibbs in higher dimensions under the fixed iteration limit we used (set for time), where higher number of iterations should theoretically produce comparable performance. The close alignment of improvements in ARI, log-likelihood, and RMSE confirms that modeling posterior uncertainty over missing values translates into better clustering and reconstruction.

We explored a restricted MNAR mechanism in which the observation mask depends on the latent component, adding a Bernoulli model for the mask with Beta priors. On synthetic data generated under this assumption, and on the MP voting dataset, the MNAR-aware BMM and GMM Gibbs samplers improved clustering over an otherwise identical MCAR model. Notably, gains were larger when missingness exhibited feature-specific patterns by cluster. Because the MNAR likelihood ideally becomes constant under MCAR, the extension remains compatible when data are actually MCAR. This offers a flexible path that can exploit mask signal when present without harming performance otherwise.

## 10.1 Limitations

- **Prior and hyperparameter sensitivity.** All models used the same, non–fine-tuned (noninformative) priors. While this avoids injecting strong prior bias, mixture models can be sensitive to hyperparameters.
- **Gibbs iteration budget and diagnostics.** The number of Gibbs iterations was

capped for time rather than tuned for convergence, especially in higher dimensions. We relied on sample alignment and empirical MAP summaries where other strategies for sampling can be explored.

- **Missingness mechanisms.** The study focuses on MCAR and a restricted cluster-dependent MNAR model with feature-wise independence. MAR mechanisms and more general MNAR formulations were not explored, and identifiability under richer mechanisms was not assessed.
- **Model selection for $K$.** The number of components was taken as fixed. We did not perform model selection or consider Bayesian nonparametric alternatives, which could affect both clustering quality and imputation accuracy.
- **Dataset and evaluation scope.** Datasets were limited in size, dimensionality, and diversity, particularly for the GMM approaches which are more computationally demanding. We additionally do not consider execution time and resource utilization for each model as to evaluate performance-cost tradeoffs.
- **MNAR Constraints.** VBEM for the MNAR extension and hyperparameter inference were not implemented due to time constraints. We additionally only measure clustering accuracy and do so for a limited selection of datasets.

# Chapter 11

# Appendix

## 11.1 Ethics Form

---

University of St Andrews | FOUNDED 1413

School of Computer Science Ethics Committee

23 June 2025

Dear James Zhang

Thank you for submitting an Ethics Application Form application for review by the Computer Science ethics committee. The committee reviewed this application and accompanying documents on 23 June 2025. The outcome of this review is given below:

**Project Title** Bayesian Unsupervised Learning with Missing Data
**Researcher(s)** James Zhang
**Supervisor(s)** Dr Lei Fang
**Application Ref** 0299 - CS-0299-735-2025
**Decision Date** 23 June 2025 **Decision Expiry Date** 23 June 2030
**Review Outcome** Favourable opinion
**Specific Conditions** *(optional)* None
**Ethics committee comments** *(optional)* None

The following supporting documents are acknowledged:

None

**Favourable opinions**

A favourable opinion is conditional upon any conditions set by the committee, if any, as described in the 'Specific conditions' section above.

A favourable opinion is valid for 5 years from the decision date, see the expiry date above.

If you wish for this opinion to apply to any subsequent changes made to the project, you must first submit an amendment request to the ethics committee, using the University's ethics amendment application form. Changes made to the research without the submission of such a request will invalidate this favourable opinion.

Ethics opinions must be renewed every 5 years by submission of a new application. If only a short extension is required, for example to finish writing up, you can request a discretionary extension of up to 6 months from the ethics committee.

You must report any serious adverse events, or significant changes not covered by this approval, related to this study immediately to the ethics committee.

A favourable opinion is given on the condition that:

- you abide by any specific conditions set by the ethics committee
- you conduct your research in line with:
  - the details provided in your ethical application.
  - relevant University policies and procedures, including the Principles of Good Research Conduct.
  - the conditions of any funding associated with your work.
  - any local legal or ethical requirements.
- all applicable approvals, permissions, or documents are obtained before research commences.

A favourable opinion by a University ethics committee does not confer any kind of approval for the research, be it governance, legal or otherwise. However, a favourable opinion is necessary for the research to proceed.

You should retain this approval letter with your study paperwork.

Yours sincerely,

Computer Science

## 11.2 Train vs. Test Performance Trace-plots for Gibbs & VBEM

### 11.2.1 BMM Synthetic Dataset

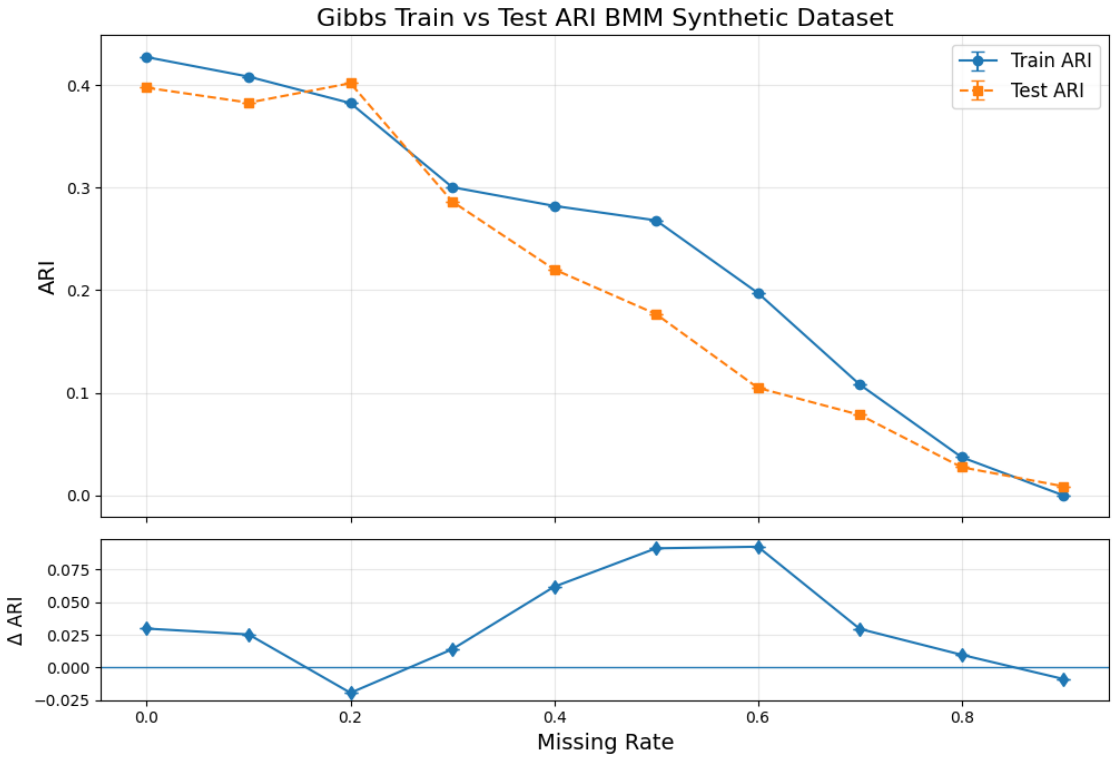



Figure 11.2: Train vs. Test Clustering Performance of Gibbs Sampling for BMM on the BMM Synthetic Dataset

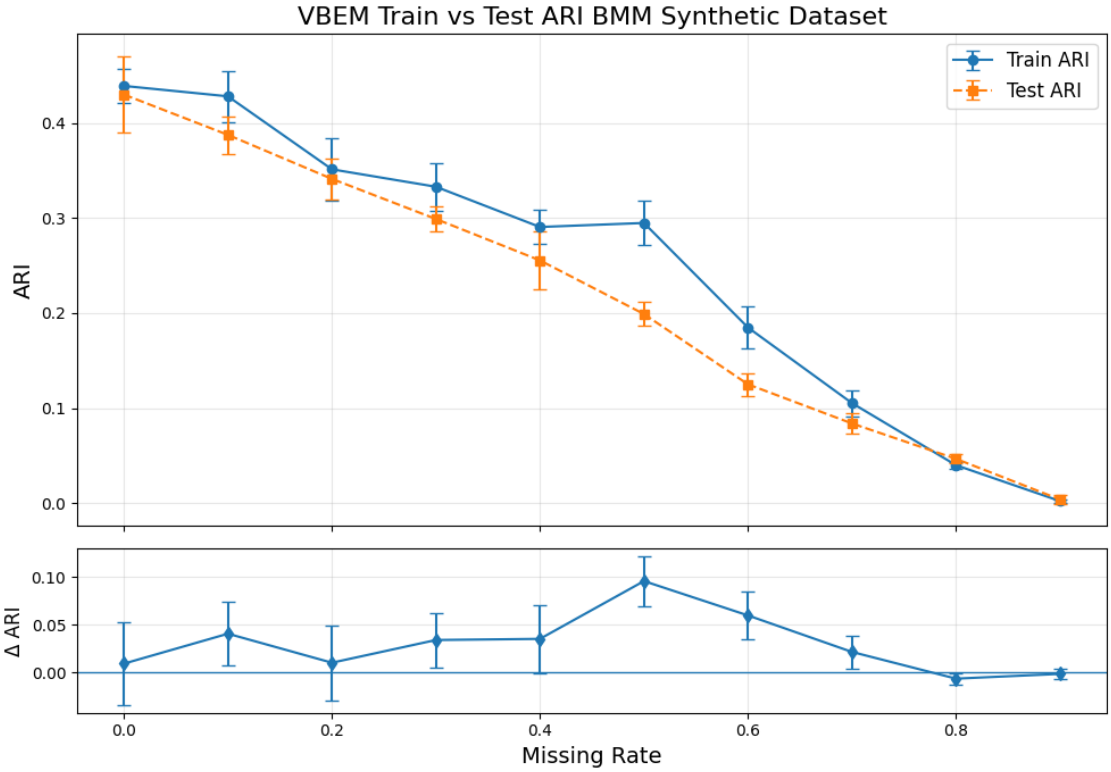



Figure 11.3: Train vs. Test Clustering Performance of VBEM for BMM on the BMM Synthetic Dataset

### 11.2.2 BMM Shapes Dataset

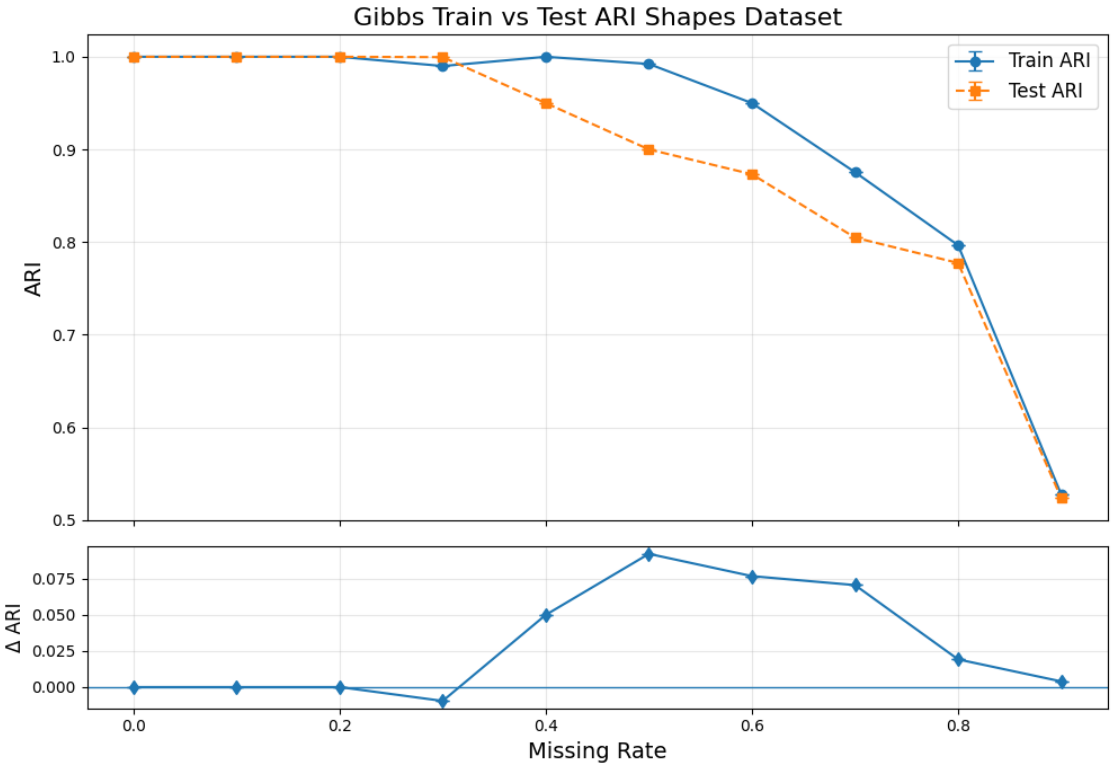



Figure 11.4: Train vs. Test Clustering Performance of Gibbs Sampling for BMM on the BMM Shapes Dataset

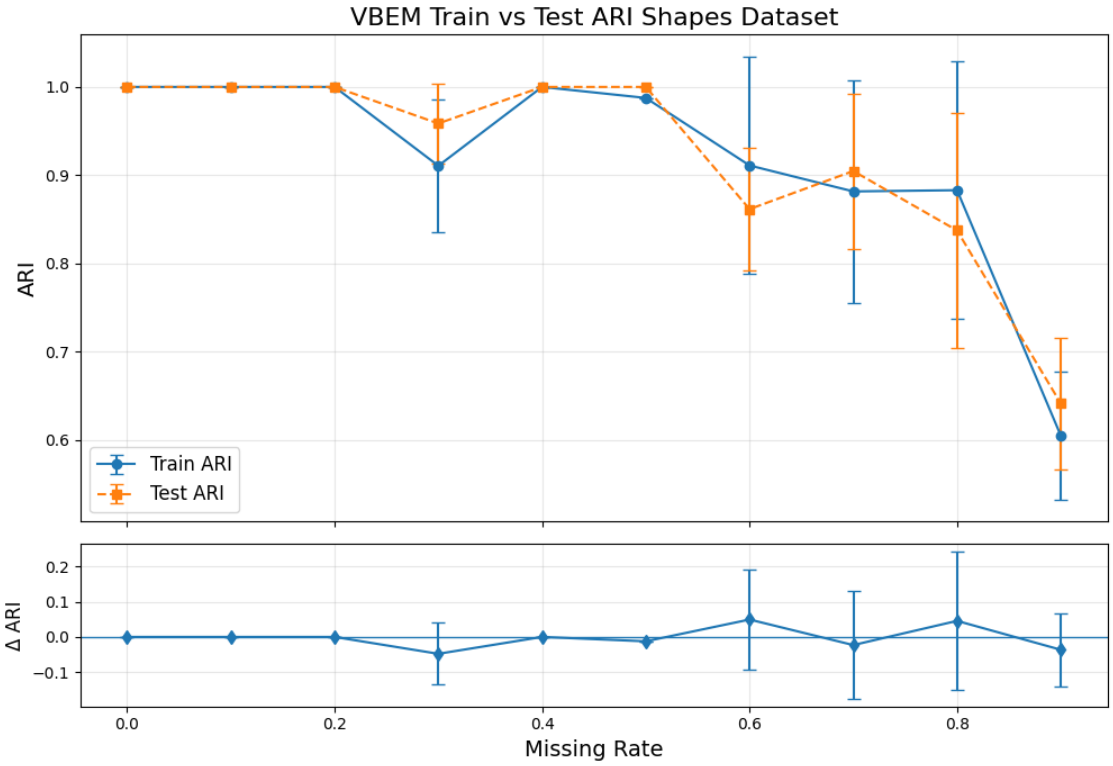



Figure 11.5: Train vs. Test Clustering Performance of VBEM for BMM on the BMM Shapes Dataset

