Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II.

Morpheus: An R package for binary regression mixture models using spectral moments method

Mor Absa LOUM
UFR des Sciences de l'Ingénieur
Université Iba Der THIAM de Thiès
Thiès, Sénégal

Benjamin AUDER
Université Paris-Saclay, CNRS
Laboratoire de mathématiques d'Orsay
91405 Orsay, France

Abstract

morpheus is an R package for estimating parameters in mixture regression models within the context of binary classification. The package implements a hybrid methodology that leverages a novel algebraic identification procedure alongside conventional likelihood-based optimization. Initial parameter estimates are derived using a moment-based least squares approach integrated with tensor decomposition techniques, which facilitates the efficient recovery of latent mixture components. These initializations are subsequently refined through standard numerical optimization to achieve maximum likelihood. Performance evaluations on both synthetic and empirical datasets highlight the accuracy, numerical stability, and practical utility of the proposed framework.

Keywords: R, Generalized linear model, Mixture model, Moment method, Spectral method, Binary regression.

1. Introduction

Generalized linear models are widely used across various fields to analyze nonlinear relationships between discrete responses and covariates (Wedel and DeSarbo 1995). When the data originate from multiple subpopulations, finite mixture models are typically employed to account for such underlying heterogeneity (Frühwirth-Schnatter 2006; Fruhwirth-Schnatter, Celeux, and Robert 2019). Consequently, numerous algorithms have been developed to estimate these models; most existing approaches rely on likelihood-based methods, such as the expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin 1977) or Bayesian inference (Diebolt and Robert 1994). However, a well-known limitation of these methods is their propensity to converge toward spurious local maxima, often requiring a computationally

expensive search over multiple initializations (Wu 1983).

To address these challenges, moment-based methods—leveraging recent advances in tensor decomposition—offer a more robust alternative. Tensor methods rely on the fact that, in many latent variable models, low-order moments provide sufficient information to render the inverse problem efficiently solvable through straightforward computational steps (Anandkumar, Ge, Hsu, Kakade, and Telgarsky 2014). Building on this paradigm, the **morpheus** package implements the M3LS procedure. This approach combines a spectral initialization step, based on the joint diagonalization of third-order moment matrices, with a subsequent weighted least squares optimization. The spectral phase yields consistent preliminary estimates of the regression directions, significantly reducing sensitivity to local optima during the refinement stage. The resulting two-step procedure provides a robust and computationally efficient alternative to traditional likelihood-based estimation.

2. Model and estimation method

2.1. Model specification

Let $\mathbf{x} \in \mathbb{R}^d$ denote a vector of covariates and $Y \in \{0, 1\}$ be the associated binary response. Consider a parameter vector $\boldsymbol{\beta} \in \mathbb{R}^d$ and an intercept $b \in \mathbb{R}$. The quantity $\langle \boldsymbol{\beta}, \mathbf{x} \rangle + b$ represents an affine transformation of the input, mapping the covariates to a real scalar. In a classification framework, where the output is binary, this value is typically transformed into a probability using a non-decreasing link function $g : \mathbb{R} \to [0, 1]$. Thus, a baseline model is defined as follows:

$$\mathbb{P}(Y = 1 \mid \mathbf{X} = \mathbf{x}) = g(\langle \boldsymbol{\beta}, \mathbf{x} \rangle + b) \tag{1}$$

While this approach is effective when the data are homogeneous, many practical datasets exhibit latent subpopulations, requiring a more flexible representation. To account for such heterogeneity, we consider a finite mixture of binary regressions. Under this model, the conditional probability is given by:

$$\mathbb{P}(Y = 1 \mid \mathbf{X} = \mathbf{x}) = \sum_{j=1}^{k} \omega_j g(\langle \boldsymbol{\beta}_j, \mathbf{x} \rangle + b_j) \tag{2}$$

where k is the number of components, and ω_j represents the mixing proportions, satisfying $\omega_j \geq 0$ and $\sum_{j=1}^{k} \omega_j = 1$. Each component is characterized by its own regression coefficients $\boldsymbol{\beta}_j$ and intercept b_j.

