Graphical Abstract

Conditional Latent Block gated Mixture of Experts

De Santiago, Kylliann, Ambroise, Christophe, Szafranski, Marie

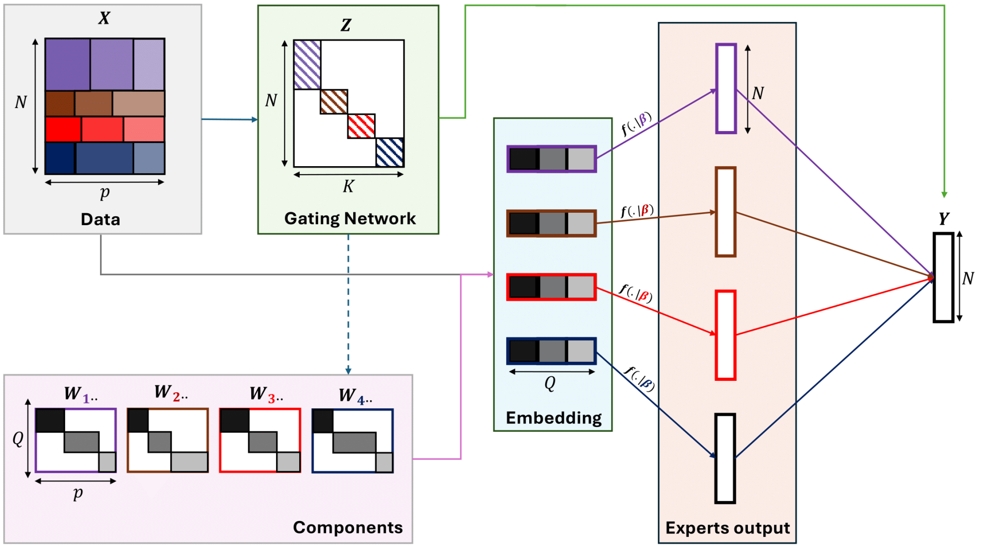


Figure 1: Schematic representation of *MoEBIUS*. The input data $\mathbf{X}$, of dimension $N \times p$, is used to define selection weights for the K experts (gating network $\mathbf{Z}$, with K communities). Each expert is associated with a projection matrix $\mathbf{W}_k$, of dimension $p \times Q$, that groups the covariates into components. The data, coupled with the projection matrices, are transformed into matrices of size $N \times Q$, which are then passed through a regression function $f_k(. \mid \boldsymbol{\beta}_k)$ specific to each expert. The outputs of the experts are combined according to the gating network's weights (based on community membership) to produce a final prediction $\mathbf{y}$, representing the target variable.

Highlights

Conditional Latent Block gated Mixture of Experts

De Santiago, Kylliann, Ambroise, Christophe, Szafranski, Marie

- **Structured Expert Assignment:** The gating network is enhanced with a conditional latent block model, allowing it to group variables into coherent components. This leads to more interpretable expert assignments.
- **Redundancy Reduction:** By clustering correlated features, MoE-BIUS minimizes redundant information, improving both computational efficiency and model generalization.
- **Improved Predictive Performance:** The integration of latent block models refines expert specialization, leading to higher accuracy in predictive tasks.

Conditional Latent Block gated Mixture of Experts

De Santiago, Kylliann[1], Ambroise, Christophe[1], Szafranski, Marie[1]

[a]*LaMME, UMR CNRS 8071, University of Paris-Saclay, 23 boulevard de France, Evry, 91000, , France*

Abstract

A key advantage of Mixture of Experts (MoE) is their capability for data stratification, enabling the identification and characterization of subpopulations by forming distinct communities. This is achieved through a gating network that dynamically assigns observations to specialized experts. However, a notable limitation of MoE models is their potential to function as black boxes, making variable interpretation challenging . The complexity of interpretation is influenced by the type of gating network and expert models employed.

We introduce an interpretable MoE framework that not only characterizes subpopulations but also captures the redundancy and complementarity of information specific to each community. To achieve this, we propose extending MoE models by incorporating a conditional biclustering structure as a gating network. This approach leads to interpretable MoEs that reveal the redundancy and complementarity of information by clustering variables into components. Additionally, this fusion enhances computational efficiency by allowing specialized experts to focus on the redescription of variables, with representative variables characterizing the components.

Keywords:

Mixture of Experts, Latent Block Model, Explainability

1. Introduction

Since its introduction more than two decades ago (?) Mixture of Experts (MoE) models have been widely used in machine learning due to their ability to partition data into meaningful subpopulations and assign specialized models to different regions of the input space.

Different types of expert architectures have been proposed, such as SVMs (?), Gaussian Processes (???), Dirichlet Processes (?), and deep networks.

These models rely on a gating function that dynamically assigns observations to experts, allowing for adaptive learning across heterogeneous datasets.

Conventional MoE models face challenges in interpretability. MoE models can act as black boxes, making it difficult to extract meaningful insights into how experts contribute to predictions.

To address these issues, we propose the MoEBIUS algorithm, which incorporates a conditional latent block structure into the traditional MoE framework. The key contributions of MoEBIUS are:

- **Structured Expert Assignment:** The gating network is enhanced with a conditional latent block model, allowing it to group variables into coherent components. This leads to more interpretable expert assignments.