### 11.2.3 GMM Synthetic Dataset

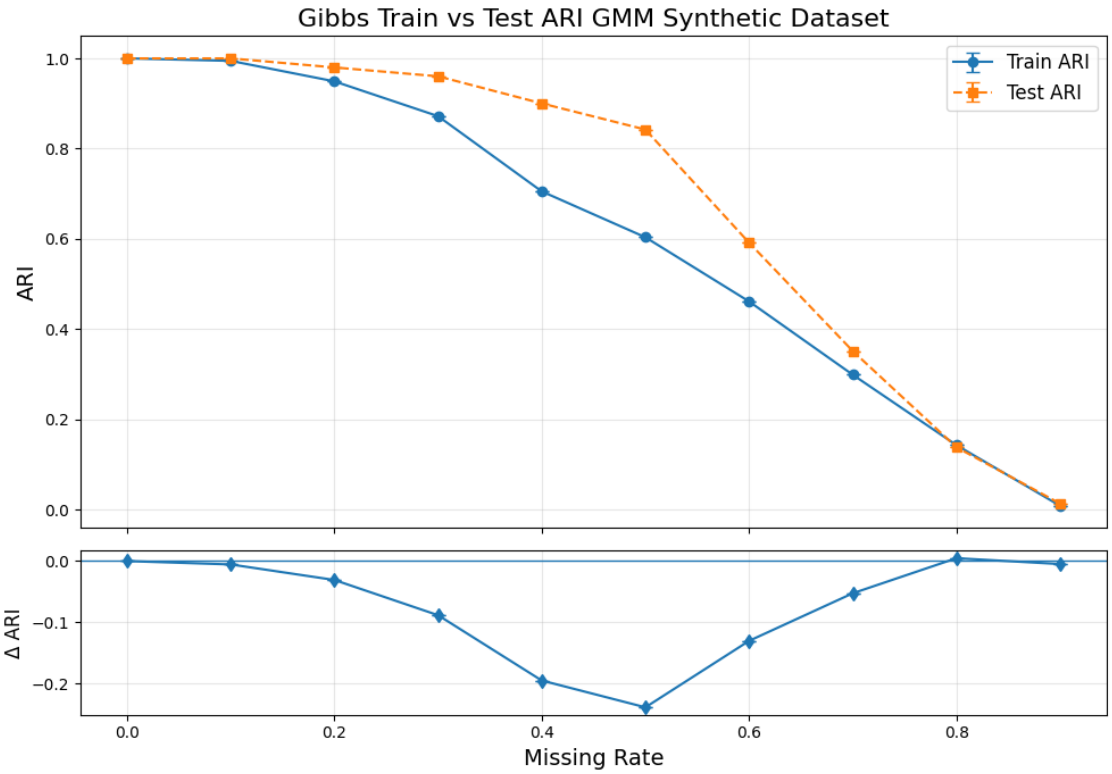



Figure 11.6: Train vs. Test Clustering Performance of Gibbs Sampling for GMM on the GMM Synthetic Dataset

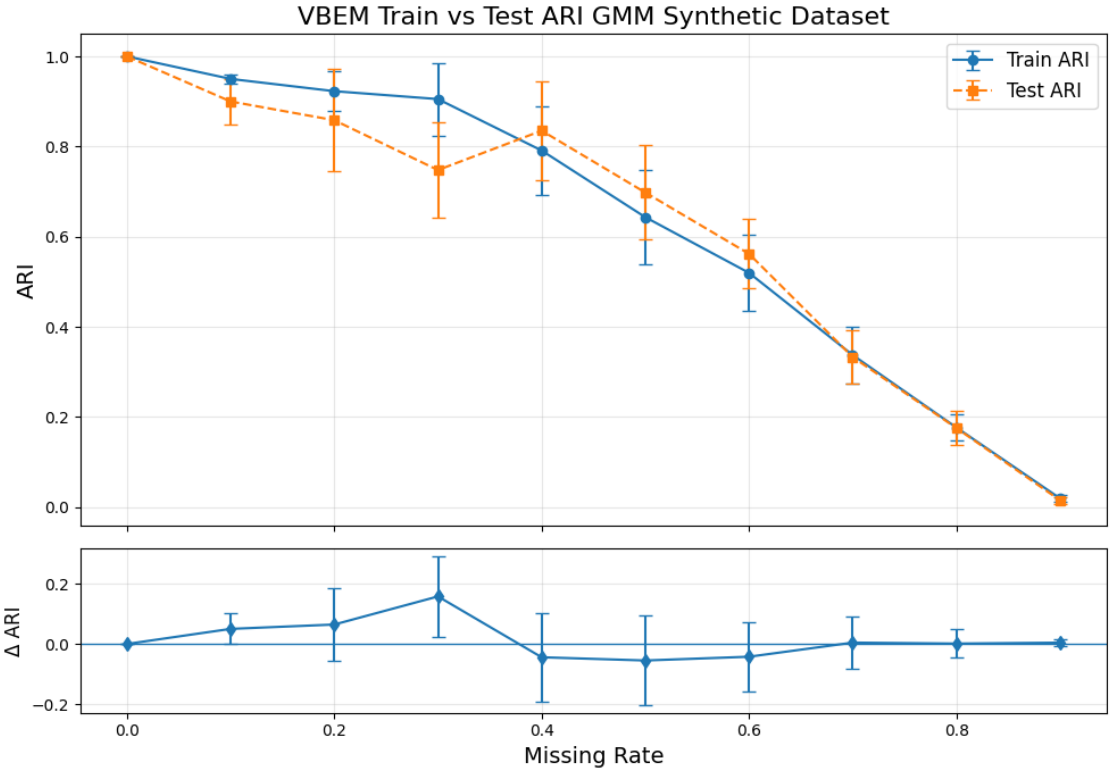



Figure 11.7: Train vs. Test Clustering Performance of VBEM for GMM on the GMM Synthetic Dataset

### 11.2.4 GMM Iris Dataset


Figure 11.8: Train vs. Test Clustering Performance of Gibbs Sampling for GMM on the GMM Iris Dataset


Figure 11.9: Train vs. Test Clustering Performance of VBEM for GMM on the GMM Iris Dataset

### 11.2.5 GMM Digits Dataset

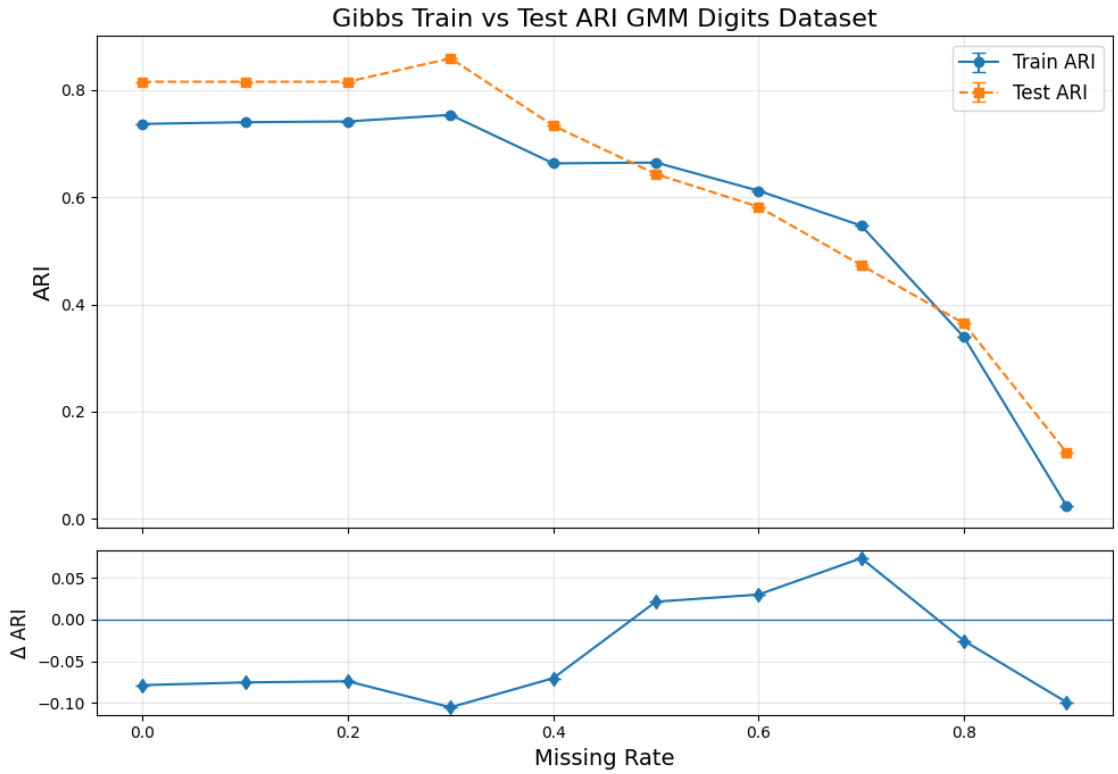



Figure 11.10: Train vs. Test Clustering Performance of Gibbs Sampling for GMM on the GMM Digits Dataset

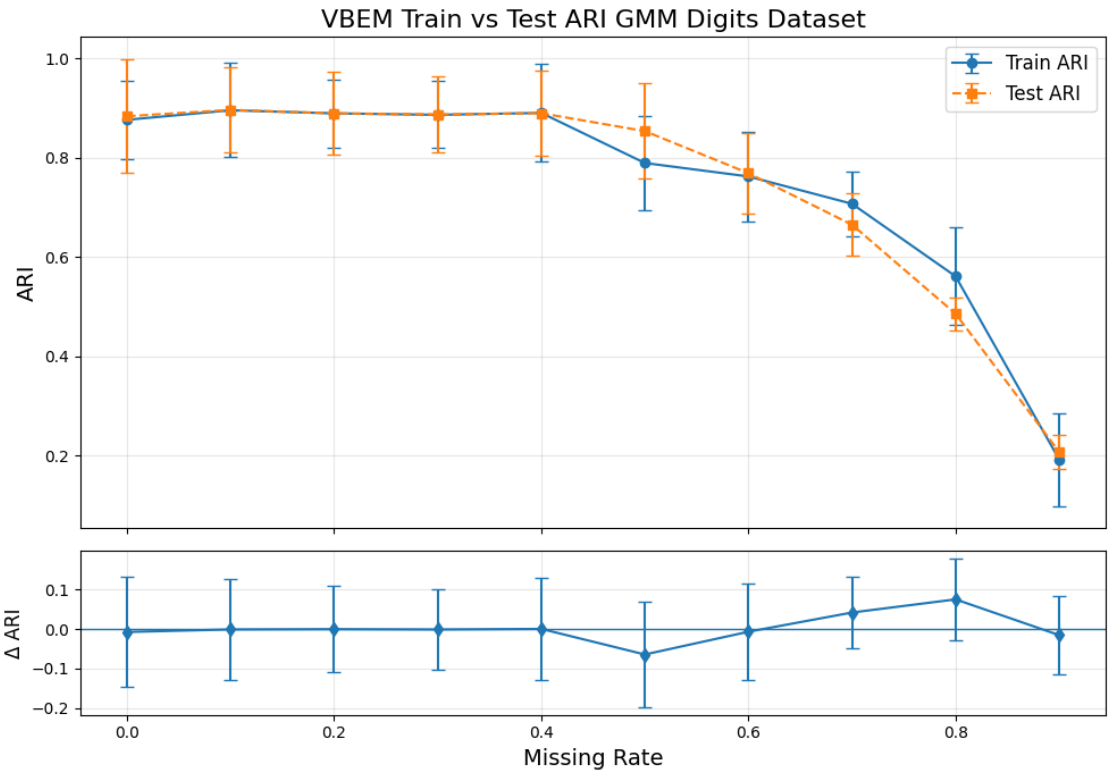



Figure 11.11: Train vs. Test Clustering Performance of VBEM for GMM on the GMM Digits Dataset

## 11.3 Derivations

Note that for the following derivations, we use $n \in \{1, ..., N\}$ rather than $i \in \{1, ..., N\}$ and $\mathbf{X}_H, \mathbf{X}_O$ to denote missing and observed parts of the data $\mathbf{X}$ rather than $\mathbf{X}_h, \mathbf{X}_o$.