2.2. Estimation method

The proposed estimation framework is based on minimizing an objective function defined by the discrepancy between theoretical moments and their empirical estimates. Auder, Gassiat, and Loum (2021) demonstrated that model parameters can be recovered by matching cross-

moments up to the third order. The empirical moments are defined as follows:

$$\widehat{\mathbf{M}}_1 = \frac{1}{n}\sum_{i=1}^{n} Y_i \mathbf{X}_i,$$
$$\widehat{\mathbf{M}}_2 = \frac{1}{n}\sum_{i=1}^{n} Y_i \left(\mathbf{X}_i \otimes \mathbf{X}_i - \sum_{j=1}^{d} \mathbf{e}_j \otimes \mathbf{e}_j \right),$$
$$\widehat{\mathbf{M}}_3 = \frac{1}{n}\sum_{i=1}^{n} Y_i \left(\mathbf{X}_i \otimes \mathbf{X}_i \otimes \mathbf{X}_i - \sum_{j=1}^{d} [\mathbf{X}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_j + \mathbf{e}_j \otimes \mathbf{X}_i \otimes \mathbf{e}_j + \mathbf{e}_j \otimes \mathbf{e}_j \otimes \mathbf{X}_i] \right),$$

where $\mathbf{e}_j$ denotes the j-th canonical basis vector. To estimate the parameter vector θ, we minimize the following least-squares objective function:

$$Q_n(\theta) = \sum_j \{\widehat{\mathbf{M}}_1[j] - \mathbf{M}_1(\theta)[j]\}^2 + \sum_{j,k} \{\widehat{\mathbf{M}}_2[j,k] - \mathbf{M}_2(\theta)[j,k]\}^2 + \sum_{j,k,l} \{\widehat{\mathbf{M}}_3[j,k,l] - \mathbf{M}_3(\theta)[j,k,l]\}^2. \tag{3}$$

The resulting least-squares moment method (LSMM) estimator is then:

$$\widehat{\theta}_n = \arg\min_{\theta \in \Theta} Q_n(\theta). \tag{4}$$

Under the assumptions detailed in Auder *et al.* (2021), $\widehat{\theta}_n$ is a consistent estimator and the model is identifiable.

Following the Generalized Method of Moments (GMM) framework (Hansen 1982), we introduce a weighted version of the estimator. Let $\tilde{\mathbf{M}}_i(\theta)$ be the residual vector for observation i:

$$\tilde{\mathbf{M}}_i(\theta) = Y_i \begin{pmatrix} \mathbf{X}_i \\ \mathrm{vec}(\mathbf{X}_i \otimes \mathbf{X}_i - \dots) \\ \mathrm{vec}(\mathbf{X}_i \otimes \mathbf{X}_i \otimes \mathbf{X}_i - \dots) \end{pmatrix} - \begin{pmatrix} \mathbf{M}_1(\theta) \\ \mathbf{M}_2(\theta) \\ \mathbf{M}_3(\theta) \end{pmatrix}. \tag{5}$$

For a symmetric positive definite weighting matrix $\mathbf{W}$ of size $(d + d^2 + d^3)$, the generalized objective function is defined as:

$$Q_n^{\mathbf{W}}(\theta) = \left(\frac{1}{n}\sum_{i=1}^{n} \tilde{\mathbf{M}}_i(\theta) \right)^\top \mathbf{W} \left(\frac{1}{n}\sum_{i=1}^{n} \tilde{\mathbf{M}}_i(\theta) \right), \tag{6}$$

yielding the estimator $\widehat{\theta}_n^{\mathbf{W}} = \arg\min_{\theta \in \Theta} Q_n^{\mathbf{W}}(\theta)$. Note that $\widehat{\theta}_n^{\mathbf{I}} = \widehat{\theta}_n$ for the identity matrix $\mathbf{I}$, an optimal choice of $\mathbf{W}$ typically improves efficiency. The **morpheus** package handles the joint estimation of the optimal weighting matrix and the parameters θ, leveraging the asymptotic properties established in Auder *et al.* (2021).

2.3. Algorithmic implementation

The **morpheus** package implements a multi-stage estimation procedure. Following a strategy similar to the iterative refinement used in GMM, the algorithm alternates between optimizing the objective $Q_n^{\mathbf{W}}(\theta)$ with respect to θ and updating the weighting matrix $\mathbf{W}$. Initially, we set $\mathbf{W} = \mathbf{I}_d$ to obtain a preliminary estimate, which is then used to compute an empirical optimal weighting matrix for a subsequent refinement step.