- **Redundancy Reduction:** By clustering correlated features, MoEBIUS minimizes redundant information, improving both computational efficiency and model generalization.

- **Improved Predictive Performance:** The integration of latent block models refines expert specialization, leading to higher accuracy in predictive tasks.

The remainder of this paper is structured as follows: Section 2 is a background section presenting the context of Mixture of Experts and how our proposal relates to other close work, Section 3 discusses the methodology and formalizes the MoEBIUS model. Section 4 provides experimental validation, comparing MoEBIUS against standard MoE approaches. Finally, Section 5 concludes with a discussion on potential applications and future research directions.

2. Background and related work

In Mixture of Experts, although each expert receives all input variables, only a subset is truly relevant to the specific task assigned to the expert. This variable selection is frequently implicit, as each expert focuses on the most useful characteristics for its task. However, this selection can be made explicit through regularization mechanisms, such as L_1 (**?**), L_2 (**?**), or elastic net (**??**). For K experts, this would be equivalent to maximize the following quantity:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\pi}; \mathbf{y}, \mathbf{X}) = \prod_{i=1}^{N} \sum_{k=1}^{K} & \underbrace{g\left(\mathbf{X}_i; \boldsymbol{\pi}\right)_k}_{\text{Gating Network}} \; \underbrace{f_k\left(y_i; \mathbf{X}_i, \boldsymbol{\beta}_k\right)}_{\text{Expert}} \\ & + \underbrace{\lambda\left(\alpha \sum_{k=1}^{K} \sum_{j=1}^{p} |\pi_{kj}| + \frac{1-\alpha}{2} \sum_{k=1}^{K} \sum_{j=1}^{p} \pi_{kj}^2\right)}_{\text{Regularization terms}}, \qquad (1)\end{aligned}$$

where $g\left(\mathbf{X}_i;\boldsymbol{\pi}\right)_k$ is the k-th output of the gating network detailed in Equation (**??**), $\boldsymbol{\pi}$ are gating network parameters, $(\boldsymbol{\beta}_k)_{k=1:K}$ are experts parameters, $\lambda > 0$ is an hyperparameter linked to regularization strength and $\alpha > 0$ is an hyperparameter for elastic-net regularization.

These regularizations, primarily applied to the gating network, enhance interpretability by controlling expert selection directly. The gating network plays a central role in variable and expert selection, and different approaches can be classified into three main categories **?**: dense (**??**), sparse (**???**), and soft (**??**).

In sparse methods, regularizations are designed to activate a limited number of experts or variables, which improves the interpretability by aligning predictions with the variables involved (**?**). The main approach is to use the top-L experts, meaning the L experts with the highest values according to the gating network: in general, the gating network g is composed of $(g_k)_{k=1:K}$ functions, and is defined as

$$g(\mathbf{X}_i;\boldsymbol{\pi}) = \operatorname{softmax}\left(g_1\left(\mathbf{X}_i;\boldsymbol{\pi}_1\right), \cdots, g_K\left(\mathbf{X}_i;\boldsymbol{\pi}_K\right)\right). \tag{2}$$

Typically, to limit the number of activated experts, according to **?**, the gating network output $g(\mathbf{X}_i)_k$ is replaced by:

$$\tilde{g}(\mathbf{X}_i)_k = \operatorname{softmax}\left(\operatorname{TopL}\left(g_1\left(\mathbf{X}_i;\boldsymbol{\pi}_1\right)\right), \cdots, \operatorname{TopL}\left(g_K\left(\mathbf{X}_i;\boldsymbol{\pi}_K\right)\right)\right)_k, \tag{3}$$

where

$$\operatorname{TopL}(g_k(\mathbf{X}_i;\boldsymbol{\pi})) = \begin{cases} g_k(\mathbf{X}_i;\boldsymbol{\pi}) & \text{if } g(\mathbf{X}_i;\boldsymbol{\pi})_k \in \text{top-L elements of } g(\mathbf{X}_i;\boldsymbol{\pi}), \\ -\infty & \text{else.} \end{cases} \tag{4}$$

Currently, sparse MoEs typically refers to limiting the number of active experts, thereby reducing computational cost. However, an increased specialization among a large pool of experts does not necessarily ensure sparsity in terms of variable selection. On the other hand, soft approaches allow more gradual and partial expert activation, offering a balance between sparsity and flexibility. All MoEs with regularization mechanisms on the gating network fall into this class of models. While these models achieve sparsity in terms of variable selection, they remain soft in terms of expert activation. Indeed, it is not always possible to ensure that only a minimal subset of experts will be activated.

The integration of these penalties structures the gating network's behavior and encourages a parsimonious selection of variables. This simplifies interpretation by reducing the number of variables and experts involved in decision-making, while maintaining computational efficiency (**?**). However, the separation between variables used by the gating network and those leveraged by experts in the final prediction can complicate the interpretability, as the most influential variables for selection are not always the most relevant for prediction. This disjunction introduces a potential bias in model interpretation. Several efforts have been made to improve the interpretability of MoEs, for instance:

- **?** developed an interpretable MoE model, introducing an approach that makes expert decisions more transparent by employing explanation techniques such as assignment modules and studying expert contributions.