### 11.3.1 VI Evidence Lower Bound Derivation

$$\mathcal{J}(q) = \mathbb{KL}(q \,||\, p) \tag{11.1}$$

$$= \int q(\mathbf{Z}) \log \frac{q(\mathbf{Z})}{p(\mathbf{Z} \mid \mathbf{X})} \, d\mathbf{Z} \tag{11.2}$$

$$= \int q(\mathbf{Z}) \log q(\mathbf{Z}) d\mathbf{Z} - \int q(\mathbf{Z}) \log p(\mathbf{Z}, \mathbf{X}) d\mathbf{Z} + \log p(\mathbf{X}) \tag{11.3}$$

$$= \int q(\mathbf{Z}) \log \frac{q(\mathbf{Z})}{p(\mathbf{Z}, \mathbf{X})} d\mathbf{Z} + \log p(\mathbf{X}) \tag{11.4}$$

$$= - \int q(\mathbf{Z}) \log \frac{p(\mathbf{Z}, \mathbf{X})}{q(\mathbf{Z})} d\mathbf{Z} + \log p(\mathbf{X}) \tag{11.5}$$

$$\tag{11.6}$$

Define ELBO function $\mathcal{L}(q)$ as

$$\mathcal{L}(q) = -\int q(\mathbf{Z}) \log \frac{p(\mathbf{Z}, \mathbf{X})}{q(\mathbf{Z})} d\mathbf{Z} \tag{11.7}$$

$$\implies \mathcal{L}(q) = \log p(\mathbf{X}) - \mathcal{J}(q) \tag{11.8}$$

$$= \log p(\mathbf{X}) - \mathbb{KL}(q \,||\, p) \tag{11.9}$$

$$\implies \mathcal{L}(q) \leq \log p(\mathbf{X}) \tag{11.10}$$

### 11.3.2 EM Algorithm Lower Bound Derivation

$$\log p(\mathbf{X}|\boldsymbol{\phi}) = \log \int p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\phi})\, d\mathbf{Z} \tag{11.11}$$

$$= \log \int p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\phi}^{(t-1)}) \frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\phi})}{p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\phi}^{(t-1)})}\, d\mathbf{Z}, \quad \text{Introduce posterior } p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\phi}^{(t-1)}) \tag{11.12}$$

$$\geq \int p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\phi}^{(t-1)}) \log \frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\phi})}{p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\phi}^{(t-1)})}\, d\mathbf{Z}, \quad \text{By Jensen's Equality} \tag{11.13}$$

$$= \mathbb{E}_{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\phi}^{(t-1)})} \Big[ p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\phi}) \Big] \tag{11.14}$$

$$= Q(\boldsymbol{\phi} \mid \boldsymbol{\phi}^{(t-1)}) \tag{11.15}$$

$$\implies \log p(\mathbf{X}|\boldsymbol{\phi}) \geq Q(\boldsymbol{\phi} \mid \boldsymbol{\phi}^{(t-1)}) \tag{11.16}$$

### 11.3.3 Dirichlet Categorical Conjugacy

$$p(\boldsymbol{\pi} \mid \mathbf{z}) \propto p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}_0) \tag{11.17}$$

$$= \left( \prod_i^N \prod_k^K \pi_k^{z_{ik}} \right) \left( \prod_k^K \pi_k^{(\alpha_{0,k}-1)} \right) \tag{11.18}$$

$$= \left( \prod_k^K \pi_k^{N_k} \right) \left( \prod_k^K \pi_k^{(\alpha_{0,k}-1)} \right), \quad \text{where } N_k = \sum_n^N \mathbb{1}(z_i = k) \tag{11.19}$$

$$= \underbrace{\prod_k^K \pi_k^{\alpha_{0,k}+N_k-1}}_{\text{Dirichlet Kernel}} \tag{11.20}$$

$$\implies p(\boldsymbol{\pi} \mid \mathbf{z}) = \text{Dir}(\boldsymbol{\pi} \mid \alpha_{0,1} + N_1, ..., \alpha_{0,k} + N_k) \tag{11.21}$$

### 11.3.4 Beta Bernoulli Conjugacy

$$p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{z}) \propto p(\mathbf{X} \mid \boldsymbol{\theta}, \mathbf{z})\, p(\boldsymbol{\theta} \mid \mathbf{a}_0, \mathbf{b}_0) \tag{11.22}$$

$$= \prod_i^N \prod_k^K \prod_d^D \left[\theta_{kd}^{x_{id}} (1-\theta_{kd})^{(1-x_{id})}\right] \times \prod_k^K \prod_d^D \theta_{kd}^{\mathbf{a}_{0,k}-1} (1-\theta_{k,d})^{b_{0,k}-1} \tag{11.23}$$

$$= \prod_k^K \prod_d^D \left[\theta_{kd}^{N_{kd}^{(1)}} (1-\theta_{kd})^{N_{kd}^{(0)}}\right] \times \prod_k^K \prod_d^D \theta_{kd}^{\mathbf{a}_{0,k}-1} (1-\theta_{k,d})^{b_{0,k}-1}, \quad \text{where} \tag{11.24}$$

$$N_{kd}^{(1)} = \sum_n^N \mathbb{1}(x_{id} = 1, z_i = k) \tag{11.25}$$

$$N_{kd}^{(0)} = \sum_n^N \mathbb{1}(x_{id} = 0, z_i = k) \tag{11.26}$$

$$= \underbrace{\prod_k^K \prod_d^D \theta_{kd}^{a_{0k}+N_{kd}^{(1)}-1} (1-\theta_{kd})^{b_{0k}+N_{kd}^{(0)}-1}}_{\text{Beta Kernel}} \tag{11.27}$$

### 11.3.5 VBEM update for mixing weights $\boldsymbol{\pi}$

$$\log q(\boldsymbol{\pi}) = \log p(\boldsymbol{\pi}) + \Big\langle \log p(\mathbf{z}|\boldsymbol{\pi}) \Big\rangle_{q(\mathbf{z})} + \text{const} \tag{11.28}$$

$$= \log \text{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0) + \Big\langle \sum_n^N \log p(z_n|\boldsymbol{\pi}) \Big\rangle_{q(\mathbf{z})} + \text{const} \tag{11.29}$$

$$= \log \Big[ \frac{1}{B(\alpha_0)} \prod_k^K \pi_k^{\alpha_0 - 1} \Big] + \Big\langle \sum_n^N \sum_k^K \log \pi_k^{z_{nk}} \Big\rangle_{q(\mathbf{z})} + \text{const} \tag{11.30}$$

$$= \underbrace{-\log B(\alpha_0)}_{\text{Independent of } \pi} + \sum_k^K \log \pi_k^{\alpha_0 - 1} + \Big\langle \sum_k^K \sum_n^N z_{nk} \log \pi_k \Big\rangle_{q(\mathbf{z})} + \text{const} \tag{11.31}$$

$$= \sum_k^K (\alpha_0 - 1) \log \pi_k + \sum_k^K \sum_n^N \Big\langle z_{nk} \Big\rangle_{q(\mathbf{z})} \log \pi_k + \text{const} \tag{11.32}$$

$$= \sum_k^K (\alpha_0 - 1) \log \pi_k + \sum_k^K \sum_n^N r_{nk} \log \pi_k + \text{const} \tag{11.33}$$

$$= \sum_k^K \Big[ (\alpha_0 - 1) \log \pi_k + \log \pi_k \sum_n^N r_{nk} \Big] + \text{const} \tag{11.34}$$

$$= \sum_k^K \Big[ (\alpha_0 + N_k - 1) \log \pi_k \Big] + \text{const}, \quad \text{where } N_k = \sum_n^N r_{nk} \tag{11.35}$$

$$\implies q(\boldsymbol{\pi}) = \prod_k^K \pi_k^{\alpha_0 + N_k - 1} + \text{const} = \text{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0 + N), \quad \text{where } N = [N_0, ..., N_K] \tag{11.36}$$

### 11.3.6 VBEM Update for Complete Data Bernoulli mean $\boldsymbol{\theta}$

$$\log q(\boldsymbol{\theta}) = \sum_k^K \log p(\boldsymbol{\theta}_k) + \Big\langle \sum_n^N \sum_k^K z_{nk} \cdot \log p(\mathbf{x}_n|\boldsymbol{\theta}_k) \Big\rangle_{q(\mathbf{z})} + \text{const} \tag{11.37}$$

$$= \sum_k^K \log p(\boldsymbol{\theta}_k) + \sum_n^N \sum_k^K r_{nk} \cdot \log p(\mathbf{x}_n|\boldsymbol{\theta}_k) + \text{const} \tag{11.38}$$

$$\tag{11.39}$$

$$\implies \log q(\boldsymbol{\theta}_k) = \log p(\boldsymbol{\theta}_k) + \sum_n^N r_{nk} \cdot \log p(\mathbf{x}_n|\boldsymbol{\theta}_k) + \text{const} \tag{11.40}$$

$$= \sum_d^D \log \text{Beta}(\theta_{k,d} \mid \mathbf{a}_{0,d}, \mathbf{b}_{0,d}) \tag{11.41}$$

$$+ \sum_n^N r_{nk} \cdot \sum_d^D \log \text{Bern}(x_{n,d} \mid \boldsymbol{\theta}_{k,d}) + \text{const} \tag{11.42}$$

$$\implies \log q(\theta_{k,d}) = \log \text{Beta}(\theta_{k,d} \mid \mathbf{a}_{0,d}, \mathbf{b}_{0,d}) + \sum_n^N r_{nk} \cdot \log \text{Bern}(x_{n,d} \mid \boldsymbol{\theta}_{k,d}) + \text{const} \tag{11.43}$$

$$= (a_{0,d} - 1) \log \theta_{k,d} + (b_{0,d} - 1) \log(1 - \theta_{k,d}) \tag{11.44}$$

$$+ \sum_n^N r_{nk} \cdot \Big( x_{n,d} \log \theta_{k,d} + (1 - x_{n,d}) \log(1 - \mu_{k,d}) \Big) + \text{const} \tag{11.45}$$

$$= \Big( a_{0,d} + \sum_n^N r_{nk} x_{n,d} - 1 \Big) \log \theta_{k,d} + \tag{11.46}$$

$$\Big( b_{0,d} + \sum_n^N r_{nk}(1 - x_{n,d}) - 1 \Big) \log(1 - \theta_{k,d}) + \text{const} \tag{11.47}$$

This is the form of a Beta distribution, hence

$$q(\mu_{k,d}) = \text{Beta}(a_{kd}, b_{kd}), \quad \text{where} \tag{11.48}$$

$$a_{kd} = a_{0,d} + \sum_n^N r_{nk} x_{n,d} \tag{11.49}$$

$$b_{kd} = b_{0,d} + \sum_n^N r_{nk}(1 - x_{n,d}) \tag{11.50}$$

### 11.3.7 VBEM BMM Update $q(z_i = k)$ complete case

$$\log q(\mathbf{z}) = \Big\langle \log p(\mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\pi}) \Big\rangle_{q(\mu,\pi)} + \mathrm{C} \tag{11.51}$$

$$= \Big\langle \log p(\mathbf{X}|\boldsymbol{\phi}) \Big\rangle_{q(\mu,\pi)} + \Big\langle \log p(\mathbf{z}|\boldsymbol{\pi}) \Big\rangle_{q(\mu,\pi)} + \underbrace{\log p(\boldsymbol{\phi}) + \log p(\boldsymbol{\pi})}_{\text{Independent of } \mathbf{z}} + \mathrm{C} \tag{11.52}$$

$$= \Big\langle \log p(\mathbf{X}|\boldsymbol{\phi}) \Big\rangle_{q(\mu,\pi)} + \Big\langle \log p(\mathbf{z}|\boldsymbol{\pi}) \Big\rangle_{q(\mu,\pi)} + \mathrm{C} \tag{11.53}$$

$$= \Big\langle \sum_n^N \sum_k^K \log p(\mathbf{x}_n|\boldsymbol{\phi}_k)^{z_{nk}} \Big\rangle_{q(\mu,\pi)} + \Big\langle \sum_n^N \sum_k^K \log p(z_n = k|\boldsymbol{\pi})^{z_{nk}} \Big\rangle_{q(\mu,\pi)} + \mathrm{C} \tag{11.54}$$

$$= \sum_n^N \sum_k^K \Big\langle z_{nk} \log p(\mathbf{x}_n|\boldsymbol{\phi}_k) \Big\rangle_{q(\mu,\pi)} + \sum_n^N \sum_k^K \Big\langle z_{nk} \log p(z_n = k|\boldsymbol{\pi}) \Big\rangle_{q(\mu,\pi)} + \mathrm{C} \tag{11.55}$$

$$= \sum_n^N \left[ \sum_k^K \Big\langle z_{nk} \log p(\mathbf{x}_n|\boldsymbol{\phi}_k) \Big\rangle_{q(\mu,\pi)} + \sum_k^K \Big\langle z_{nk} \log p(z_n = k|\boldsymbol{\pi}) \Big\rangle_{q(\mu,\pi)} \right] + \mathrm{C} \tag{11.56}$$

Hence

$$\log q(z_n) = \sum_k^K \left[ \Big\langle z_{nk} \log p(\mathbf{x}_n|\boldsymbol{\phi}_k) \Big\rangle_{q(\mu,\pi)} + \Big\langle z_{nk} \log p(z_n = k|\boldsymbol{\pi}) \Big\rangle_{q(\mu,\pi)} \right] + \mathrm{C} \tag{11.57}$$

This gives

$$\log p(z_n = k) = \underbrace{\Big\langle \log p(\mathbf{x}_n|\boldsymbol{\phi}_k) \Big\rangle_{q(\mu,\pi)}}_{(1)} + \underbrace{\Big\langle \log \boldsymbol{\pi}_k \Big\rangle_{q(\mu,\pi)}}_{(2)} + \mathrm{C} \tag{11.58}$$

For mixed-bern $\boldsymbol{\phi} = \{\boldsymbol{\mu}\}$. Hence,

$$(1) \quad \Big\langle \log p(\mathbf{x}_n|\boldsymbol{\phi}_k) \Big\rangle_{q(\mu)} = \Big\langle \log \mathrm{Bern}(\mathbf{x}_n|\boldsymbol{\mu}_k) \Big\rangle_{q(\mu)} \tag{11.59}$$

$$= \Big\langle \sum_d^D \log \mathrm{Bern}(x_{n,d}|\mu_{k,d}) \Big\rangle_{q(\mu)} \tag{11.60}$$

$$= \Big\langle \sum_d^D x_{n,d} \log \mu_{k,d} + (1 - x_{n,d}) \log(1 - \mu_{k,d}) \Big\rangle_{q(\mu)} \tag{11.61}$$

$$= \sum_d^D x_{n,d} \big\langle \log \mu_{k,d} \big\rangle_{q(\mu)} + (1 - x_{n,d}) \big\langle \log(1 - \mu_{k,d}) \big\rangle_{q(\mu)} \tag{11.62}$$

$$= \sum_d^D x_{n,d} \Big( \psi(a_{k,d}) - \psi(a_{k,d} + b_{k,d}) \Big) + (1 - x_{n,d}) \Big( \psi(b_{k,d}) - \psi(a_{k,d} + b_{k,d}) \Big) \tag{11.63}$$

For (2), given that each marginal of a Dirichlet distribution is a Beta distribution with : $\boldsymbol{\pi}_k \sim \text{Beta}(\alpha_k, \sum_{k'} \alpha_{k'} - \alpha_k)$ And that if $X \sim \text{Beta}(a, b)$, then $\mathbb{E}[\log X] = \psi(a) - \psi(a + b)$

$$(2) \quad \mathbb{E}_{q(\boldsymbol{\pi})}\Big[\log \boldsymbol{\pi}_k\Big] = \psi(\alpha_k) - \psi\Big(\sum_{k'} \alpha_{k'}\Big) \tag{11.64}$$

Hence we get the update step :

$$\log q(z_n = k) = \tag{11.65}$$

$$\psi(\alpha_k) - \psi\Big(\sum_{k'} \alpha_{k'}\Big) + \tag{11.66}$$

$$\sum_{d}^{D} \Big[ x_{n,d}\big(\psi(a_{k,d}) - \psi(a_{k,d} + b_{k,d})\big) + (1 - x_{n,d})\big(\psi(b_{k,d}) - \psi(a_{k,d} + b_{k,d})\big) \Big] + \text{C} \tag{11.67}$$