Algorithm 1: *InitDir* — Spectral Direction Estimation	
Input:	Data $(\mathbf{X}, Y)$, number of components K.
Step 1:	Compute empirical moments $\widehat{\mathbf{M}}_1, \widehat{\mathbf{M}}_2$, and $\widehat{\mathbf{M}}_3$.
Step 2:	Select a set of vectors $\{\mathbf{z}_1, \ldots, \mathbf{z}_P\} \subset \mathbb{R}^d$ (e.g., canonical basis).
Step 3:	Form the matrices $\widehat{\mathbf{B}}(\mathbf{z}_p)$ for $p = 1, \ldots, P$.
Step 4:	Perform joint diagonalization to find $\mathbf{V}$ such that $\mathbf{V}\widehat{\mathbf{B}}(\mathbf{z}_p)\mathbf{V}^\top$ are approximately diagonal for all p.
Step 5:	Extract $\mathbf{U}$ as the first K columns of $\mathbf{V}^{-1}$, ordered by decreasing absolute eigenvalues.
Step 6:	Resolve sign ambiguity: compute $\mathbf{o} = \mathbf{U}^\dagger \widehat{\mathbf{M}}_1$.
Step 7:	Update $\mathbf{U}_{\cdot,k} \leftarrow -\mathbf{U}_{\cdot,k}$ for all k such that $\mathbf{o}_k < 0$.
Output:	Preliminary directional estimators $\widehat{\boldsymbol{\mu}}_1, \ldots, \widehat{\boldsymbol{\mu}}_K$.

Table 1: The *InitDir* algorithm for joint diagonalization-based direction recovery.

The parameter vector θ is parameterized by the regression vectors $\boldsymbol{\beta}_k$, each decomposed into a direction $\boldsymbol{\mu}_k$ (where $\|\boldsymbol{\mu}_k\| = 1$) and a magnitude λ_k, such that $\boldsymbol{\beta}_k = \lambda_k \boldsymbol{\mu}_k$ for $k = 1, \ldots, K$. To mitigate the local optima issues typical of purely likelihood-based methods, **morpheus** employs a spectral initialization to recover the latent directions.

Spectral initialization of directions The preliminary estimation of the directions $\{\boldsymbol{\mu}_k\}_{k=1}^K$ relies on the properties of the third-order moment tensor. For any $\mathbf{z} \in \mathbb{R}^d$, let $\mathbf{B}(\mathbf{z})$ be the $d \times d$ matrix defined by:

$$\mathbf{B}(\mathbf{z})_{i,j} := \sum_{s=1}^{d} \mathbf{M}_3(\theta)_{i,j,s} z_s. \tag{7}$$

As established in Auder *et al.* (2021), this matrix admits the following decomposition:

$$\mathbf{B}(\mathbf{z}) = \sum_{k=1}^{K} \omega_k \lambda_k^3 \mathbb{E}[g^{(3)}(\lambda_k \langle \mathbf{X}, \boldsymbol{\mu}_k \rangle + b_k)] \langle \boldsymbol{\mu}_k, \mathbf{z} \rangle \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top. \tag{8}$$

Since the matrices $\mathbf{B}(\mathbf{z}_p)$ share the same set of eigenvectors $\{\boldsymbol{\mu}_k\}$ for any choice of $\mathbf{z}_p$, the directions can be recovered via joint diagonalization (Afsari 2008). This involves finding an orthogonal matrix $\mathbf{V}$ that simultaneously diagonalizes a set of matrices $\{\mathbf{B}(\mathbf{z}_1), \ldots, \mathbf{B}(\mathbf{z}_P)\}$ for $P \geq 2$. In practice, the columns of $\mathbf{V}^\top$ (or the first K rows of the diagonalizing matrix) provide the estimated directions, subject to potential sign and permutation indeterminacy.

To resolve the sign ambiguity, we leverage the first-order moment. Let $\mathbf{U}$ be the matrix whose columns are the K recovered directions. By considering the projection $\mathbf{o} = \mathbf{U}^\dagger \mathbf{M}_1$, where $\mathbf{U}^\dagger$ denotes the Moore-Penrose pseudoinverse, the elements of $\mathbf{o}$ correspond to the terms $\omega_k \lambda_k \mathbb{E}[g'(\lambda_k \langle \mathbf{X}, \boldsymbol{\mu}_k \rangle + b_k)]$. Since these terms are non-negative under standard link functions, any direction $\boldsymbol{\mu}_k$ associated with a negative entry in $\mathbf{o}$ is multiplied by -1. This spectral procedure, summarized in Table 1, provides a robust starting point for the subsequent numerical optimization of the moment-matching objective.

In the second stage, the estimator $\widehat{\theta}_n^{\mathbf{W}}$ is computed using standard numerical optimization routines. Initially, the objective function $Q_n(\theta)$ is minimized with $\mathbf{W} = \mathbf{I}_d$ (or a user-specified matrix), starting from the spectral estimates obtained via Algorithm 1. To enhance efficiency, the weighting matrix $\mathbf{W}$ can be re-estimated using these first-pass parameters, followed by a

final refinement of θ to minimize $Q_n^{\mathbf{W}}$. The complete estimation workflow, denoted as M3LS, is summarized in Table 2 and illustrated in Figure 1.