- Since 2017, the integration of attention mechanisms in MoEs, as presented by **?**, has made the decision process more explicit by focusing the model's attention on specific aspects of the data. Additionally, this work introduced the concept of the MoE-layer, which has been widely reused in recent Deep Learning architectures (**???**).

- In the context of Multitask Learning, **?** proposed using multiple gating networks, linked to the associated prediction tasks, providing a clearer connection between the experts utilized, the gating network predictions, and the final outcomes.

Ensuring interpretability is an important aspect in machine learning applied to sensitive areas. In Mixture Of Experts framework, which can be achieved not only through the design of the gating mechanism but also by employing interpretable experts. Experts based on linear regression are a common choice in this regard, as they allow for a clear justification of the decisions made by each expert. Moreover, experts based on linear regression are often favored due to their computational efficiency.

These type of Mixture of Experts models generally fall within the category of Latent Regression Models (LRMs), where the experts are based on latent subpopulations, and the regression coefficients capture subgroup-specific effects (**??**). More specifically, the general model is often defined as follows:

$$\mathbf{Z}_i \sim \mathcal{M}\left(1; \boldsymbol{\pi} = (\pi_1, \cdots, \pi_K)\right), \tag{5}$$

$$y_i \mid \mathbf{X}_i, Z_{ik} = 1 \sim \mathcal{N}\left(\mathbf{X}_i \boldsymbol{\beta}_k, \sigma_k^2\right), \tag{6}$$

where $\mathbf{Z}_i$ is a latent variable for expert allocation for observation i and $\boldsymbol{\beta}_k$ are regression parameters for the k-th expert (or community).

In this framework, the role of the gating network is associated with the variable Z. This variable refers to community membership, it is assumed that individuals from the same cluster are studied by the same expert.

Nevertheless, several extensions have been developed, including longitudinal data models and co-clustering regression models.

Longitudinal data models:. Individuals within the same community share similar dynamics over time. Each latent class has its own risk profile for the event under study or its own longitudinal pattern (**??**). Following the model of **?**, and with our notations, we would have :

$$\mathbf{Z}_i \sim \mathcal{M}\left(1; \text{softmax}\left(\mathbf{X}_i \boldsymbol{\pi}\right)\right), \tag{7}$$

$$y_{iv} \mid \mathbf{X}_i, \mathbf{T}_i, Z_{ik} = 1 \sim \mathcal{N}\left(\sum_{r=1}^{R} \beta_{kvr} T_{iv}^{r}, \sigma_{vk}^{2}\right). \tag{8}$$

In this context, $\mathbf{X}$ is considered as a $N \times p$ matrix. The latent variable is obtained by a multiclass regression parameterized by a $p \times K$ matrix $\boldsymbol{\pi}$. On the v-th visit and for the community k, the polynomial regression is parameterized by $\boldsymbol{\beta}_{kv}$, and T_{iv} is the time metric (the patient's age or time since the disease was first diagnosed, for instance).

Co-clustering Regression Model:. **?** proposed a multitask regression algorithm based on an *LBM* partitioning of the $N \times p$ label matrix $\mathbf{y}$. Each block of the matrix is associated with regression parameters for the covariates $\mathbf{X}$. The Latent Block Regression Model (*LBRM*) developed by **?** extends this by partitioning the $N \times p \times d$-tensor $\mathbf{X}$ based on the same stratification as the label matrix $\mathbf{y}$. The last dimension of the tensor $\mathbf{X}$ corresponds to the covariates used in the regression. More precisely, and with our notations :

$$\mathbf{Z}_i \sim \mathcal{M}\left(1; \boldsymbol{\pi} = (\pi_1, \cdots, \pi_K)\right), \tag{9}$$

$$\mathbf{W}_j \sim \mathcal{M}\left(1; \boldsymbol{\rho} = (\rho_1, \cdots, \rho_Q)\right), \tag{10}$$

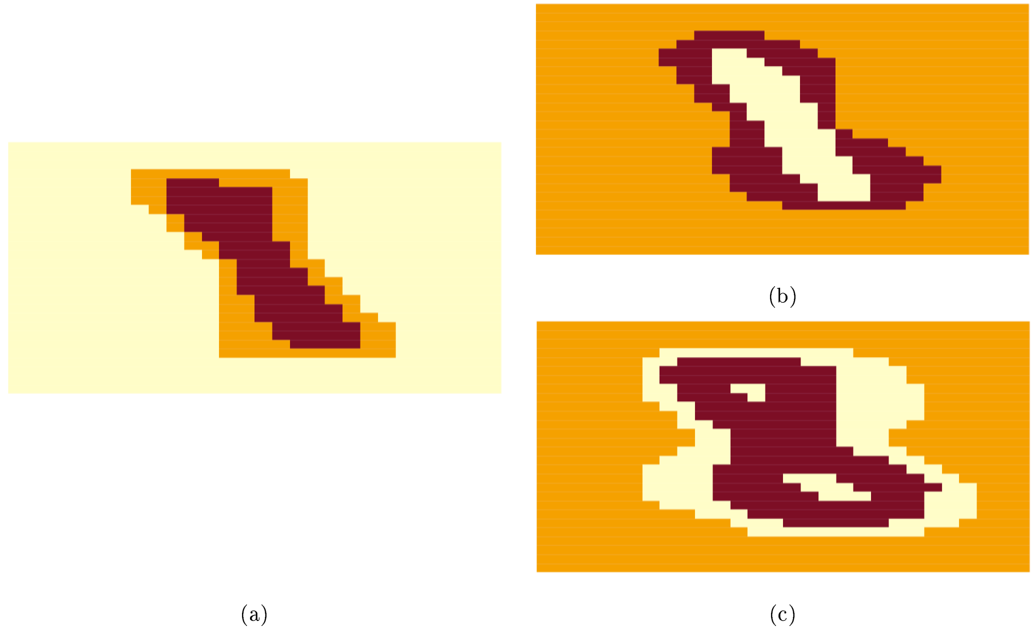
$$y_{ij} \mid \mathbf{X}_{ij}, Z_{ik} = 1, W_{js} = 1 \sim \mathcal{N}\left(\mathbf{X}_{ij}\boldsymbol{\beta}_{ks}, \sigma^2_{ks}\right). \tag{11}$$