### 11.3.8 VBEM BMM Update for $q(\boldsymbol{\theta})$ Missing Data Case

$$\ln q(\boldsymbol{\mu}_k) = \ln p(\boldsymbol{\mu}_k) + \sum_n^N \Big\langle z_{nk} \cdot \ln p(\mathbf{x}_O^n, \mathbf{x}_H^n | \boldsymbol{\mu}_k) \Big\rangle_{q(z_n, \mathbf{x}_H^n)} + \text{C} \tag{11.68}$$

$$= \ln p(\boldsymbol{\mu}_k) + \sum_n^N \Big\langle z_{nk} \Big\rangle_{q(z_n)} \cdot \Big\langle \ln p(\mathbf{x}_O^n, \mathbf{x}_H^n | \boldsymbol{\mu}_k) \Big\rangle_{q(\mathbf{x}_H^n | z_n)} + \text{C} \tag{11.69}$$

$$= \ln p(\boldsymbol{\mu}_k) + \sum_n^N r_{nk} \cdot \Big\langle \ln p(\mathbf{x}_O^n, \mathbf{x}_H^n | \boldsymbol{\mu}_k) \Big\rangle_{q(\mathbf{x}_H^n | z_n)} + \text{C} \tag{11.70}$$

$$= \ln \text{Beta}(\boldsymbol{\mu}_k | \mathbf{a}_0, \mathbf{b}_0) + \sum_n^N r_{nk} \cdot \Big\langle \ln \text{Bern}(\mathbf{x}_O^n, \mathbf{x}_H^n | \boldsymbol{\mu}_k) \Big\rangle_{q(\mathbf{x}_H^n | z_n)} + \text{C} \tag{11.71}$$

$$= \sum_d^D (a_{0,d} - 1) \ln \mu_{k,d} + (b_{0,d} - 1) \ln(1 - \mu_{k,d}) + \tag{11.72}$$

$$\sum_n^N r_{nk} \cdot \Big\langle [x_O^n, x_H^n] \ln \mu_k + (1 - [x_O^n, x_H^n]) \ln(1 - \mu_k) \Big\rangle_{q(\mathbf{x}_H^n | z_n)} + \text{C} \tag{11.73}$$

$$= \sum_d^D (a_{0,d} - 1) \ln \mu_{k,d} + (b_{0,d} - 1) \ln(1 - \mu_{k,d}) + \tag{11.74}$$

$$\sum_d^D \sum_n^N r_{nk} \cdot \Big\langle x_d^n \ln \mu_{k,d} + (1 - x_d^n) \ln(1 - \mu_{k,d}) \Big\rangle_{q(\mathbf{x}_H^n | z_n)} + \text{C} \tag{11.75}$$

$$= \sum_d^D (a_{0,d} - 1) \ln \mu_{k,d} + (b_{0,d} - 1) \ln(1 - \mu_{k,d}) + \tag{11.76}$$

$$\sum_d^D \sum_n^N r_{nk} \cdot \Big[ \big\langle x_d^n \big\rangle_k \ln \mu_{k,d} + (1 - \big\langle x_d^n \big\rangle_k) \ln(1 - \mu_{k,d}) \Big] + \text{C} \tag{11.77}$$

$$\implies \ln q(\mu_{k,d}) = (a_{0,d} - 1) \ln \mu_{k,d} + (b_{0,d} - 1) \ln(1 - \mu_{k,d}) + \tag{11.78}$$

$$\sum_n^N r_{nk} \big\langle x_d^n \big\rangle_k \ln \mu_{k,d} + \sum_n^N r_{nk} (1 - \big\langle x_d^n \big\rangle_k) \ln(1 - \mu_k) + \text{C} \tag{11.79}$$

$$= \Big( a_{0,d} + \sum_n^N r_{nk} \big\langle x_d^n \big\rangle_k - 1 \Big) \ln \mu_{k,d} + \tag{11.80}$$

$$\Big( b_{0,d} + \sum_n^N r_{nk} (1 - \big\langle x_d^n \big\rangle_k) - 1 \Big) \ln(1 - \mu_{k,d}) + \text{C} \tag{11.81}$$

This is Beta form. Hence

$$q(\mu_{k,d}) = \mathcal{B}\rceil\sqcup\dashv(\mu_{k,d}|a_{k,d}, b_{k,d}) \tag{11.82}$$

$$a_{k,d} = a_{0,d} + \sum_n^N r_{nk}\left\langle x_d^n \right\rangle_k \tag{11.83}$$

$$b_{k,d} = b_{0,d} + \sum_n^N r_{nk}(1 - \left\langle x_d^n \right\rangle_k) \tag{11.84}$$

Where for $d \in H$

$$\langle x_d \rangle_{q(\mathbf{x}_H^n|z_n=k)} = \frac{\exp\Big(\langle \ln \mu_{k,d} \rangle\Big)}{\exp\Big(\langle \ln \mu_{k,d} \rangle\Big) + \exp\Big(\langle \ln(1-\mu_{k,d}) \rangle\Big)} \tag{11.85}$$

$$= \frac{\exp\Big(\psi(a_{k,d}) - \psi(a_{k,d}+b_{k,d})\Big)}{\exp\Big(\psi(a_{k,d}) - \psi(a_{k,d}+b_{k,d})\Big) + \exp\Big(\psi(b_{k,d}) - \psi(a_{k,d}+b_{k,d})\Big)} \tag{11.86}$$

### 11.3.9 VBEM BMM Update for $q(\mathbf{z}, \mathbf{X}_H)$

Noting that

$$q(\mathbf{z}, \mathbf{X}_H) = \underbrace{q(\mathbf{z})}_{(1)} \underbrace{q(\mathbf{X}_H|\mathbf{z})}_{(2)} \tag{11.87}$$

We have

$$(2) \quad \ln q(\mathbf{X}_H|\mathbf{z}) = \ln q(\mathbf{X}_H, \mathbf{z}) + \mathrm{C} \tag{11.88}$$

$$= \Big\langle \ln p(\mathbf{X}_O, \mathbf{X}_H, \boldsymbol{\mu}, \boldsymbol{\pi}) \Big\rangle_{q(\mu,\pi)} + \mathrm{C} \tag{11.89}$$

$$= \Big\langle \ln p(\mathbf{X}_O, \mathbf{X}_H|\boldsymbol{\mu}, \mathbf{z}) \Big\rangle_{q(\mu)} + \underbrace{\Big\langle \ln p(\mathbf{z}|\boldsymbol{\pi}) \Big\rangle_{q(\pi)} + \ln p(\boldsymbol{\mu}) + \ln p(\boldsymbol{\pi})}_{\text{Independent of } \mathbf{X}_H} + \mathrm{C} \tag{11.90}$$

$$= \Big\langle \ln p(\mathbf{X}_O, \mathbf{X}_H|\boldsymbol{\mu}, \mathbf{z}) \Big\rangle_{q(\mu)} + \mathrm{C} \tag{11.91}$$

$$= \Big\langle \sum_n^N \sum_k^K z_{nk} \cdot \ln p(\mathbf{x}_O^n, \mathbf{x}_H^n|\boldsymbol{\mu}_k) \Big\rangle_{q(\mu)} + \mathrm{const} \tag{11.92}$$

$$= \sum_n^N \Big\langle \sum_k^K z_{nk} \ln p(\mathbf{x}_O^n|\boldsymbol{\mu}_k) + z_{nk} \ln p(\mathbf{x}_H^n|\mathbf{x}_O^n, \boldsymbol{\mu}_k) \Big\rangle_{q(\mu)} + \mathrm{C} \tag{11.93}$$

$$\implies \ln q(\mathbf{x}_H^n|z_n) = \left\langle \sum_k^K z_{nk} \ln p(\mathbf{x}_O^n|\boldsymbol{\mu}_k) + z_{nk} \ln p(\mathbf{x}_H^n|\mathbf{x}_O^n, \boldsymbol{\mu}_k) \right\rangle_{q(\mu)} + \text{C} \tag{11.94}$$

$$= \underbrace{\left\langle \sum_k^K z_{nk} \ln p(\mathbf{x}_O^n|\boldsymbol{\mu}_k) \right\rangle_{q(\mu)}}_{\text{Independent of } \mathbf{x}_H^n} + \left\langle \sum_k^K z_{nk} \ln p(\mathbf{x}_H^n|\mathbf{x}_O^n, \boldsymbol{\mu}_k) \right\rangle_{q(\mu)} + \text{C} \tag{11.95}$$

$$= \sum_k^K \left\langle z_{nk} \ln p(\mathbf{x}_H^n|\mathbf{x}_O^n, \boldsymbol{\mu}_k) \right\rangle_{q(\mu)} + \text{C} \tag{11.96}$$

$$\implies \ln q(\mathbf{x}_H^n|z_n = k) = \left\langle \ln p(\mathbf{x}_H^n|\mathbf{x}_O^n, \boldsymbol{\mu}_k) \right\rangle_{q(\mu)} + \text{C} \tag{11.97}$$

$$= \left\langle \ln \text{Bern}(\mathbf{x}_H^n|\mathbf{x}_O^n, \boldsymbol{\mu}_k) \right\rangle_{q(\mu)} + \text{C} = \left\langle \ln \text{Bern}(\mathbf{x}_H^n|\boldsymbol{\mu}_k) \right\rangle_{q(\mu)} + \text{C} \tag{11.98}$$

$$= \sum_{d \in H} \left[ x_d^n \Big\langle \ln \mu_{k,d} \Big\rangle_{q(\mu)} + (1 - x_d^n) \Big\langle \ln(1 - \mu_{k,d}) \Big\rangle_{q(\mu)} \right] + \text{C} \tag{11.99}$$

$$\implies \ln q(x_d^n|z_n = k) = x_d^n \Big\langle \ln \mu_{k,d} \Big\rangle_{q(\mu)} + (1 - x_d^n) \Big\langle \ln(1 - \mu_{k,d}) \Big\rangle_{q(\mu)} + \text{C} \tag{11.100}$$

$$= x_d^n \Big( \psi(a_{k,d}) - \psi(a_{k,d} + b_{k,d}) \Big) + (1 - x_d^n) \Big( \psi(b_{k,d}) - \psi(a_{k,d} + b_{k,d}) \Big) \tag{11.101}$$

And

$$(1) \quad \ln q(z_n) = \ln q(z_n, \mathbf{x}_H^n) - \ln q(\mathbf{x}_H^n) + \text{C} \tag{11.102}$$

$$= \Big\langle \sum_k^K z_{nk} \ln \boldsymbol{\pi}_k + \sum_k^K z_{nk} \ln p(\mathbf{x}_O^n, \mathbf{x}_H^n|\boldsymbol{\mu}_k) \Big\rangle_{q(\mu,\pi)} - \Big\langle \sum_k^K z_{nk} \ln p(\mathbf{x}_H^n|\mathbf{x}_O^n, \boldsymbol{\mu}_k) \Big\rangle_{q(\mu)} + \text{C} \tag{11.103}$$

$$= \left\langle \sum_k^K z_{nk} \ln \boldsymbol{\pi}_k + \sum_k^K z_{nk} \ln p(\mathbf{x}_O^n|\boldsymbol{\mu}_k) + \sum_k^K z_{nk} \ln p(\mathbf{x}_H^n|\mathbf{x}_O^n, \boldsymbol{\mu}_k) \right\rangle_{q(\mu,\pi)} - \left\langle \sum_k^K z_{nk} \ln p(\mathbf{x}_H^n|\mathbf{x}_O^n, \boldsymbol{\mu}_k) \right\rangle_{q(\mu)} + \text{C} \tag{11.104}$$

$$= \left\langle \sum_k^K z_{nk} \ln \boldsymbol{\pi}_k + \sum_k^K z_{nk} \ln p(\mathbf{x}_O^n|\boldsymbol{\mu}_k) \right\rangle_{q(\mu,\pi)} + \text{C} \tag{11.105}$$

$$= \sum_k^K \left[ \Big\langle z_{nk} \ln \boldsymbol{\pi}_k \Big\rangle_{q(\pi)} + \Big\langle z_{nk} \ln p(\mathbf{x}_O^n|\boldsymbol{\mu}_k) \Big\rangle_{q(\mu)} \right] + \text{C} \tag{11.106}$$

$$\implies \ln q(z_n = k) = \Big\langle \ln \boldsymbol{\pi}_k \Big\rangle_{q(\pi)} + \Big\langle \ln p(\mathbf{x}_O^n | \boldsymbol{\mu}_k) \Big\rangle_{q(\mu)} + \mathrm{C} \tag{11.107}$$

$$= \psi(\alpha_k) - \psi\Big(\sum_{k'} \alpha_{k'}\Big) + \Big\langle \sum_{d \in O} x_d^n \ln(1 - \mu_{k,d}) + (1 - x_d^n) \ln(1 - \mu_{k,d}) \Big\rangle_{q(\mu)} + \mathrm{C} \tag{11.108}$$

$$= \psi(\alpha_k) - \psi\Big(\sum_{k'} \alpha_{k'}\Big) + \sum_{d \in O} x_d^n \Big\langle \ln \mu_{k,d} \Big\rangle + (1 - x_d^n) \Big\langle \ln(1 - \mu_{k,d}) \Big\rangle + \mathrm{C} \tag{11.109}$$

$$= \psi(\alpha_k) - \psi\Big(\sum_{k'} \alpha_{k'}\Big) \tag{11.110}$$

$$+ \sum_{d \in O} x_d^n \Big(\psi(a_{k,d}) - \psi(a_{k,d} + b_{k,d})\Big) + (1 - x_d^n) \Big(\psi(b_{k,d}) - \psi(a_{k,d} + b_{k,d}\Big)) + \mathrm{C}$$

### 11.3.10 VBEM GMM Update for $\boldsymbol{\mu}_k$ with complete data

$$\log q(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{11.111}$$

$$= \underbrace{\log p(\boldsymbol{\Sigma}_k) + \log p(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k)}_{\text{NIW}} + \sum_n^N r_{nk} \log \underbrace{p(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{\text{MVN}} + \text{C} \tag{11.112}$$

$$= \log \text{NIW}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k|\mathbf{m}_{0,k}, \kappa_{0,k}, \nu_{0,k}, \mathbf{S}_{0,k}) + \sum_n^N r_{nk} \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \text{C} \tag{11.113}$$

$$= \log \Big[\mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_{0,k}, \frac{1}{\kappa_{0,k}}\boldsymbol{\Sigma}_k) \cdot \mathcal{IW}(\boldsymbol{\Sigma}_k|\nu_{0,k}, \mathbf{S}_{0,k})\Big] + \sum_n^N r_{nk} \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \text{C} \tag{11.114}$$

$$= \log \underbrace{\big(|\boldsymbol{\Sigma}_k|^{-\frac{\nu_0+D+1}{2}}\big) + \Big( -\frac{\kappa_0}{2}(\boldsymbol{\mu}_k - \mathbf{m}_{0,k})^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu}_k - \mathbf{m}_{0,k}) - \frac{1}{2}\text{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{S}_{0,k})\Big)}_{\text{Terms dependent on } \mu_k, \Sigma_k} \tag{11.115}$$

$$+ \sum_n^N r_{nk}\big( \underbrace{-\frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)}_{\text{Terms dependent on } \mu_k, \Sigma_k}\big) + \text{C} \tag{11.116}$$

$$= \log \big(|\boldsymbol{\Sigma}_k|^{-\frac{\nu_0+D+1}{2}}\big) + \Big( -\frac{\kappa_0}{2}(\boldsymbol{\mu}_k - \mathbf{m}_{0,k})^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu}_k - \mathbf{m}_{0,k}) - \frac{1}{2}\text{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{S}_{0,k})\Big) \tag{11.117}$$

$$+ \sum_n^N r_{nk}\big( -\frac{1}{2}\log|\boldsymbol{\Sigma}_k|\big) + \sum_n^N r_{nk}\big( -\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)\big) + \text{C} \tag{11.118}$$

$$= \log \big(|\boldsymbol{\Sigma}_k|^{-\frac{\nu_0+D+1}{2}}\big) + \Big( -\frac{\kappa_0}{2}(\boldsymbol{\mu}_k - \mathbf{m}_{0,k})^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu}_k - \mathbf{m}_{0,k}) - \frac{1}{2}\text{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{S}_{0,k})\Big) \tag{11.119}$$

$$+ \sum_n^N -\frac{r_{nk}}{2}\log|\boldsymbol{\Sigma}_k| + \sum_n^N r_{nk}\big( -\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)\big) + \text{C} \tag{11.120}$$

$$= \Big(-\frac{\nu_0 + D + 1}{2}\Big)\log\big|\boldsymbol{\Sigma}_k\big| + \Big( -\frac{\kappa_0}{2}(\boldsymbol{\mu}_k - \mathbf{m}_{0,k})^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu}_k - \mathbf{m}_{0,k}) - \frac{1}{2}\text{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{S}_{0,k})\Big) \tag{11.121}$$

$$+ \log|\boldsymbol{\Sigma}_k| \sum_n^N -\frac{r_{nk}}{2} + \sum_n^N r_{nk}\big( -\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)\big) + \text{C} \tag{11.122}$$

$$(\kappa_0+N_k)\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k-2\Big(\kappa_0\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k}+N_k\bar{\mathbf{x}}_k^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\Big)+\kappa_0\mathbf{m}_{0,k}^T\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k}+N_k\bar{\mathbf{x}}_k^T\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k\Bigg]+$$