Algorithm 2: *M3LS* — Full Parameter Estimation	
Input:	Data $(\mathbf{X}, Y)$, number of components K, link function g.
Step 1:	Initialize directions $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ using Algorithm *InitDir*.
Step 2:	Minimize $Q_n(\theta)$ using the spectral directions as starting values.
Step 3:	(Optional) Update $\mathbf{W}$ and re-optimize to obtain the GLSMM estimator.
Output:	Final parameter estimate $\widehat{\theta}$.

Table 2: The *M3LS* procedure for robust mixture regression estimation.

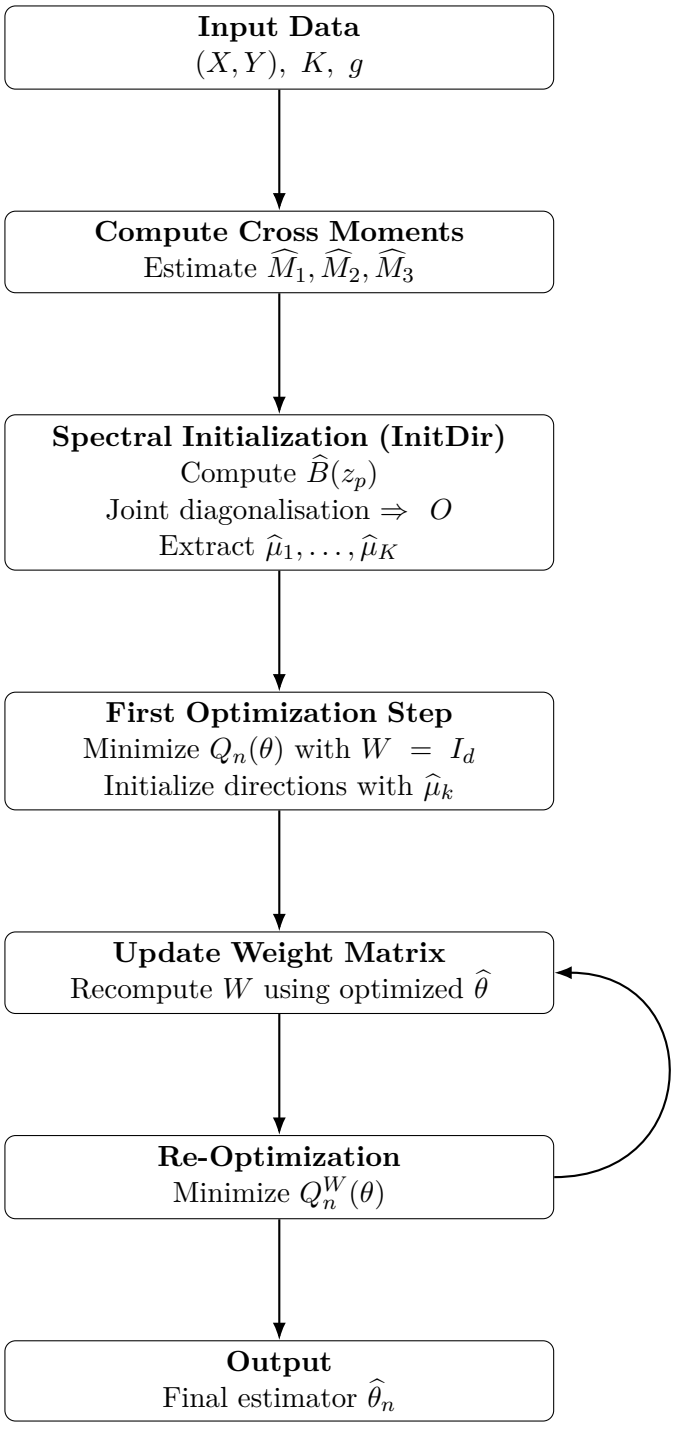

Figure 1: Workflow of the two-step M3LS estimation procedure combining spectral initialization and iterative weighted optimization.

3. Package implementation and usage

The **morpheus** package provides three core functions for model estimation:

- `computeMu()`: Implements the spectral method to estimate regression directions.
- `optimParams()`: Constructs an optimization object to estimate the full parameter set, starting from the spectral initialization.
- `multiRun()`: A wrapper designed for Monte Carlo simulations or bootstrap procedures across various model specifications.

Additionally, the package includes `generateSampleIO()`, a utility to simulate multivariate data according to the mixture models described in Section 2.

3.1. Direction estimation

In a typical workflow, the user provides a covariate matrix $\mathbf{X}$ and a binary response vector Y. For demonstration purposes, **morpheus** includes a data generator:

```
R> library("morpheus")
R> beta <- matrix(c(1, 0, 0, 1), ncol = 2)
R> io <- generateSampleIO(n = 10000, p = c(0.5, 0.5), beta = beta,
+                         b = c(0, 0), link = "probit")
```

In `generateSampleIO()`, `n` specifies the sample size, `p` is the vector of mixing proportions ω (summing to 1), `beta` is the $d \times K$ matrix of coefficients, and `b` is the vector of intercepts. The `link` argument supports `"logit"` or `"probit"` specifications. The resulting object `io` contains the simulated `X` and `Y`.