For an observation i belonging to community k and a problem j in category s, a regression is performed along the third dimension, on $(\mathbf{X}_{ijr})_{r=1:d}$, parameterized by a coefficient vector $\boldsymbol{\beta}_{ks} \in \mathbb{R}^d$.

In the supervised framework, we aim to achieve both interpretability and predictive performance. The objective of our approach is to incorporate an additional structure into the data used for prediction. While the stratification of individuals is already induced by the Mixture of Experts framework, we aim to extend this by also integrating a structure on the variables. Through a mixture model applied to covariates, the goal is to capture the emergence of both redundant information within components and complementary information between components. This approach mirrors the work on *mimi-SBM*, which handles similar challenges but for multi-view clustering.

As illustrated in Figure **??**, to obtain more specific partitions compared to a standard mixture model on covariates, a conditional stratification of variables, depending on the community structure, is needed. The resulting model, named Mixture Of Experts and BIclustering Unified Strategy (*MoE-BIUS*), merges MoEs with a conditinal biclustering algorithm to offer precise predictions alongside clear interpretation of latent communities and components.

More specifically, the model summarizes the components into a representative variable for each, then performs a regression on these representations, where the regression parameters depend on the community to which the observation belongs, an illustration of MoEBIUS is provided in Figure **??**.


(b)

(a)

(c)

Figure 2: Different component partitions on MNIST data for digits 1 and 8. Figure (a) shows a partition defined by an *LBM*, where the overall shape of both digits is captured. Figures (b) and (c) are derived from the *Conditional LBM* (**?**), where the partitioning of variables depends on the observations (i.e., the digits). For each partition, the key pixels of interest specific to each digit are more clearly identified.

3. MoEBIUS: Mixture of Experts and BIclustering Unified Strategy

3.1. Model

MoEBIUS relies on incorporating a conditional variable stratification mechanism within Mixtures of Experts. To achieve this, we introduce a new random variable, denoted as $\mathbf{W}$. The purpose of the variable $\mathbf{W}$ is to determine which covariables are considered and how they are utilized for prediction. This leads us to the following general formulation:

$$\mathbb{P}\left(\mathbf{y},\mathbf{W},\mathbf{Z}\mid\mathbf{X},\boldsymbol{\beta},\boldsymbol{\rho},\boldsymbol{\pi}\right)=\underbrace{\mathbb{P}\left(\mathbf{y}\mid\mathbf{W},\mathbf{Z},\mathbf{X},\boldsymbol{\beta}\right)}_{\text{Experts}}\underbrace{\mathbb{P}\left(\mathbf{W}\mid\mathbf{Z},\mathbf{X},\boldsymbol{\rho}\right)}_{\text{Variable stratification}}\underbrace{\mathbb{P}\left(\mathbf{Z}\mid\mathbf{X},\boldsymbol{\pi}\right)}_{\text{Gating network}}. \tag{12}$$

At this stage, the specific dimensions of variables are not crucial; these details will be provided in Section **??**.

The definition of experts and the law associated with $\mathbf{y}$ are intrinsically dependent on the nature of the data. However, they can be formalized through a function f such that $f(\mathbf{y},\mathbf{W},\mathbf{Z},\mathbf{X},\boldsymbol{\beta})$, with $\boldsymbol{\beta}$ the expert parameters. The variable $\mathbf{W}$ is influenced by both $\mathbf{Z}$ and $\mathbf{X}$, allowing for a stratification that depends simultaneously on the communities and the individual-specific characteristics. Finally, the modeling of communities via $\mathbf{Z}$ is itself conditioned by $\mathbf{X}$; however, a marginal law could be considered.

3.2. Mixture Of Experts and BIclustering Unified Strategy model

As in **?**, the assignment to a community is obtained through a multiclass regression.

$$\mathbf{Z}_i\mid\mathbf{X}_i\sim\mathcal{M}\left(1;\operatorname{softmax}\left(\mathbf{X}_i\boldsymbol{\pi}\right)\right), \tag{13}$$

where $\boldsymbol{\pi}$ is a matrix of dimension $p\times K$.