C

$$= -\tfrac{1}{2}\Bigg[(\nu_0+N_k+D+1)\log\big|\boldsymbol{\Sigma}_k\big|+\mathrm{tr}\Big(\boldsymbol{\Sigma}_k^{-1}(\mathbf{S}_{0,k}+\mathbf{S}_{\bar{x}_k})\Big)+$$

$$\boldsymbol{\mu}_k^T\underbrace{\Big((\kappa_0+N_k)\boldsymbol{\Sigma}_k^{-1}\Big)}_{\mathbf{M}}\boldsymbol{\mu}_k-2\boldsymbol{\mu}_k^T\underbrace{\Big(\kappa_0\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k}+N_k\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k\Big)}_{\mathbf{b}}+\kappa_0\mathbf{m}_{0,k}^T\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k}+N_k\bar{\mathbf{x}}_k^T\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k\Bigg]+\mathrm{C}$$

$$= -\frac{1}{2}\Bigg[(\nu_0+N_k+D+1)\log\big|\boldsymbol{\Sigma}_k\big|+\mathrm{tr}\Big(\boldsymbol{\Sigma}_k^{-1}(\mathbf{S}_{0,k}+\mathbf{S}_{\bar{x}_k})\Big)+ \tag{11.123}$$

$$\underline{(\boldsymbol{\mu}_k-\mathbf{M}^{-1}\mathbf{b})^T\mathbf{M}(\boldsymbol{\mu}_k-\mathbf{M}^{-1}\mathbf{b})}-\mathbf{b}^T\mathbf{M}^{-1}\mathbf{b}+\kappa_0\mathbf{m}_{0,k}^T\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k}+N_k\bar{\mathbf{x}}_k^T\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k\Bigg]+\mathrm{C}$$

complete the square where,

$$\mathbf{M}=(\kappa_0+N_k)\boldsymbol{\Sigma}_k^{-1}$$

$$\mathbf{b}=(\kappa_0\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k}+N_k\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k)$$

$$= -\frac{1}{2}\Bigg[(\cdots)-(\kappa_0\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k}+N_k\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k)^T\left(\frac{1}{\kappa_0+N_k}\boldsymbol{\Sigma}_k\right)(\kappa_0\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k}+N_k\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k)+$$

$$\kappa_0\mathbf{m}_{0,k}^T\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k}+N_k\bar{\mathbf{x}}_k^T\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k\Bigg]+\mathrm{C} \tag{11.124}$$

$$= -\frac{1}{2}\Bigg[(\cdots)-\left(\frac{1}{\kappa_0+N_k}\right)\Big[\kappa_0^2\mathbf{m}_{0,k}^T\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k}+\kappa_0N_k(\kappa_0\mathbf{m}_{0,k}^T\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k+\kappa_0\bar{\mathbf{x}}_k^T\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k})+N_k^2\bar{\mathbf{x}}_k^T\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k\Big]+$$

$$\kappa_0\mathbf{m}_{0,k}^T\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k}+N_k\bar{\mathbf{x}}_k^T\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k\Bigg]+\mathrm{C}$$

$$= -\frac{1}{2}\Bigg[(\cdots)-\frac{\kappa_0^2}{\kappa_0+N_k}\mathbf{m}_{0,k}^T\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k}+-\frac{k_0N_k}{\kappa_0+N_k}(\kappa_0\mathbf{m}_{0,k}^T\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k+\kappa_0\bar{\mathbf{x}}_k^T\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k})- \tag{11.125}$$

$$\frac{N_k^2}{\kappa_0+N_k}\bar{\mathbf{x}}_k^T\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k+\kappa_0\mathbf{m}_{0,k}^T\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k}+N_k\bar{\mathbf{x}}_k^T\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k\Bigg]+\mathrm{C}$$

$$= -\frac{1}{2}\Bigg[(\cdots)\frac{k_0N_k}{\kappa_0+N_k}\mathbf{m}_{0,k}^T\boldsymbol{\Sigma}_k^{-1}\mathbf{m}_{0,k}-\frac{k_0N_k}{\kappa_0+N_k}(2\kappa_0\mathbf{m}_{0,k}^T\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k)+\frac{k_0N_k}{\kappa_0+N_k}\bar{\mathbf{x}}_k^T\boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k\Bigg]+\mathrm{C} \tag{11.126}$$

$$= -\frac{1}{2}\Bigg[(\cdots) + \frac{k_0 N_k}{\kappa_0 + N_k}\Big((\mathbf{m}_{0,k} - \bar{\mathbf{x}}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{m}_{0,k} - \bar{\mathbf{x}}_k)\Big)\Bigg] + \mathrm{C} \tag{11.127}$$

$$= -\frac{1}{2}\Bigg[(\nu_0 + N_k + D + 1)\log\big|\boldsymbol{\Sigma}_k\big| + \mathrm{tr}\Big(\boldsymbol{\Sigma}_k^{-1}(\mathbf{S}_{0,k} + \mathbf{S}_{\bar{x}_k})\Big) + \tag{11.128}$$

$$\mathrm{tr}\Big(\frac{k_0 N_k}{\kappa_0 + N_k}\boldsymbol{\Sigma}_k^{-1}(\mathbf{m}_{0,k} - \bar{\mathbf{x}}_k)(\mathbf{m}_{0,k} - \bar{\mathbf{x}}_k)^T\Big)\Bigg] + \mathrm{C}$$

$$= -\frac{1}{2}\Bigg[(\nu_0 + N_k + D + 1)\log\big|\boldsymbol{\Sigma}_k\big| + \tag{11.129}$$

$$\mathrm{tr}\Big(\boldsymbol{\Sigma}_k^{-1}(\mathbf{S}_{0,k} + \mathbf{S}_{\bar{x}_k}) + \frac{k_0 N_k}{\kappa_0 + N_k}\boldsymbol{\Sigma}_k^{-1}(\mathbf{m}_{0,k} - \bar{\mathbf{x}}_k)(\mathbf{m}_{0,k} - \bar{\mathbf{x}}_k)^T\Big) +$$

$$(\boldsymbol{\mu}_k - \mathbf{M}^{-1}\mathbf{b})^T\mathbf{M}(\boldsymbol{\mu}_k - \mathbf{M}^{-1}\mathbf{b})\Bigg] + \mathrm{C}$$

$$= -\frac{1}{2}\Bigg[(\nu_0 + N_k + D + 1)\log\big|\boldsymbol{\Sigma}_k\big| +$$

$$\mathrm{tr}\Big(\boldsymbol{\Sigma}_k^{-1}(\mathbf{S}_{0,k} + \mathbf{S}_{\bar{x}_k} + \frac{k_0 N_k}{\kappa_0 + N_k})(\mathbf{m}_{0,k} - \bar{\mathbf{x}}_k)(\mathbf{m}_{0,k} - \bar{\mathbf{x}}_k)^T\Big) +$$

$$(\boldsymbol{\mu}_k - \mathbf{M}^{-1}\mathbf{b})^T\mathbf{M}(\boldsymbol{\mu}_k - \mathbf{M}^{-1}\mathbf{b})\Bigg] + \mathrm{C}$$

$$= -\frac{1}{2}\Bigg[(\nu_0 + N_k + D + 1)\log\big|\boldsymbol{\Sigma}_k\big| + \tag{11.130}$$

$$\mathrm{tr}\Big(\boldsymbol{\Sigma}_k^{-1}(\mathbf{S}_{0,k} + \mathbf{S}_{\bar{x}_k} + \frac{k_0 N_k}{\kappa_0 + N_k}(\mathbf{m}_{0,k} - \bar{\mathbf{x}}_k)(\mathbf{m}_{0,k} - \bar{\mathbf{x}}_k)^T)\Big) +$$

$$(\boldsymbol{\mu}_k - \mathbf{M}^{-1}\mathbf{b})^T\underline{((\kappa_0 + N_k)\boldsymbol{\Sigma}_k^{-1})}(\boldsymbol{\mu}_k - \mathbf{M}^{-1}\mathbf{b})\Bigg] + \mathrm{const}$$

$$\implies q(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \tag{11.131}$$

$$|\boldsymbol{\Sigma}_k|^{-\frac{\nu + N_k + D + 1}{2}} \exp\Big(-\frac{\kappa_0 + N_k}{2}(\boldsymbol{\mu}_k - \mathbf{M}^{-1}\mathbf{b})^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu}_k - \mathbf{M}^{-1}\mathbf{b}) - \frac{1}{2}\mathrm{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{W}_k)\Big) \times \underbrace{\mathrm{C}}_{\text{Normalization}}$$

This is Normal Inverse Wishart form. Hence,

$$q(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_k, \frac{1}{\kappa_k}\boldsymbol{\Sigma}_k)\mathcal{IW}(\boldsymbol{\Sigma}_k|\mathbf{W}_k, \nu_k) \tag{11.132}$$

$$\bar{\mathbf{x}}_k = \frac{1}{N_k}\sum_n^N r_{n,k} \cdot \mathbf{x}_n$$

$$\mathbf{m}_k = \mathbf{M}^{-1}\mathbf{b} = \frac{1}{\kappa_k}(\kappa_0\mathbf{m}_{0,k} + N_k\bar{\mathbf{x}}_k)$$

$$\kappa_k = \kappa_0 + N_k$$

$$\mathbf{W}_k = \mathbf{S}_{0,k} + \mathbf{S}_{\bar{x}_k} + \frac{k_0 N_k}{\kappa_0 + N_k}(\mathbf{m}_{0,k} - \bar{\mathbf{x}}_k)(\mathbf{m}_{0,k} - \bar{\mathbf{x}}_k)^T$$

$$\mathbf{S}_{\bar{\mathbf{x}}_k} = \sum_n^N r_{nk}(\mathbf{x}_n - \bar{\mathbf{x}}_k)(\mathbf{x}_n - \bar{\mathbf{x}}_k)^T$$

$$\nu_k = \nu_0 + N_k$$

### 11.3.11 VBEM GMM Update for $q(\mathbf{z})$ complete data case

$$\log q(\mathbf{z}) = \Big\langle \log p(\mathbf{X}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}, \mathbf{z}) \Big\rangle_{q(\mu,\Sigma,\pi)} + \mathrm{C} \tag{11.133}$$

$$= \Big\langle \log p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{z}) p(\boldsymbol{\mu}|\boldsymbol{\Sigma}) p(\boldsymbol{\Sigma}) \Big\rangle_{q(\mu,\Sigma)} + \Big\langle \log p(\mathbf{z}|\boldsymbol{\pi}) p(\boldsymbol{\pi}) \Big\rangle_{q(\pi)} + \mathrm{C} \tag{11.134}$$

$$= \Big\langle \log p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{z}) \Big\rangle_{q(\mu,\Sigma)} + \Big\langle \log p(\mathbf{z}|\boldsymbol{\pi}) \Big\rangle_{q(\pi)}$$
$$+ \underbrace{\Big\langle \log p(\boldsymbol{\mu}|\boldsymbol{\Sigma}) p(\boldsymbol{\Sigma}) \Big\rangle_{q(\mu,\Sigma)} + \Big\langle \log p(\boldsymbol{\pi}) \Big\rangle_{q(\pi)}}_{\text{Terms independent of } \mathbf{z}} + \mathrm{C}$$

$$= \Big\langle \sum_n^N \sum_k^K \log p(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}} \Big\rangle_{q(\mu,\Sigma)} + \Big\langle \sum_n^N \sum_k^K \log p(z_n = k|\boldsymbol{\pi})^{z_{nk}} \Big\rangle_{q(\pi)} + \mathrm{C} \tag{11.135}$$

$$= \Big\langle \sum_n^N \sum_k^K z_{nk} \cdot \log p(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \Big\rangle_{q(\mu,\Sigma)} + \Big\langle \sum_n^N \sum_k^K z_{nk} \cdot \log \boldsymbol{\pi}_k \Big\rangle_{q(\pi)} + \mathrm{C} \tag{11.136}$$

$$= \sum_n^N \left[ \sum_k^K \Big\langle z_{nk} \cdot \log p(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \Big\rangle_{q(\mu,\Sigma)} + \sum_k^K \Big\langle z_{nk} \cdot \log \boldsymbol{\pi}_k \Big\rangle_{q(\pi)} \right] + \mathrm{C} \tag{11.137}$$

$$\implies \log q(\mathbf{z}) = \sum_n^N \log q(z_n) \tag{11.138}$$

Hence,

$$\log q(z_n) = \sum_k^K \Big\langle z_{nk} \cdot \log p(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\Big\rangle_{q(\mu,\Sigma)} + \sum_k^K \Big\langle z_{nk} \cdot \log \boldsymbol{\pi}_k \Big\rangle_{q(\pi)} + \mathrm{C} \tag{11.139}$$

$$= \sum_k^K \Big[\Big\langle z_{nk} \cdot \log p(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\Big\rangle_{q(\mu,\Sigma)} + \Big\langle z_{nk} \cdot \log \boldsymbol{\pi}_k \Big\rangle_{q(\pi)}\Big] + \mathrm{C} \tag{11.140}$$

This gives

$$\log q(z_n = k) = \underbrace{\Big\langle \log p(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\Big\rangle_{q(\mu,\Sigma)}}_{(1)} + \underbrace{\Big\langle \log \boldsymbol{\pi}_k \Big\rangle_{q(\pi)}}_{(2)} + \mathrm{C} \tag{11.141}$$