The regression directions can then be recovered as follows:

```
R> mu <- computeMu(io$X, io$Y, optargs = list(K = 2))
```

The `optargs` list allows for fine-grained control, including:

- `K`: The number of components (clusters).
- `M`: An optional pre-computed moments object from `computeMoments()`.
- `jd_method`: The joint-diagonalization algorithm (e.g., `"uwedge"` or `"jedi"`).
- `P`: The number of random vectors used for diagonalization.

3.2. Full parameter refinement

Once directions are obtained, the remaining parameters are estimated by solving the moment-matching optimization problem. The function `optimParams()` initializes an optimization runner:

```
R> # Optional argument: M <- computeMoments(io$X, io$Y)
R> runner <- optimParams(io$X, io$Y, K = 2, link = "probit")
R> # Initialize at spectral directions with b = 0 and uniform omega
R> theta <- runner$run(list(beta = mu))
```

The returned `theta` is a list containing the estimated proportions (ω), the regression matrix (β), and the intercepts (b).

3.3. Monte Carlo simulations and bootstrapping

The **morpheus** package includes a flexible wrapper, `multiRun()`, designed to facilitate comparative studies and resampling methods. This function coordinates parallel execution (via the **parallel** package) of multiple estimation strategies on shared datasets. It requires three primary components: a list of common parameters, a list of estimation functions to be evaluated, and a `prepareArgs` callback to pre-process data (e.g., generating new samples or drawing bootstrap indices) for each iteration.

Example 1: Bootstrap comparison with flexmix In this scenario, we compare the spectral direction estimation of **morpheus** against the **flexmix** package (Gruen and Leisch 2025) using bootstrapped samples.

```
R> beta <- matrix(c(1, -2, 3, 1), ncol = 2)
R> io   <- generateSampleIO(n = 1000, p = 0.5, beta = beta, b = c(0, 0),
+                           link = "logit")
R> target_mu <- normalize(beta)
R>
R> # Define estimation functions
R> methods <- list(
+    morpheus = function(fargs) {
+      ind <- fargs$ind
+      computeMu(fargs$X[ind, ], fargs$Y[ind], fargs$optargs)
+    },
+    flexmix = function(fargs) {
+      dat <- as.data.frame(cbind(fargs$Y[fargs$ind], fargs$X[fargs$ind, ]))
+      m   <- flexmix(cbind(V1, 1 - V1) ~ 0 + ., data = dat, k = fargs$K,
+                     model = FLXMRglm(family = "binomial"))
+      out <- refit(m)
+      normalize(matrix(out@coef[1:(ncol(fargs$X) * fargs$K)], ncol = fargs$K))
+    }
+  )
R>
R> # Execution with 10 bootstrap iterations on 3 cores
R> mr1 <- multiRun(list(X = io$X, Y = io$Y, K = 2, optargs = list(K = 2)),
+                  methods, prepareArgs = function(fargs, index) {
+                    fargs$ind <- if(index == 1) 1:nrow(fargs$X)
+                                 else sample(1:nrow(fargs$X), replace = TRUE)
```

```
+                     fargs
+                   }, N = 10, ncores = 3)
```

Since mixture models are subject to label switching, the package provides `alignMatrices()` to reorder the estimated columns against a reference before visualization.

Example 2: Monte Carlo simulation The following example evaluates the full `M3LS` pipeline against **flexmix** by generating fresh datasets in each iteration.

```
R> mr2 <- multiRun(list(n = 1000, p = 0.5, beta = beta, b = c(0, 0)),
+                   methods, prepareArgs = function(fargs, index) {
+                     io <- generateSampleIO(fargs$n, fargs$p, fargs$beta,
+                                            fargs$b, link = "logit")
+                     list(X = io$X, Y = io$Y, K = 2, link = "logit")
+                   }, N = 10, ncores = 3)
```

3.4. Visualization of results

To analyze the distribution and precision of the estimators, **morpheus** provides several plotting utilities: `plotBox()`, `plotHist()`, and `plotCoefs()`.

- `plotBox()`: Displays boxplots of specific coefficients across different methods (see Figure 2).
- `plotCoefs()`: Visualizes the estimated coefficients against their true values, ordered by magnitude, to assess bias and variance (see Figure 3).