The goal is to predict the latent variable $\mathbf{Z}_+$ for a new observation $\mathbf{X}_+$, leveraging information obtained during model training. In this context, a regression-based approach is more appropriate than a marginal distribution, as it directly incorporates the relationships learned between the input variables and the latent variables.

Now, based on the Conditional Latent block Model (CLBM) approach developed by **?**, we define the $K \times p \times Q$ tensor $\mathbf{W}$, which models the partition of the p variables into Q components.

These partitions are conditioned by the K communities, meaning that for each community k, the partition of the variables may differ, hence introducing the conditional aspect. Conditionally to the cluster k, each row of the component membership matrix follows:

$$\mathbf{W}_{kj} \sim \mathcal{M}\left(1; \boldsymbol{\rho}_k = (\rho_{k1}, \ldots, \rho_{kQ})\right). \tag{14}$$

One could have considered incorporating the contribution of $\mathbf{X}_i$ in the modeling process, allowing for a partition dependent on the individual rather than the community. However, this would have complicated the interpretability. The rationale behind this approach is to partially summarize the behavior of individuals through the communities to which they belong, making this modeling choice preferable.

In addition, an analysis of the effectiveness of the CLBM under the assumption of constant communities and conditional variable partitions has been conducted, through a study of the Co-Conditional Latent Block Model (*Co-CoLBM*). Parameter estimation, model selection and performance on simulated data have been studied. The code is available on Github (`https://github.com/Kdesantiago`). The detailed results of this study are presented in Appendix **??**.

In modeling the response variable $\mathbf{y}$ conditionally on covariates $\mathbf{X}$ and latent variables $\mathbf{Z}$ and $\mathbf{W}$, different strategies can be employed depending on the nature of the response variable's support. In this chapter, we focus on two primary settings: regression for continuous outcomes and multinomial

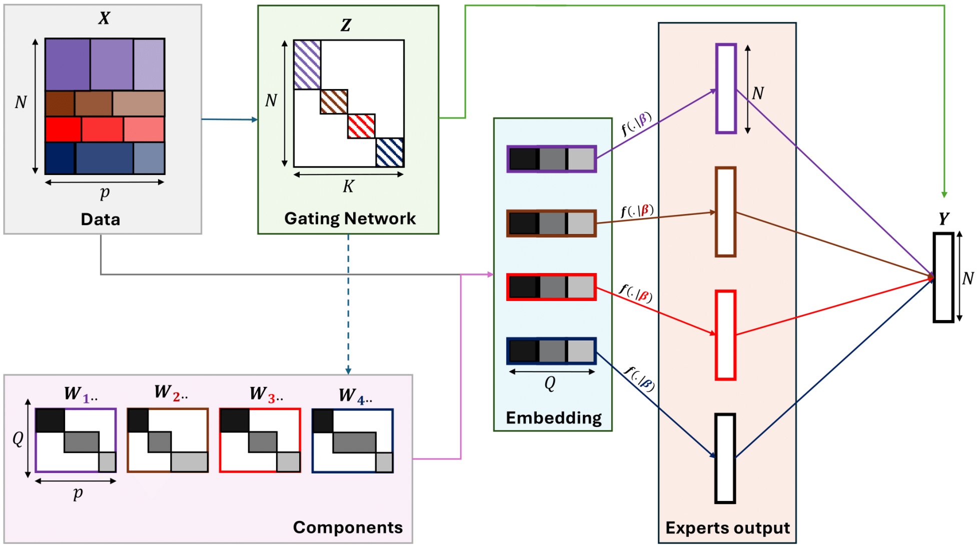


Figure 3: Schematic representation of *MoEBIUS*. The input data $\mathbf{X}$, of dimension $N \times p$, is used to define selection weights for the K experts (gating network $\mathbf{Z}$, with K communities). Each expert is associated with a projection matrix $\mathbf{W}_k$, of dimension $p \times Q$, that groups the covariates into components. The data, coupled with the projection matrices, are transformed into matrices of size $N \times Q$, which are then passed through a regression function $f_k(. \mid \boldsymbol{\beta}_k)$ specific to each expert. The outputs of the experts are combined according to the gating network's weights (based on community membership) to produce a final prediction $\mathbf{y}$, representing the target variable.

logistic regression for categorical outcomes.

Regression. When the support of $\mathbf{y}$ lies in $\mathbb{R}^N$, we adopt the following regression model:

$$y_i \mid \mathbf{X}_i, Z_{ik} = 1, \mathbf{W}_k \sim \mathcal{N}\left(y_i; \mathbf{X}_i \mathbf{W}_k \boldsymbol{\beta}_k, \sigma_k^2\right). \tag{15}$$

In this formulation, the regression parameters $\boldsymbol{\beta}_k \in \mathbb{R}^Q$ for each latent class k must be estimated, and σ_k^2, representing the class-specific noise variance, too.