(1) $$\Big\langle \log p(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\Big\rangle_{q(\mu,\Sigma)} = \Big\langle \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\Big\rangle_{q(\mu,\Sigma)} \tag{11.142}$$

$$= -\frac{1}{2}\Big\langle \log|\boldsymbol{\Sigma}_k| + (\mathbf{x}_n - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)\Big\rangle_{q(\mu,\Sigma)} \tag{11.143}$$

$$= -\frac{1}{2}\Big\langle \log|\boldsymbol{\Sigma}_k| + \mathbf{x}_n^T\boldsymbol{\Sigma}_k^{-1}\mathbf{x}_n - 2\mathbf{x}_n^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k + \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\Big\rangle_{q(\mu,\Sigma)} \tag{11.144}$$

$$= -\frac{1}{2}\Big[\Big\langle \log|\boldsymbol{\Sigma}_k|\Big\rangle_{q(\Sigma)} + \Big\langle \mathbf{x}_n^T\boldsymbol{\Sigma}_k^{-1}\mathbf{x}_n\Big\rangle_{q(\Sigma)} - \Big\langle 2\mathbf{x}_n^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\Big\rangle_{q(\mu,\Sigma)} + \Big\langle \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\Big\rangle_{q(\mu,\Sigma)}\Big]$$

$$= -\frac{1}{2}\Big[\big\langle \log|\boldsymbol{\Sigma}_k|\big\rangle_{q(\Sigma)} + \mathbf{x}_n^T\big\langle\boldsymbol{\Sigma}_k^{-1}\big\rangle_{q(\Sigma)}\mathbf{x}_n - 2\mathbf{x}_n^T\big\langle\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\big\rangle_{q(\mu,\Sigma)} + \Big\langle \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\Big\rangle_{q(\mu,\Sigma)}\Big] \tag{11.145}$$

Given that $\boldsymbol{\Sigma}_k \sim \mathcal{IW}(\boldsymbol{\Sigma}_k|\mathbf{W}_k, \nu_k)$, $\langle\boldsymbol{\Sigma}_k^{-1}\rangle = \nu_k\mathbf{W}_k^{-1}$, and $\boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_k, \frac{1}{\kappa_k}\boldsymbol{\Sigma}_k)$

$$= -\frac{1}{2}\Big[\big\langle \log|\boldsymbol{\Sigma}_k|\big\rangle_{q(\Sigma)} + \mathbf{x}_n^T(\nu_k\mathbf{W}_k^{-1})\mathbf{x}_n - 2\mathbf{x}_n^T(\nu_k\mathbf{W}_k^{-1})\mathbf{m}_k + \Big\langle \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\Big\rangle_{q(\mu,\Sigma)}\Big] \tag{11.146}$$

$$= -\frac{1}{2}\Big[\big\langle \log|\boldsymbol{\Sigma}_k|\big\rangle_{q(\Sigma)} + \mathbf{x}_n^T(\nu_k\mathbf{W}_k^{-1})\mathbf{x}_n - 2\mathbf{x}_n^T(\nu_k\mathbf{W}_k^{-1})\mathbf{m}_k + \Big\langle \mathrm{tr}\big(\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\big)\Big\rangle_{q(\mu,\Sigma)}\Big] \tag{11.147}$$

$$= -\frac{1}{2}\Big[\cdots + \Big\langle \mathrm{tr}\big(\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T\big)\Big\rangle_{q(\mu,\Sigma)}\Big] \tag{11.148}$$

$$= -\frac{1}{2}\Big[\cdots + \mathrm{tr}\Big(\big\langle\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T\big\rangle_{q(\mu,\Sigma)}\Big)\Big] \tag{11.149}$$

$$= -\frac{1}{2}\Bigg[ \cdots + \operatorname{tr}\Big(\Big\langle \boldsymbol{\Sigma}_k^{-1}(\mathbf{m}_k\mathbf{m}_k^T + \frac{1}{\kappa_k}\boldsymbol{\Sigma}_k)\Big\rangle_{q(\Sigma)}\Big)\Bigg] \tag{11.150}$$

$$= -\frac{1}{2}\Bigg[ \cdots + \operatorname{tr}\Big(\Big\langle \boldsymbol{\Sigma}_k^{-1}\mathbf{m}_k\mathbf{m}_k^T + \frac{1}{\kappa_k}\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\Sigma}_k\Big\rangle_{q(\Sigma)}\Big)\Bigg] \tag{11.151}$$

$$= -\frac{1}{2}\Bigg[ \cdots + \operatorname{tr}\Big((\nu_k\mathbf{W}_k^{-1})\mathbf{m}_k\mathbf{m}_k^T + \frac{1}{\kappa_k}I\Big)\Bigg] \tag{11.152}$$

$$= -\frac{1}{2}\Bigg[ \cdots + \nu_k\mathbf{m}_k^T\mathbf{W}_k^{-1}\mathbf{m}_k + \frac{D}{\kappa_k}\Bigg] \tag{11.153}$$

$$= -\frac{1}{2}\Bigg[ \big\langle \log|\boldsymbol{\Sigma}_k|\big\rangle_{q(\Sigma)} + \mathbf{x}_n^T(\nu_k\mathbf{W}_k^{-1})\mathbf{x}_n - 2\mathbf{x}_n^T(\nu_k\mathbf{W}_k^{-1})\mathbf{m}_k + \nu_k\mathbf{m}_k^T\mathbf{W}_k^{-1}\mathbf{m}_k + \frac{D}{\kappa_k}\Bigg] \tag{11.154}$$

$$= -\frac{1}{2}\Bigg[ \log|\mathbf{W}_k| - \sum_i^D \psi\left(\frac{\nu_k+1-i}{2}\right) + D\log 2 +$$

$$\mathbf{x}_n^T(\nu_k\mathbf{W}_k^{-1})\mathbf{x}_n - 2\mathbf{x}_n^T(\nu_k\mathbf{W}_k^{-1})\mathbf{m}_k + \nu_k\mathbf{m}_k^T\mathbf{W}_k^{-1}\mathbf{m}_k + \frac{D}{\kappa_k}\Bigg]$$

$$= -\frac{1}{2}\Bigg[ \log|\mathbf{W}_k| - \sum_i^D \psi\left(\frac{\nu_k+1-i}{2}\right) + D\log 2 + \nu_k(\mathbf{x}_n-\mathbf{m}_k)^T\mathbf{W}_k^{-1}(\mathbf{x}_n-\mathbf{m}_k) + \frac{D}{\kappa_k}\Bigg]$$

$$(2) \quad \mathbb{E}_{q(\boldsymbol{\pi})}\Big[\log\boldsymbol{\pi}_k\Big] = \psi(\alpha_k) - \psi\Big(\sum_{k'}\alpha_{k'}\Big) \tag{11.155}$$

Hence we get the update step :

$$\log q(z_n = k) \tag{11.156}$$

$$= -\frac{1}{2}\Bigg[ \log|\mathbf{W}_k| - \sum_i^D \psi\left(\frac{\nu_k+1-i}{2}\right) - D\log 2 + \nu_k(\mathbf{x}_n-\mathbf{m}_k)^T\mathbf{W}_k^{-1}(\mathbf{x}_n-\mathbf{m}_k) + \frac{D}{\kappa_k}\Bigg]$$

$$+ \psi(\alpha_k) - \psi\Big(\sum_{k'}\alpha_{k'}\Big) + \mathrm{C}$$

$$\implies q(z_n = k) = \exp\Bigg(\psi(\alpha_k) - \psi(\sum_{k'}\alpha_{k'}) \tag{11.157}$$

$$-\frac{1}{2}\Bigg[ \log|\mathbf{W}_k| - \sum_i^D \psi\left(\frac{\nu_k+1-i}{2}\right)$$

$$- D\log 2 + \nu_k(\mathbf{x}_n-\mathbf{m}_k)^T\mathbf{W}_k^{-1}(\mathbf{x}_n-\mathbf{m}_k) + \frac{D}{\kappa_k}\Bigg] + \underbrace{\mathrm{C}}_{\text{Normalization}}\Bigg)$$

### 11.3.12 VBEM GMM Update for $\boldsymbol{\mu}_k$ with missing data and MAP estimation for $\Sigma_k$

$$\log q(\boldsymbol{\mu}_k|\hat{\boldsymbol{\Sigma}}_k) = \tag{11.158}$$

$$\sum_n^N r_{nk}\Big\langle \log p(\mathbf{x}_n|\boldsymbol{\mu}_k, \hat{\boldsymbol{\Sigma}}_k)\Big\rangle_{q(X_H^n|z_n=k)} + \log p(\boldsymbol{\mu}_k|\hat{\boldsymbol{\Sigma}}_k) + \mathrm{C} \tag{11.159}$$

$$= \sum_n^N r_{nk}\Big\langle \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \hat{\boldsymbol{\Sigma}}_k)\Big\rangle_{q(X_H^n|z_n=k)} + \log \mathcal{N}(\boldsymbol{\mu}|\mathbf{m}_{0,k}, \boldsymbol{\Lambda}_k) + \mathrm{C} \tag{11.160}$$

$$\text{where } \boldsymbol{\Lambda}_k = \kappa_0 \hat{\boldsymbol{\Sigma}}_k^{-1} \tag{11.161}$$

$$= \sum_n^N r_{nk} \cdot \Big\langle -\frac{1}{2}\Big(\mathbf{x}_n^T\boldsymbol{\Sigma}_k^{-1}\mathbf{x}_n - 2\mathbf{x}_n^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k + \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\Big)\Big\rangle_{q(X_H^n|z_n=k)} \tag{11.162}$$

$$-\frac{1}{2}\Big(\boldsymbol{\mu}_k^T\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k - 2\boldsymbol{\mu}_k^T\boldsymbol{\Lambda}_k\mathbf{m}_{0,k} + \mathbf{m}_{0,k}^T\boldsymbol{\Lambda}_k\mathbf{m}_{0,k}\Big) + \mathrm{C} \tag{11.163}$$

$$= \sum_n^N r_{nk} \cdot \Big\langle -\frac{1}{2}\Big(\mathrm{tr}(\hat{\boldsymbol{\Sigma}}_k^{-1}\mathbf{x}_n\mathbf{x}_n^T) - 2\mathbf{x}_n^T\hat{\boldsymbol{\Sigma}}_k^{-1}\boldsymbol{\mu}_k + \boldsymbol{\mu}_k^T\hat{\boldsymbol{\Sigma}}_k^{-1}\boldsymbol{\mu}_k\Big)\Big\rangle_{q(X_H^n|z_n=k)} \tag{11.164}$$

$$-\frac{1}{2}\Big(\boldsymbol{\mu}_k^T\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k - 2\boldsymbol{\mu}_k^T\boldsymbol{\Lambda}_k\mathbf{m}_{0,k} + \mathbf{m}_{0,k}^T\boldsymbol{\Lambda}_k\mathbf{m}_{0,k}\Big) + \mathrm{C} \tag{11.165}$$

$$= \sum_n^N r_{nk} \cdot -\frac{1}{2}\Big(\mathrm{tr}(\hat{\boldsymbol{\Sigma}}_k^{-1}\langle\mathbf{x}_n\mathbf{x}_n^T\rangle_k) - 2\boldsymbol{\mu}_k^T\hat{\boldsymbol{\Sigma}}_k^{-1}\langle\mathbf{x}_n\rangle_k + \boldsymbol{\mu}_k^T\hat{\boldsymbol{\Sigma}}_k^{-1}\boldsymbol{\mu}_k\Big) \tag{11.166}$$

$$-\frac{1}{2}\Big(\boldsymbol{\mu}_k^T\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k - 2\boldsymbol{\mu}_k^T\boldsymbol{\Lambda}_k\mathbf{m}_{0,k} + \mathbf{m}_{0,k}^T\boldsymbol{\Lambda}_k\mathbf{m}_{0,k}\Big) + \mathrm{C} \tag{11.167}$$

$$= \sum_n^N r_{nk}\boldsymbol{\mu}_k^T\hat{\boldsymbol{\Sigma}}_k^{-1}\langle\mathbf{x}_n\rangle_k - \frac{1}{2}\sum_n^N r_{nk}\boldsymbol{\mu}_k^T\hat{\boldsymbol{\Sigma}}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k + \boldsymbol{\mu}_k^T\boldsymbol{\Lambda}_k\mathbf{m}_{0,k} + \mathrm{C} \tag{11.168}$$

$$= -\frac{1}{2}\left[N_k\boldsymbol{\mu}_k^T\hat{\boldsymbol{\Sigma}}_k^{-1}\boldsymbol{\mu}_k + \boldsymbol{\mu}_k^T\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k - 2\boldsymbol{\mu}_k^T\hat{\boldsymbol{\Sigma}}_k^{-1}\sum_n^N r_{nk}\langle\mathbf{x}_n\rangle_k - 2\boldsymbol{\mu}_k^T\boldsymbol{\Lambda}_k\mathbf{m}_{0,k}\right] + \mathrm{C} \tag{11.169}$$

$$= -\frac{1}{2}\left[\boldsymbol{\mu}_k^T\Big(N_K\hat{\boldsymbol{\Sigma}}_k^{-1} + \boldsymbol{\Lambda}_k\Big) - 2\boldsymbol{\mu}_k^T\Big(\hat{\boldsymbol{\Sigma}}_k^{-1}\sum_n^N r_{nk}\langle\mathbf{x}_n\rangle_k + \boldsymbol{\Lambda}_k\mathbf{m}_{0,k}\Big)\right] + \mathrm{C} \tag{11.170}$$

Complete the square (11.171)

$$= -\frac{1}{2}\Bigg[(\boldsymbol{\mu}_k - \mathbf{M}^{-1}\mathbf{b})^T\mathbf{M}(\boldsymbol{\mu}_k - \mathbf{M}^{-1}\mathbf{b}) + \underbrace{\mathbf{b}^T\mathbf{M}^{-1}\mathbf{b}}_{\text{Independent of } \boldsymbol{\mu}_k}\Bigg] + \text{C}, \quad \text{where} \tag{11.172}$$

$$\mathbf{M} = N_K\hat{\boldsymbol{\Sigma}}_k^{-1} + \boldsymbol{\Lambda}_k \tag{11.173}$$

$$\mathbf{b} = \boldsymbol{\Lambda}_k\mathbf{m}_{0,k} + \hat{\boldsymbol{\Sigma}}_k^{-1}\sum_n^N r_{nk}\langle\mathbf{x}_n\rangle_k \tag{11.174}$$

$$= -\frac{1}{2}(\boldsymbol{\mu}_k - \mathbf{M}^{-1}\mathbf{b})^T\mathbf{M}(\boldsymbol{\mu}_k - \mathbf{M}^{-1}\mathbf{b}) + \text{C} \tag{11.175}$$

$$\implies q(\boldsymbol{\mu}_k) = \exp(-\frac{1}{2}(\boldsymbol{\mu}_k - \mathbf{M}^{-1}\mathbf{b})^T\mathbf{M}(\boldsymbol{\mu}_k - \mathbf{M}^{-1}\mathbf{b})) + \text{C} \tag{11.176}$$