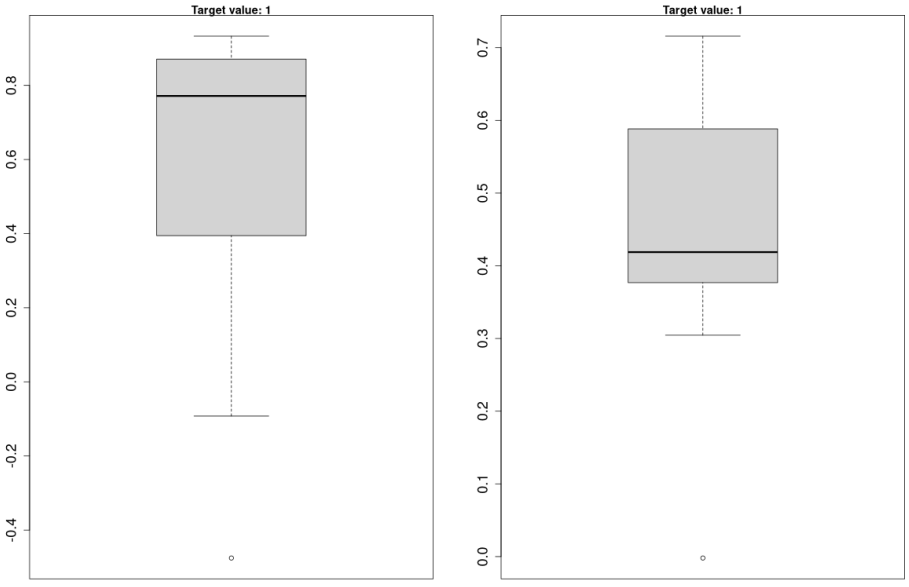

Figure 2: Boxplot comparison of the coefficient $\beta_{2,2}$ for **morpheus** and **flexmix** over 10 bootstrap runs.

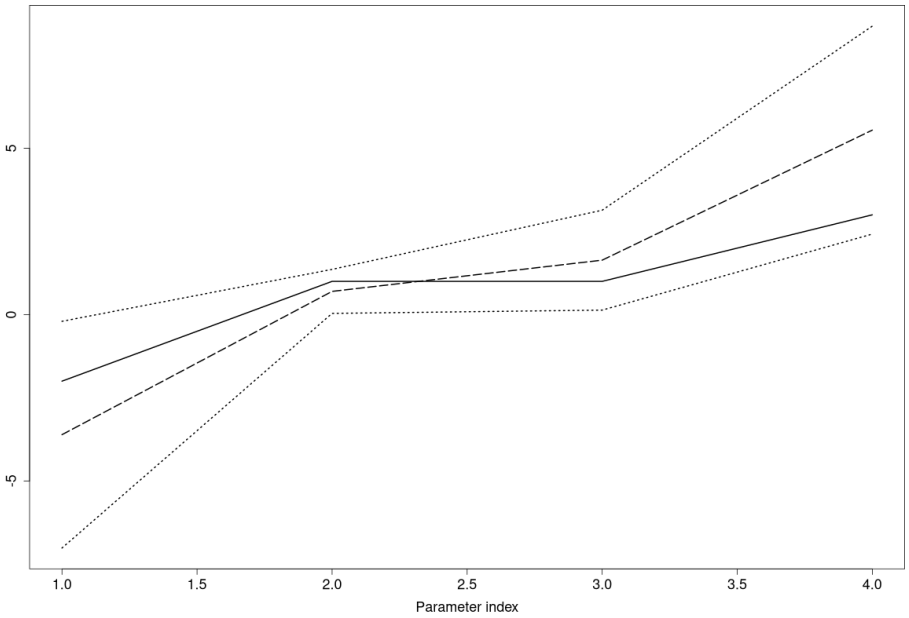

Figure 3: Estimated vs. true coefficients. Solid lines represent true parameters, while dotted lines show the range of estimations.

3.5. Variable selection and p-values

To identify the most influential covariates, a natural approach consists in testing the significance of the regression coefficients. Specifically, we test the null hypothesis $H_0 : \beta_{jk} = 0$ against the alternative $H_1 : \beta_{jk} \neq 0$. Under the asymptotic normality results established in Auder *et al.* (2021), the Wald-type statistic defined by $W = \hat{\beta}_{jk}^2 / \widehat{\mathrm{Var}}(\hat{\beta}_{jk})$ converges in distribution to a $\chi^2(1)$ law, where $\widehat{\mathrm{Var}}(\hat{\beta}_{jk})$ denotes the empirical variance estimated via bootstrapping. This approximation allows for the straightforward computation of p-values for each estimated coefficient.