Multinomial logistic regression. For the case where the support of $\mathbf{y}$ is $\{1, \ldots, C\}^N$, we use a multinomial logistic regression type model:

$$y_i \mid \mathbf{X}_i, Z_{ik} = 1, \mathbf{W}_k \sim \mathcal{M}\left(1; \operatorname{softmax}\left(\mathbf{X}_i \mathbf{W}_k \boldsymbol{\beta}_k\right)\right). \tag{16}$$

In this case, the classification parameters $\boldsymbol{\beta}_k \in \mathbb{R}^{Q \times C}$ are parameter matrices corresponding to each latent class k.

For a given community k, this model can be interpreted as a regression on the vector $(\mathbf{X}_i \mathbf{W}_k) \in \mathbb{R}^Q$. The components of this vector represent synthesized features of individual i, conditioned on the partition of community k, and reduced to Q variables. In other words, the p original variables are summarized into Q representative variables, each corresponding to a specific component. The graphical representation of the proposed model is provided in Figure **??**.

The complete likelihood can be expressed as:

$$\begin{aligned}\mathcal{L}^c\left(\boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\pi}; \mathbf{y}, \mathbf{Z}, \mathbf{W} \mid \mathbf{X}\right) &= \mathbb{P}\left(\mathbf{y}, \mathbf{Z}, \mathbf{W} \mid \mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\beta}\right) \\ &= \prod_{i=1}^{N} \prod_{k=1}^{K} \mathbb{P}\left(y_i \mid \mathbf{Z}_{ik}, \mathbf{W}_k, \mathbf{X}_i, \boldsymbol{\beta}_k\right) \prod_{k=1}^{K} \prod_{j=1}^{p} \mathbb{P}\left(\mathbf{W}_{kj} \mid \mathbf{Z}, \boldsymbol{\rho}\right) \\ &\prod_{i=1}^{N} \mathbb{P}\left(\mathbf{Z}_i \mid \mathbf{X}_i, \boldsymbol{\pi}\right).\end{aligned} \tag{17}$$

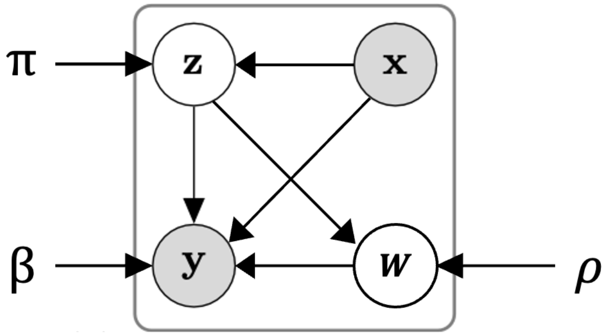


Figure 4: Graphical model for *MoEBIUS*. The joint distribution can be factorized as follows:
$\mathbb{P}(\mathbf{y}, \mathbf{Z}, \mathbf{W} \mid \mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\beta}) = \mathbb{P}(\mathbf{y} \mid \mathbf{X}, \mathbf{Z}, \mathbf{W}, \boldsymbol{\beta}) \, \mathbb{P}(\mathbf{W} \mid \mathbf{Z}, \boldsymbol{\rho}) \, \mathbb{P}(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\pi})$.

3.3. Model Selection

The model selection process for *MoEBIUS* is based on the ICL criterion proposed by **?** for conditional co-clustering and the BIC criterion introduced by **?** to penalize the regression component.

$$\begin{aligned}\text{BIC\_ICL}(\mathbf{y}, \mathbf{X}, K, Q) = & \log \mathbb{P}\left(\mathbf{y}, \hat{\mathbf{Z}}, \hat{\mathbf{W}} \mid \mathbf{X}, \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\rho}}, \hat{\boldsymbol{\beta}}\right) - p\frac{K-1}{2}\log(N) \\ & - K\frac{Q-1}{2}\log(p) - \frac{O_{prob}}{2}\log(N), \end{aligned} \quad (18)$$

where O_{prob} represents the number of parameters depending on the type of model used for $\mathbf{y}$.

The criterion penalizes the log-likelihood based on the number of free parameters, thus preventing overfitting:

- For communities $\boldsymbol{\pi}$: $p \times (K-1)$ free parameters.

- For components $\boldsymbol{\rho}$: $K \times (Q-1)$ free parameters.
- For regression $\boldsymbol{\beta}$: $O_{prob} = 2 \times K \times Q$ free parameters.
- For multiclass classification $\boldsymbol{\beta}$: $O_{prob} = (C-1) \times K \times Q$ free parameters.

The best model is the one that maximizes the BIC_ICL criterion, achieving a balance between predictive performance and model complexity.

Appendix A. Optimization via SEM-Gibbs algorithm

The Variational EM, as applied in the *mimiSBM* model (Section **??**), is not utilized in this context due to the increased computational complexity. This complexity stems from the dependency between the latent variables forming the tensor $\mathbf{W}$, which is introduced by the regression component.

However, Gibbs sampling EM emerges as a viable alternative. This approach relies on sampling from the conditional distributions of each variable given the others, which, in our case, are known. This makes Gibbs sampling a practical and implementable strategy for optimizing the model while managing the latent structure effectively.

Through a stochastic EM formulation based on Gibbs sampling, we can estimate latent variables and model parameters for Latent Block Models (**?**). Moreover, Gibbs sampling is a widely used Monte Carlo Markov Chain method for Bayesian inference in complex statistical models (**???**).