This is MVN form, hence

$$q(\boldsymbol{\mu}_k) = \mathcal{N}(\boldsymbol{\mu}_k \mid \mathbf{m}_k, \mathbf{V}_k) \tag{11.177}$$

$$N_k = \sum_n^N r_{nk} \tag{11.178}$$

$$\boldsymbol{\Lambda}_k = \kappa_0\hat{\boldsymbol{\Sigma}}_k^{-1} \tag{11.179}$$

$$\mathbf{V}_k = \Big(N_k\hat{\boldsymbol{\Sigma}}_k^{-1} + \boldsymbol{\Lambda}_k\Big)^{-1} \tag{11.180}$$

$$= \Big(N_k\hat{\boldsymbol{\Sigma}}_k^{-1} + \kappa_0\hat{\boldsymbol{\Sigma}}_k^{-1}\Big)^{-1} \tag{11.181}$$

$$= \frac{1}{N_k - \kappa_0}\hat{\boldsymbol{\Sigma}}_k \tag{11.182}$$

$$\mathbf{m}_k = \mathbf{V}_k\Big(\boldsymbol{\Lambda}_k\mathbf{m}_{0,k} + \hat{\boldsymbol{\Sigma}}_k^{-1}\sum_n^N r_{nk}\langle\mathbf{x}_n\rangle_k\Big) \tag{11.183}$$

$$= \mathbf{V}_k\Big(\kappa_0\hat{\boldsymbol{\Sigma}}_k^{-1}\mathbf{m}_{0,k} + \hat{\boldsymbol{\Sigma}}_k^{-1}\sum_n^N r_{nk}\langle\mathbf{x}_n\rangle_k\Big) \tag{11.184}$$

$$= \frac{1}{N_k - \kappa_0}\Big(\kappa_0\mathbf{m}_{0,k} + \sum_n^N r_{nk}\langle\mathbf{x}_n\rangle_k\Big) \tag{11.185}$$

### 11.3.13 VBEM GMM Derivation of update step for missing entries and latent component assignments $q(\mathbf{z}, \mathbf{X}_h)$

$$\log q(\mathbf{z}, \mathbf{X}_H) = \langle \log p(\mathbf{X}_O, \mathbf{X}_H) \rangle_{q(\boldsymbol{\phi}, \boldsymbol{\pi})} + \mathrm{C} \tag{11.186}$$

$$= \Bigg\langle \underbrace{\log p(\boldsymbol{\pi}) + \sum_k^K \log p(\boldsymbol{\phi}_k)}_{\text{Independent of } \mathbf{X}_H \text{ and } \mathbf{z}} + \sum_n^N \sum_k^K z_{nk} \log p(\boldsymbol{\pi}) + \sum_n^N \sum_k^K z_{nk} \log p(\mathbf{x}_O^n, \mathbf{x}_H^n \mid \boldsymbol{\phi}_k) \Bigg\rangle_{q(\boldsymbol{\phi}, \boldsymbol{\pi})} + \mathrm{C} \tag{11.187}$$

$$= \sum_n^N \left\langle \sum_k^K z_{nk} \log p(\boldsymbol{\pi}) + \sum_k^K z_{nk} \log p(\mathbf{x}_O^n, \mathbf{x}_H^n \mid \boldsymbol{\phi}_k) \right\rangle_{q(\boldsymbol{\phi}, \boldsymbol{\pi})} + \mathrm{C} \tag{11.188}$$

$$\implies \log q(\mathbf{z}, \mathbf{X}_H) = \sum_n^N \log q(z_n, \mathbf{x}_n) \tag{11.189}$$

$$\implies \log q(z_n, \mathbf{x}_n) = \left\langle \sum_k^K z_{nk} \log p(\boldsymbol{\pi}) + \sum_k^K z_{nk} \log p(\mathbf{x}_O^n, \mathbf{x}_H^n \mid \boldsymbol{\phi}_k) \right\rangle_{q(\boldsymbol{\phi}, \boldsymbol{\pi})} + \mathrm{C} \tag{11.190}$$

Assuming

$$\log q(\mathbf{z}, \mathbf{X}_H) = \log q(\mathbf{z}) + \log q(\mathbf{X}_H | \mathbf{z})$$

$$\log q(\mathbf{x}_H^n | z_n) = \ln q(\mathbf{x}_H^n, z_n) - \ln q(z_n) \tag{11.191}$$

$$= \log q(\mathbf{x}_H^n, z_n) + \mathrm{C} \tag{11.192}$$

$$= \Bigg\langle \underbrace{\sum_k^K z_{nk} \log p(\boldsymbol{\pi})}_{\text{Independent of } \mathbf{x}_H^n} + \sum_k^K z_{nk} \log p(\mathbf{x}_O^n, \mathbf{x}_H^n \mid \boldsymbol{\phi}_k) \Bigg\rangle_{q(\boldsymbol{\phi}, \boldsymbol{\pi})} + \mathrm{C} \tag{11.193}$$

$$= \sum_k^K \Big\langle z_{nk} \log p(\mathbf{x}_O^n, \mathbf{x}_H^n \mid \boldsymbol{\phi}_k) \Big\rangle_{q(\boldsymbol{\phi})} + \mathrm{C} \tag{11.194}$$

Hence if we condition on $z_n = k$

$$\log q(\mathbf{x}_H^n | z_n = k) = \Big\langle \log p(\mathbf{x}_O^n, \mathbf{x}_H^n \mid \boldsymbol{\phi}_k) \Big\rangle_{q(\boldsymbol{\phi})} + \mathrm{C} \tag{11.195}$$

$$\log q(\mathbf{x}_H^n|z_n=k) = \left\langle \sum_k^K \ln \mathcal{N}\left( \begin{bmatrix} \mathbf{x}_O^n \\ \mathbf{x}_H^n \end{bmatrix} \mid \boldsymbol{\phi} \right) \right\rangle_{q(\phi)} + \mathrm{C} \tag{11.196}$$

$$= \underbrace{-\frac{1}{2}(\mathbf{x}_O^n - \boldsymbol{\mu}_k^O)^T \boldsymbol{\Lambda}_k^{OO}(\mathbf{x}_O^n - \boldsymbol{\mu}_k^O)}_{\text{Independent of } \mathbf{x}_H^n} - \tag{11.197}$$

$$\left\langle \frac{1}{2}(\mathbf{x}_H^n - \boldsymbol{\mu}_k^H)^T \boldsymbol{\Lambda}_k^{HH}(\mathbf{x}_H^n - \boldsymbol{\mu}_k^H) - (\mathbf{x}_H^n - \boldsymbol{\mu}_k^H)^T \boldsymbol{\Lambda}_k^{HO}(\mathbf{x}_O^n - \boldsymbol{\mu}_k^O) \right\rangle_{q(\phi)} + \mathrm{C}$$

$$= \left\langle -\frac{1}{2}(\mathbf{x}_H^n - \boldsymbol{\mu}_k^H)^T \boldsymbol{\Lambda}_k^{HH}(\mathbf{x}_H^n - \boldsymbol{\mu}_k^H) - (\mathbf{x}_H^n - \boldsymbol{\mu}_k^H)^T \boldsymbol{\Lambda}_k^{HO}(\mathbf{x}_O^n - \boldsymbol{\mu}_k^O) \right\rangle_{q(\phi)} + \mathrm{C} \tag{11.198}$$

$$= -\frac{1}{2}(\mathbf{x}_H^n)^T \boldsymbol{\Lambda}_k^{HH} \mathbf{x}_H^n + \left\langle (\mathbf{x}_H^n)^T \left( \boldsymbol{\Lambda}_k^{HH} \boldsymbol{\mu}_k^H - \boldsymbol{\Lambda}_k^{HO}(\mathbf{x}_O^n - \boldsymbol{\mu}_k^O) \right) \right\rangle_{q(\phi)} + \mathrm{C} \tag{11.199}$$

$$= -\frac{1}{2}(\mathbf{x}_H^n)^T \boldsymbol{\Lambda}_k^{HH} \mathbf{x}_H^n + (\mathbf{x}_H^n)^T \left( \boldsymbol{\Lambda}_k^{HH} \langle \boldsymbol{\mu}_k^H \rangle - \boldsymbol{\Lambda}_k^{HO}(\mathbf{x}_O^n - \langle \boldsymbol{\mu}_k^O \rangle) \right) + \mathrm{C} \tag{11.200}$$

Completing the squares we get that this is Gaussian

$$q(\mathbf{x}_H^n|z_n=k) = \mathcal{N}(\mathbf{m}_{nk}^{H|O}, \mathbf{V}_k^{H|O}) \tag{11.201}$$

$$\mathbf{V}_k^{H|O} = (\boldsymbol{\Lambda}_k^{HH})^{-1}$$

$$\mathbf{m}_{nk}^{H|O} = \langle \boldsymbol{\mu}_k^H \rangle - (\boldsymbol{\Lambda}_k^{HH})^{-1} \boldsymbol{\Lambda}_k^{HO}(\mathbf{x}_O^n - \langle \boldsymbol{\mu}_k^O \rangle)$$

Alternatively

$$q(\mathbf{x}_H^n|z_n=k) = \mathcal{N}(\mathbf{m}_{nk}^{H|O}, \mathbf{V}_k^{H|O}) \tag{11.202}$$

$$\mathbf{V}_k^{H|O} = \boldsymbol{\Sigma}_k^{HH} - \boldsymbol{\Sigma}_k^{HO}(\boldsymbol{\Sigma}_k^{OO})^{-1} \boldsymbol{\Sigma}_k^{OH}$$

$$\mathbf{m}_{nk}^{H|O} = \mathbf{m}_k^H - \boldsymbol{\Sigma}_k^{HO}(\boldsymbol{\Sigma}_k^{OO})^{-1}(\mathbf{x}_O^n - \mathbf{m}_k^O)$$

**GMM VBEM derivation of update for latent components for missing case $q(z_n = k)$**

$$\ln q(z_n) = \ln q(z_n, \mathbf{x}_H^n) - \ln q(\mathbf{x}_H^n) + \mathrm{C} \tag{11.203}$$

$$= \Big\langle \sum_k^K z_{nk} \ln \boldsymbol{\pi}_k + \sum_k^K z_{nk} \ln p(\mathbf{x}_O^n, \mathbf{x}_H^n | \boldsymbol{\phi}_k) \Big\rangle_{q(\phi,\pi)} - \Big\langle \sum_k^K z_{nk} \ln p(\mathbf{x}_H^n | \mathbf{x}_O^n, \boldsymbol{\phi}_k) \Big\rangle_{q(\phi)} + \mathrm{C} \tag{11.204}$$

$$= \left\langle \sum_k^K z_{nk} \ln \boldsymbol{\pi}_k + \sum_k^K z_{nk} \ln p(\mathbf{x}_O^n | \boldsymbol{\phi}_k) + \sum_k^K z_{nk} \ln p(\mathbf{x}_H^n | \mathbf{x}_O^n, \boldsymbol{\phi}_k) \right\rangle_{q(\phi,\pi)} -$$

$$\left\langle \sum_k^K z_{nk} \ln p(\mathbf{x}_H^n | \mathbf{x}_O^n, \boldsymbol{\phi}_k) \right\rangle_{q(\phi)} + \mathrm{C} \tag{11.205}$$

$$= \left\langle \sum_k^K z_{nk} \ln \boldsymbol{\pi}_k + \sum_k^K z_{nk} \ln p(\mathbf{x}_O^n | \boldsymbol{\phi}_k) \right\rangle_{q(\phi,\pi)} + \mathrm{C} \tag{11.206}$$

$$= \sum_k^K \left[ \Big\langle z_{nk} \ln \boldsymbol{\pi}_k \Big\rangle_{q(\pi)} + \Big\langle z_{nk} \ln p(\mathbf{x}_O^n | \boldsymbol{\phi}_k) \Big\rangle_{q(\phi)} \right] + \mathrm{C} \tag{11.207}$$

$$\implies \ln q(z_n = k) = \Big\langle \ln \boldsymbol{\pi}_k \Big\rangle_{q(\pi)} + \Big\langle \ln p(\mathbf{x}_O^n | \boldsymbol{\phi}_k) \Big\rangle_{q(\phi)} + \mathrm{C} \tag{11.208}$$

$$= \psi(\alpha_k) - \psi\Big( \sum_{k'} \alpha_{k'} \Big) + \Big\langle \ln \mathcal{N}(\mathbf{x}_O^n | \boldsymbol{\mu}_k, \hat{\boldsymbol{\Sigma}}_k) \Big\rangle_{q(\boldsymbol{\mu}_k)} + \mathrm{C} \tag{11.209}$$

$$= (\cdots) + \Big\langle -\frac{1}{2} \log |\boldsymbol{\Sigma}_k^{OO}| - \frac{1}{2} (\mathbf{x}_O^n - \boldsymbol{\mu}_k^O)^T (\boldsymbol{\Sigma}_k^{OO})^{-1} (\mathbf{x}_O^n - \boldsymbol{\mu}_k^O) \Big\rangle_{q(\boldsymbol{\mu}_k)} + \mathrm{C} \tag{11.210}$$

$$= (\cdots) - \frac{1}{2} \Big\langle \log |\boldsymbol{\Sigma}_k^{OO}| + \mathbf{x}_O^{n\,T} (\boldsymbol{\Sigma}_k^{OO})^{-1} \mathbf{x}_O^n - 2 \mathbf{x}_O^{n\,T} (\boldsymbol{\Sigma}_k^{OO})^{-1} \boldsymbol{\mu}_k^O + \boldsymbol{\mu}_k^{O^T} (\boldsymbol{\Sigma}_k^{OO})^{-1} \boldsymbol{\mu}_k^O \Big\rangle_{q(\boldsymbol{\mu}_k)} + \mathrm{C} \tag{11.211}$$

$$= (\cdots) - \frac{1}{2} \Bigg[ \log |\boldsymbol{\Sigma}_k^{OO}| + \mathbf{x}_O^{n\,T} (\boldsymbol{\Sigma}_k^{OO})^{-1} \mathbf{x}_O^n - 2 \mathbf{x}_O^{n\,T} (\boldsymbol{\Sigma}_k^{OO})^{-1} \Big\langle \boldsymbol{\mu}_k^O \Big\rangle_{q(\boldsymbol{\mu}_k)}$$

$$+ \Big\langle \boldsymbol{\mu}_k^{O^T} (\boldsymbol{\Sigma}_k^{OO})^{-1} \boldsymbol{\mu}_k^O \Big\rangle_{q(\boldsymbol{\mu}_k)} \Bigg] + \mathrm{C}$$

$$= (\cdots) - \frac{1}{2} \left[ \log |\boldsymbol{\Sigma}_k^{OO}| + \mathbf{x}_O^{n\,T} (\boldsymbol{\Sigma}_k^{OO})^{-1} \mathbf{x}_O^n - 2 \mathbf{x}_O^{n\,T} (\boldsymbol{\Sigma}_k^{OO})^{-1} \mathbf{m}_k^O + \Big\langle \mathrm{tr}((\boldsymbol{\Sigma}_k^{OO})^{-1} \boldsymbol{\mu}_k^O \boldsymbol{\mu}_k^{O^T}) \Big\rangle_{q(\boldsymbol{\mu}_k)} \right] + \mathrm{C} \tag{11.212}$$