In **morpheus**, this procedure can be performed manually using the results from `multiRun()`:

```
R> pval <- matrix(nrow = 2, ncol = 2)
R> for (j in 1:2) {
+    for (k in 1:2) {
+      estimates <- sapply(mr2[[1]], function(m) m[j, k])
+      stat_test <- mean(estimates)^2 / var(estimates)
+      pval[j, k] <- 1 - pchisq(stat_test, df = 1)
+    }
+  }
R> pval
```

For convenience, this logic is encapsulated in the `pvalue()` function, which automates the extraction and testing process for the user.

4. Simulations and numerical results

In this section, we evaluate the performance of the **morpheus** estimation framework through three numerical experiments of increasing complexity. For each experiment, the estimated coefficients are sorted according their true values to facilitate visual comparison. In the following figures, the solid black lines represent the true parameter values, while the dotted lines indicate the average estimates over 100 independent Monte Carlo runs. The dashed lines represent the 95% confidence intervals.

Each figure consists of four panels corresponding to increasing sample sizes: $n = 5,000$ (top-left), $n = 10,000$ (top-right), $n = 50,000$ (bottom-left), and $n = 100,000$ (bottom-right).

Experiment 1: Two-dimensional case We consider a mixture of $K = 2$ components in $\mathbb{R}^2$ with mixing proportions $\boldsymbol{\omega} = (0.35, 0.65)$ and intercepts $\mathbf{b} = (-0.3, 0.3)$. The regression matrix is defined as:

$$\boldsymbol{\beta}^\top = \begin{pmatrix} 1.5 & -1 \\ 1.0 & 1.0 \end{pmatrix}. \tag{9}$$

The results in Figure 4 illustrate the consistency of the estimators, with the variance significantly decreasing as n increases.

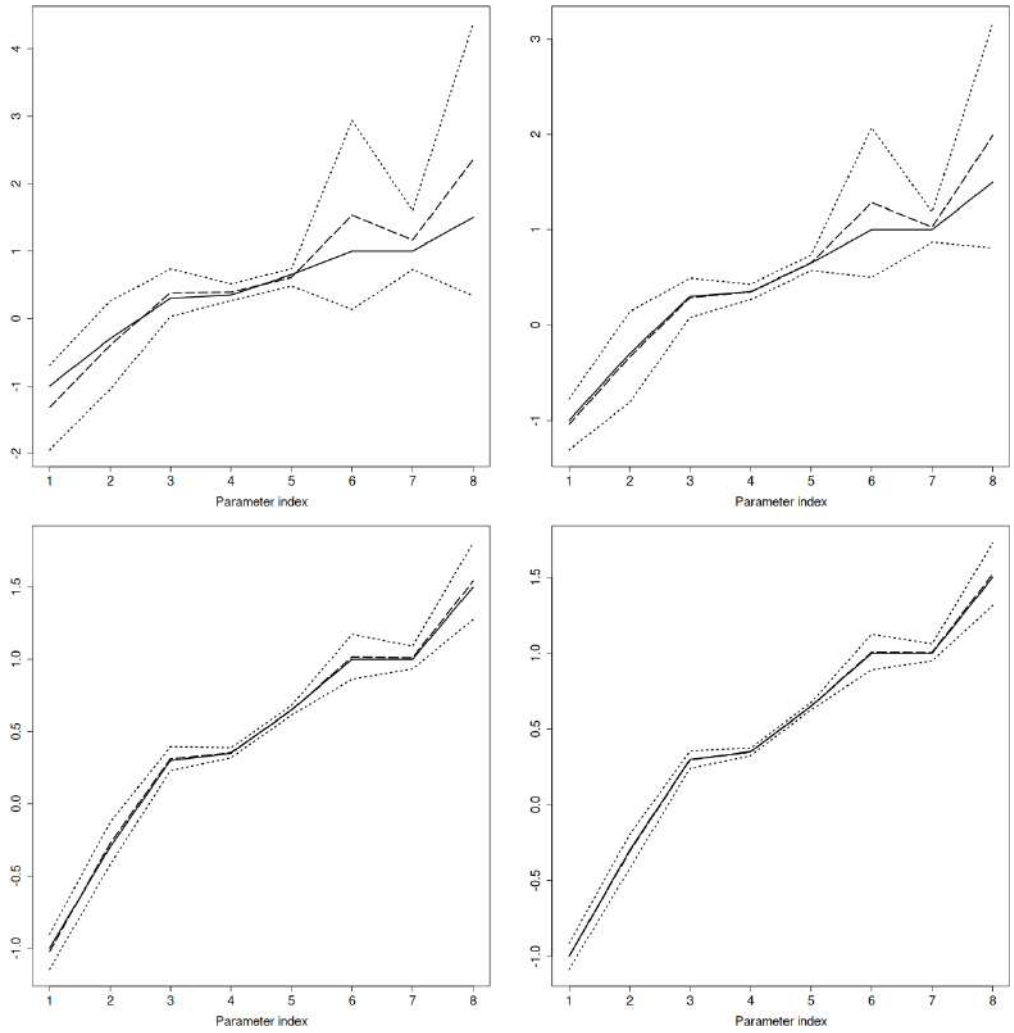

Figure 4: Experiment 1 ($d = 2, K = 2$): True parameter values vs. average estimations and 95% confidence intervals across 100 runs.