As in classical EM algorithms, this optimization consists of two steps: the *Stochastic Expectation Gibbs-sampling step* (SE-Gibbs step) and the *Maximization step* (M step). We begin by defining the objective function to be optimized at each iteration of the algorithm:

$$\mathcal{J}\left(\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\beta} \mid \boldsymbol{\pi}^{(t)}, \boldsymbol{\rho}^{(t)}, \boldsymbol{\beta}^{(t)}\right) = \mathbb{E}_{\mathbf{Z}, \mathbf{W} \sim \mathbb{P}\left(. \mid \mathbf{y}, \mathbf{X}, \boldsymbol{\pi}^{(t)}, \boldsymbol{\rho}^{(t)}, \boldsymbol{\beta}^{(t)}\right)} \left[\log \mathbb{P}\left(\mathbf{y}, \mathbf{Z}, \mathbf{W} \mid \mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\beta}\right)\right]. \tag{A.1}$$

Appendix A.1. Gibbs-sampling Expectation step

At step (t) of the algorithm, in Gibbs-sampling Expectation step, we first compute the following probabilities:

$$\tau_{ik}^{(t+1)} = \frac{\dfrac{e^{\mathbf{X}_i \boldsymbol{\pi}_{\bullet k}^{(t)}}}{\sum_{k'} e^{\mathbf{X}_i \boldsymbol{\pi}_{\bullet k'}^{(t)}}} \mathbb{P}\left(y_i \mid \mathbf{X}_i, Z_{ik} = 1, \hat{\mathbf{W}}_k^{(t)}, \boldsymbol{\beta}_k^{(t)}\right)}{\sum_{k'} \dfrac{e^{\mathbf{X}_i \boldsymbol{\pi}_{\bullet k'}^{(t)}}}{\sum_l e^{\mathbf{X}_i \boldsymbol{\pi}_{\bullet l}^{(t)}}} \mathbb{P}\left(y_i \mid \mathbf{X}_i, Z_{ik'} = 1, \hat{\mathbf{W}}_{k'}^{(t)}, \boldsymbol{\beta}_{k'}^{(t)}\right)}, \tag{A.2}$$

with $\boldsymbol{\pi}_{\bullet k} = (\pi_{ik})_{i=1:N}$.

Next, we perform a random draw from a multinomial distribution:

$$\hat{\mathbf{Z}}_i^{(t+1)} \sim \mathcal{M}\left(1; \left(\tau_{i1}^{(t+1)}, \ldots, \tau_{iK}^{(t+1)}\right)\right). \tag{A.3}$$

Similarly, the same operations are applied to $\boldsymbol{\nu}$ and $\mathbf{W}$:

$$\nu_{kjs}^{(t+1)} = \frac{\rho_{ks}^{(t)} \prod_i^N \mathbb{P}\left(y_i \mid \mathbf{X}_i, \hat{Z}_{ik}^{(t+1)}, W_{kjs} = 1, \hat{\mathbf{W}}_{-kj}^{(t)}, \boldsymbol{\beta}_k^{(t)}\right)^{\hat{Z}_{ik}^{(t+1)}}}{\sum_{s'} \rho_{ks'}^{(t)} \prod_i^N \mathbb{P}\left(y_i \mid \mathbf{X}_i, \hat{Z}_{ik}^{(t+1)}, W_{kjs'} = 1, \hat{\mathbf{W}}_{-kj}^{(t)}, \boldsymbol{\beta}_k^{(t)}\right)^{\hat{Z}_{ik}^{(t+1)}}}, \tag{A.4}$$

where the tensor $\hat{\mathbf{W}}_{-kj}^{(t)}$ corresponds to tensor $\hat{\mathbf{W}}^{(t)}$ with the third dimension element associated with the k-th entry of the first dimension and the j-th entry of the second removed.

$$\hat{\mathbf{W}}_{kj}^{(t+1)} \sim \mathcal{M}\left(1; \left(\nu_{kj1}^{(t+1)}, \ldots, \nu_{kjQ}^{(t+1)}\right)\right). \tag{A.5}$$

Appendix A.2. Maximization step

Using the results from the SE-Gibbs step, we now estimate the model parameters $\boldsymbol{\pi}, \boldsymbol{\rho}, (\boldsymbol{\beta}_k)_{k=1:K}$.

For the component parameters $\boldsymbol{\rho}$, The estimation follows a natural approach, where, conditionally on community k, the estimation is based on counting the number of variables within the component s and dividing by the total number of variables p.

$$\rho_{ks}^{(t+1)} = \frac{\sum_{j=1}^{p} \hat{\mathbf{W}}_{kjs}^{(t+1)}}{p}, \qquad \forall k \in \{1, \dots, K\}, \forall s \in \{1, \dots, Q\}. \tag{A.6}$$

The parameters $\boldsymbol{\pi}$ and $(\boldsymbol{\beta}_k)_{k=1:K}$ (for multiclass classification) are estimated using a gradient ascent approach, where a step is performed at each iteration of the *SEM-gibbs* algorithm.