$$= (\cdots) - \frac{1}{2} \left[ \log |\boldsymbol{\Sigma}_k^{OO}| + \mathbf{x}_O^{n\,T} (\boldsymbol{\Sigma}_k^{OO})^{-1} \mathbf{x}_O^n - 2 \mathbf{x}_O^{n\,T} (\boldsymbol{\Sigma}_k^{OO})^{-1} \mathbf{m}_k^O + \mathrm{tr}\Big( (\boldsymbol{\Sigma}_k^{OO})^{-1} \big\langle \boldsymbol{\mu}_k^O \boldsymbol{\mu}_k^{O^T} \big\rangle_{q(\boldsymbol{\mu}_k)} \Big) \right] + \mathrm{C} \tag{11.213}$$

$$= (\cdots) - \frac{1}{2}\Bigg[\log|\boldsymbol{\Sigma}_k^{OO}| \mathbf{x}_O^{n\,T}(\boldsymbol{\Sigma}_k^{OO})^{-1}\mathbf{x}_O^n - 2\mathbf{x}_O^{n\,T}(\boldsymbol{\Sigma}_k^{OO})^{-1}\mathbf{m}_k^O + \mathrm{tr}\Big((\boldsymbol{\Sigma}_k^{OO})^{-1}(\mathbf{m}_k^O\mathbf{m}_k^{O^T} - \mathbf{V}_k^{OO})\Big)\Bigg] + \mathrm{C} \tag{11.214}$$

$$= (\cdots) - \frac{1}{2}\Bigg[\log|\boldsymbol{\Sigma}_k^{OO}| + \mathbf{x}_O^{n\,T}(\boldsymbol{\Sigma}_k^{OO})^{-1}\mathbf{x}_O^n - 2\mathbf{x}_O^{n\,T}(\boldsymbol{\Sigma}_k^{OO})^{-1}\mathbf{m}_k^O$$
$$+ \mathrm{tr}\Big(\boldsymbol{\Sigma}_k^{OO^{-1}}\mathbf{m}_k^O\mathbf{m}_k^{O^T} - \boldsymbol{\Sigma}_k^{OO^{-1}}\mathbf{V}_k^{OO}\Big)\Bigg] + \mathrm{C} \tag{11.215}$$

$$= (\cdots) - \frac{1}{2}\Bigg[\log|\boldsymbol{\Sigma}_k^{OO}| + \mathbf{x}_O^{n\,T}(\boldsymbol{\Sigma}_k^{OO})^{-1}\mathbf{x}_O^n - 2\mathbf{x}_O^{n\,T}(\boldsymbol{\Sigma}_k^{OO})^{-1}\mathbf{m}_k^O + \mathbf{m}_k^{O^T}\boldsymbol{\Sigma}_k^{OO^{-1}}\mathbf{m}_k^O$$
$$- \mathrm{tr}\Big(\boldsymbol{\Sigma}_k^{OO^{-1}}\left(\frac{1}{N_k+\kappa_0}\boldsymbol{\Sigma}_k\right)^{OO}\Big)\Bigg] + \mathrm{C}$$

$$= (\cdots) - \frac{1}{2}\Bigg[\log|\boldsymbol{\Sigma}_k^{OO}| + \mathbf{x}_O^{n\,T}(\boldsymbol{\Sigma}_k^{OO})^{-1}\mathbf{x}_O^n - 2\mathbf{x}_O^{n\,T}(\boldsymbol{\Sigma}_k^{OO})^{-1}\mathbf{m}_k^O + \mathbf{m}_k^{O^T}\boldsymbol{\Sigma}_k^{OO^{-1}}\mathbf{m}_k^O$$
$$- \frac{1}{N_k+\kappa_0}\mathrm{tr}\Big(\boldsymbol{\Sigma}_k^{OO^{-1}}\boldsymbol{\Sigma}_k^{OO}\Big)\Bigg] + \mathrm{C} \tag{11.216}$$

$$= (\cdots) - \frac{1}{2}\Bigg[\log|\boldsymbol{\Sigma}_k^{OO}| + \mathbf{x}_O^{n\,T}(\boldsymbol{\Sigma}_k^{OO})^{-1}\mathbf{x}_O^n - 2\mathbf{x}_O^{n\,T}(\boldsymbol{\Sigma}_k^{OO})^{-1}\mathbf{m}_k^O + \mathbf{m}_k^{O^T}\boldsymbol{\Sigma}_k^{OO^{-1}}\mathbf{m}_k^O - \frac{D_O}{N_k+\kappa_0}\Bigg] + \mathrm{C} \tag{11.217}$$

Complete the Square

$$= (\cdots) - \frac{1}{2}\Bigg[\log|\boldsymbol{\Sigma}_k^{OO}| + (\mathbf{x}_O^n - \mathbf{m}_k^O)^T\boldsymbol{\Sigma}_k^{OO^{-1}}(\mathbf{x}_O^n - \mathbf{m}_k^O) - \mathbf{m}_k^{O^T}\boldsymbol{\Sigma}_k^{OO^{-1}}\mathbf{m}_k^O + \mathbf{m}_k^{O^T}\boldsymbol{\Sigma}_k^{OO^{-1}}\mathbf{m}_k^O$$
$$- \frac{D_O}{N_k+\kappa_0}\Bigg] + \mathrm{C}$$

$$= (\cdots) - \frac{1}{2}\Bigg[\log|\boldsymbol{\Sigma}_k^{OO}| + (\mathbf{x}_O^n - \mathbf{m}_k^O)^T\boldsymbol{\Sigma}_k^{OO^{-1}}(\mathbf{x}_O^n - \mathbf{m}_k^O) - \frac{D_O}{N_k+\kappa_0}\Bigg] + \mathrm{C} \tag{11.218}$$

$$= \psi(\alpha_k) - \psi\Big(\sum_{k'}\alpha_{k'}\Big) - \frac{1}{2}\Bigg[\log|\boldsymbol{\Sigma}_k^{OO}| + (\mathbf{x}_O^n - \mathbf{m}_k^O)^T\boldsymbol{\Sigma}_k^{OO^{-1}}(\mathbf{x}_O^n - \mathbf{m}_k^O) - \frac{D_O}{N_k+\kappa_0}\Bigg] + \mathrm{C} \tag{11.219}$$

# Bibliography


[1] *Adjusted Random Index*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html.

[2] Giulia Carreras et. al. "Missing not at random in end of life care studies: multiple imputation and sensitivity analysis on data from the ACTION study". In: *BMC Medical Research Methodology* (Jan. 2021). DOI: 10.1186/s12874-020-01180-y. URL: https://doi.org/10.1186/s12874-020-01180-y.

[3] Javier Albusac et al. "Multi-analysis surveillance and dynamic distribution of computational resources: Towards extensible, robust, and efficient monitoring of environments". In: *Expert Systems with Applications* 175 (2021), p. 114692. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2021.114692. URL: https://www.sciencedirect.com/science/article/pii/S0957417421001330.

[4] Lorenzo Beretta and Alessandro Santaniello. "Nearest neighbor imputation algorithms: a critical evaluation". In: *BMC Medical Informatics and Decision Making* 16 (3 July 2016). DOI: 10.1186/s12911-016-0318-z. URL: https://doi.org/10.1186/s12911-016-0318-z.

[5] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. Chap. 10 Approximate Inference.

[6] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. Chap. 9 Mixture Models and EM.

[7] Tim Bock. "5 Ways to Deal with Missing Data in Cluster Analysis". In: (). URL: https://www.displayr.com/5-ways-deal-missing-data-cluster-analysis/#:~:text=However%2C%20we%20reach%C2%A0different%20conclusions%20when,rather%20than%20the%20data%20itself.

[8] S. F. Buck. "A Method of Estimation of Missing Values in Multivariate Data Suitable for use with an Electronic Computer". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 22.2 (1960), pp. 302–306. ISSN: 00359246. URL: http://www.jstor.org/stable/2984099 (visited on 05/31/2025).

[9] Stef van Buuren and Karin Groothuis-Oudshoorn. "mice: Multivariate Imputation by Chained Equations in R". In: *Journal of Statistical Software* 45.3 (2011), pp. 1–67. DOI: 10.18637/jss.v045.i03. URL: https://www.jstatsoft.org/index.php/jss/article/view/v045i03.

[10] Harlan Campbell, Tim Morris, and Paul Gustafson. "A fully Bayesian approach for the imputation and analysis of derived outcome variables with missingness". In: (2025). arXiv: 2404.09966 [stat.ME]. URL: https://arxiv.org/abs/2404.09966.

[11] George Casella, Christian P. Robert, and Martin T. Wells. "Mixture models, latent variables and partitioned importance sampling". In: *Statistical Methodology* 1.1 (2004), pp. 1–18. ISSN: 1572-3127. DOI: https://doi.org/10.1016/j.stamet.2004.05.001. URL: https://www.sciencedirect.com/science/article/pii/S1572312704000036.

[12] Scott D. Brown Don van Ravenzwaaij Pete Cassey. “A simple introduction to Markov Chain Monte-Carlo sampling”. In: (Feb. 2018). DOI: 10.3758/s13423-016-1015-8.

[13] David A. van Dyk and Taeyoung Park. “Partially collapsed Gibbs sampling and path-adaptive Metropolis-Hastings in high-energy astrophysics”. In: *Handbook of Markov Chain Monte Carlo* (2011). URL: https://www.ma.imperial.ac.uk/~dvandyk/Research/11-mcmc.bk-astro.pdf#:~:text=Collapsing%20in%20a%20Gibbs%20sampler,aims%20to%20construct%20an%20EM.

[14] Ronald Fisher. 1936. URL: https://www.kaggle.com/datasets/arshid/iris-flower-dataset.

[15] Yulei He. “Missing data analysis using multiple imputation: getting to the heart of the matter.” In: (Jan. 2010). DOI: 10.1161/CIRCOUTCOMES.109.875658. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC2818781/.

[16] “Imputation of Missing Values”. In: (). URL: https://scikit-learn.org/stable/modules/impute.html.

[17] *KNNImputer*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html.

[18] Alice Lilly. “Votes in the House of Commons”. In: (Oct. 2019). URL: https://www.instituteforgovernment.org.uk/explainer/votes-house-commons.

[19] Jun S. Liu, Wing Hung Wong, and Augustine Kong. “Covariance Structure of the Gibbs Sampler with Applications to the Comparisons of Estimators and Augmentation Schemes”. In: *Biometrika* 81.1 (1994), pp. 27–40. ISSN: 00063444. URL: http://www.jstor.org/stable/2337047 (visited on 08/03/2025).

[20] Zhihua Ma and Guanghui Chen. “Bayesian methods for dealing with missing data problems”. In: *Journal of the Korean Statistical Society* 47.3 (2018), pp. 297–313. ISSN: 1226-3192. DOI: https://doi.org/10.1016/j.jkss.2018.03.002. URL: https://www.sciencedirect.com/science/article/pii/S1226319218300176.

[21] Leigh Metcalf and William Casey. “Chapter 2 - Metrics, similarity, and sets”. In: (2016). Ed. by Leigh Metcalf and William Casey, pp. 3–22. DOI: https://doi.org/10.1016/B978-0-12-804452-0.00002-6. URL: https://www.sciencedirect.com/science/article/pii/B9780128044520000026.

[22] Dobra Adrian Miao Zhen Chen Yen-Chi. “Bayesian finite mixtures of Ising models”. In: *Metrika* (). DOI: 10.1007/s00184-024-00970-4. URL: https://doi.org/10.1007/s00184-024-00970-4.

[23] Kevin P. Murphy. *Machine Learning A Probablistic Perspective*. The MIT Press, 2012. Chap. 4.6.3.3 NIW Posterior.

[24] Kevin P. Murphy. *Machine Learning A Probablistic Perspective*. The MIT Press, 2012. Chap. 11 Mixture models and the EM algorithm.

[25] Kevin P. Murphy. *Machine Learning A Probablistic Perspective*. The MIT Press, 2012. Chap. 20.5 Computational intractability of exact inference in the worst case.

[26] Kevin P. Murphy. *Machine Learning A Probablistic Perspective*. The MIT Press, 2012. Chap. 24 Markov chain Monte Carlo (MCMC) inference.

[27] Kevin P. Murphy. *Machine Learning A Probablistic Perspective*. The MIT Press, 2012. Chap. 21.6 Variational Bayes EM.

[28] Kevin P. Murphy. *Machine Learning A Probablistic Perspective*. The MIT Press, 2012. Chap. 11.4 The EM algorithm.

[29] Kevin P. Murphy. *Machine Learning A Probablistic Perspective*. The MIT Press, 2012. Chap. 11.6.1 EM for the MLE of MVN with missing data.

[30] Kevin P. Murphy. *Machine Learning A Probablistic Perspective*. The MIT Press, 2012. Chap. 24.2.4 Collapsed Gibbs sampling *.

[31] *Public Whip : MPs' Voting Records*. URL: https://www.publicwhip.org.uk.

[32] Trivellore E. Raghunathan. "What Do We Do with Missing Data? Some Options for Analysis of Incomplete Data". In: *Annual Review of Public Health* 25.Volume 25, 2004 (2004), pp. 99–117. ISSN: 1545-2093. DOI: https://doi.org/10.1146/annurev.publhealth.25.102802.124410. URL: https://www.annualreviews.org/content/journals/10.1146/annurev.publhealth.25.102802.124410.

[33] Dan Roth. *On the hardness of approximate reasoning*. Vol. 82. 1. 1996, pp. 273–302. DOI: https://doi.org/10.1016/0004-3702(94)00092-1. URL: https://www.sciencedirect.com/science/article/pii/0004370294000921.

[34] DONALD B. RUBIN. "Inference and missing data". In: *Biometrika* 63.3 (Dec. 1976), pp. 581–592. ISSN: 0006-3444. DOI: 10.1093/biomet/63.3.581. eprint: https://academic.oup.com/biomet/article-pdf/63/3/581/756166/63-3-581.pdf. URL: https://doi.org/10.1093/biomet/63.3.581.

[35] Donald B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons, 1987. DOI: 10.1002/9780470316696.

[36] Matthew Stephens. "Dealing With Label Switching in Mixture Models". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 62.4 (Jan. 2002), pp. 795–809. URL: https://stephenslab.uchicago.edu/assets/papers/Stephens2000b.pdf.

[37] Elise Uberoi. "MPs in Parliament: Breakdown of activities by gender and party". In: (2020). URL: https://commonslibrary.parliament.uk/mps-in-parliament-breakdown-of-activities-by-gender-and-party/.

[38] Laura Wolf and Marcus Baum. *Deterministic Gibbs Sampling for Data Association in Multi-Object Tracking*. June 2020. DOI: 10.36227/techrxiv.12435398.v1.