Regarding the logistic regression parameters $\boldsymbol{\pi}$ on the latent variables $(\mathbf{Z}_i)_{i=1:N}$:

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{\pi}} \left(\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\beta} \mid \boldsymbol{\pi}^{(t)}, \boldsymbol{\rho}^{(t)}, \boldsymbol{\beta}^{(t)} \right) = \mathbf{X}^T \left(\hat{\mathbf{Z}}^{(t+1)} - \mathbf{S}^{\boldsymbol{\pi}^{(t)}} \right), \tag{A.7}$$

with

$$S_{ik}^{\boldsymbol{\pi}} = \frac{e^{\mathbf{X}_i \boldsymbol{\pi}_{\bullet k}}}{\sum_{k'} e^{\mathbf{X}_i \boldsymbol{\pi}_{\bullet k'}}}. \tag{A.8}$$

Here, $\hat{\mathbf{Z}}^{(t+1)}$ represents the "true" community membership matrix, while $\mathbf{S}^{\boldsymbol{\pi}}$ is an estimate. In other words, the parameters $\boldsymbol{\pi}$ are optimized to produce predictions consistent with the posterior.

The parameters are updated as follows:

$$\boldsymbol{\pi}^{(t+1)} = \boldsymbol{\pi}^{(t)} + h_t \frac{\partial \mathcal{J}}{\partial \boldsymbol{\pi}} \left(\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\beta} \mid \boldsymbol{\pi}^{(t)}, \boldsymbol{\rho}^{(t)}, \boldsymbol{\beta}^{(t)} \right), \tag{A.9}$$

where h_t is the gradient ascent step size for iteration t.

The estimation of parameters $(\boldsymbol{\beta}_k)_{k=1:K}$ depends on the problem at hand.

Regression. The estimation of $(\boldsymbol{\beta}_k)_{k=1:K}$ is given by:

$$\boldsymbol{\beta}_k^{(t+1)} = \left(W_k^{(t+1)^T} \mathbf{X}^T \operatorname{diag}\left(\hat{\mathbf{Z}}_{\bullet k}^{(t+1)}\right) \mathbf{X}\mathbf{W}_k^{(t+1)}\right)^{-1} \mathbf{W}_k^{(t+1)^T} \mathbf{X}^T \operatorname{diag}\left(\hat{\mathbf{Z}}_{\bullet k}^{(t+1)}\right) \mathbf{y}, \tag{A.10}$$

with $\hat{\mathbf{Z}}_{\bullet k}^{(t+1)} = \left(\hat{\mathbf{Z}}_{ik}^{(t+1)}\right)_{i=1:N}$. This is a weighted least squares estimator, where the weighting is provided by $\hat{\mathbf{Z}}_{\bullet k}^{(t+1)}$ and the data matrix is $\mathbf{X}\mathbf{W}_k^{(t+1)}$.

The estimation of $(\sigma_k^2)_{k=1:K}$ is given by:

$$\sigma_k^{2(t+1)} = \frac{\sum_{i=1}^N \hat{Z}_{ik}^{(t+1)} \left(y_i - \mathbf{X}_i \hat{\mathbf{W}}_k^{(t+1)} \boldsymbol{\beta}_k^{(t+1)}\right)^2}{\sum_{i=1}^N \hat{Z}_{ik}^{(t+1)}}. \tag{A.11}$$

Once again, we find the weighted least squares estimator for σ_k^2.

Multiclass Classification. The update for the parameters $(\boldsymbol{\beta}_k)_{k=1:K}$ is defined as:

$$\boldsymbol{\beta}_k^{(t+1)} = \boldsymbol{\beta}_k^{(t)} + h_t \frac{\partial \mathcal{J}}{\partial \boldsymbol{\beta}_k} \left(\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\beta} \mid \boldsymbol{\pi}^{(t)}, \boldsymbol{\rho}^{(t)}, \boldsymbol{\beta}^{(t)}\right), \tag{A.12}$$

where the gradient is given by:

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{\beta}_k} \left(\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\beta} \mid \boldsymbol{\pi}^{(t+1)}, \boldsymbol{\rho}^{(t+1)}, \boldsymbol{\beta}^{(t)}\right) = \left[\left(\mathbf{X}\hat{\mathbf{W}}_k^{(t+1)}\right)^T \odot \mathbf{1}_{Q,1} \hat{\mathbf{Z}}_{\bullet k}^{(t+1)^T}\right] \left(\mathbf{y} - \mathbf{S}^{\boldsymbol{\beta}_k^{(t)}}\right), \tag{A.13}$$

with the probability of variable y_i belonging to class c, for community k and given observation $\mathbf{X}_i$, defined as:

$$S_{ic}^{\boldsymbol{\beta}_k} = \frac{e^{\mathbf{X}_i \hat{\mathbf{W}}_k^{(t+1)} \boldsymbol{\beta}_{k \bullet c}}}{\sum_{c'} e^{\mathbf{X}_i \hat{\mathbf{W}}_k^{(t+1)} \boldsymbol{\beta}_{k \bullet c'}}}. \tag{A.14}$$

Here, $\mathbf{1}_{Q,1}$ is a matrix of size $Q \times 1$ filled with ones,$\odot$ represents the Hadamard (element-wise) product and $\boldsymbol{\beta}_{k\bullet c} = (\boldsymbol{\beta}_{ksc})_{s=1:Q}$.