Experiment 2: Five-dimensional case The second experiment involves $d = 5$ covariates and $K = 2$ components, with identical proportions and intercepts to Experiment 1. The regression coefficients are given by:

$$\boldsymbol{\beta}^\top = \begin{pmatrix} 0.8 & 2.5 \\ 2.0 & -1.5 \\ 2.5 & -3.7 \\ -1.2 & 0.0 \\ 0.5 & 1.2 \end{pmatrix}. \tag{10}$$

As shown in Figure 5, the **morpheus** procedure maintains stability even as the dimensionality of the covariate space increases.

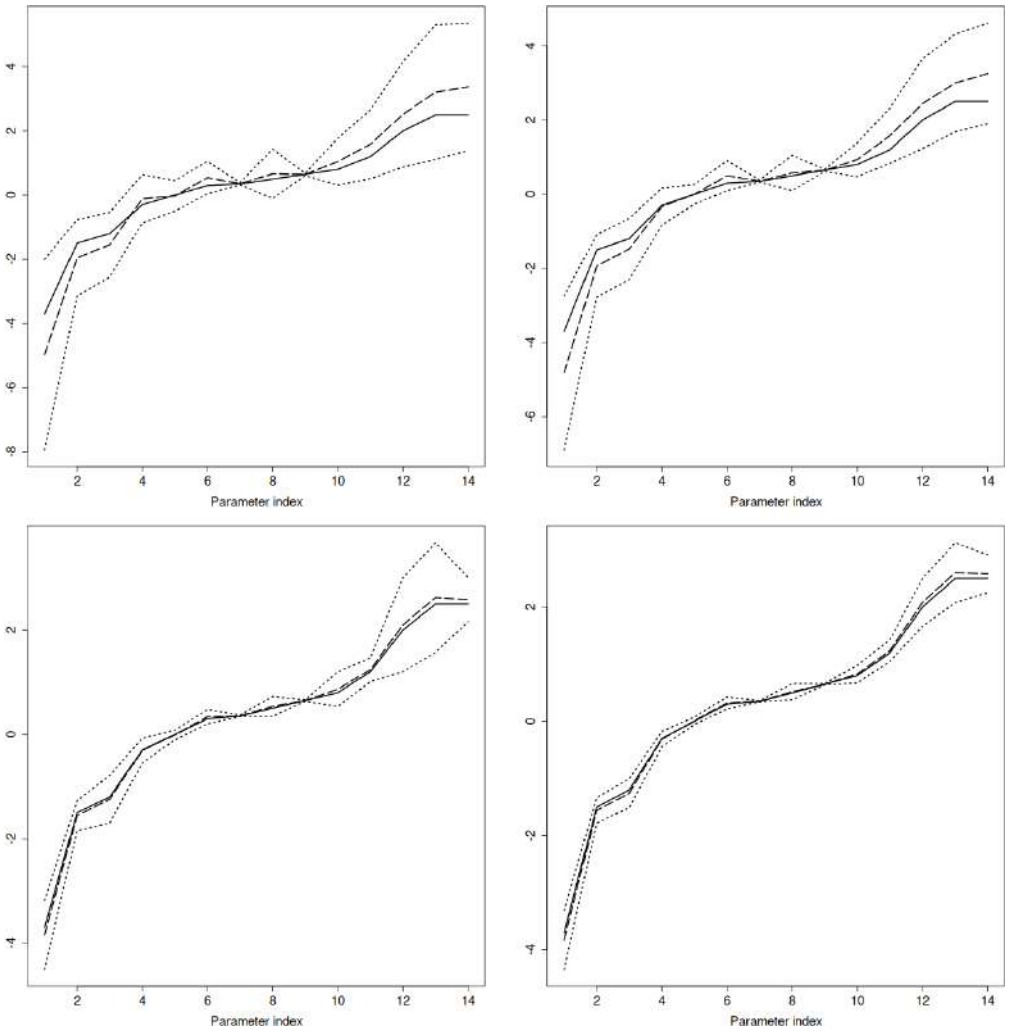

Figure 5: Experiment 2 ($d = 5, K = 2$): True parameter values vs. average estimations and 95% confidence intervals.

Experiment 3: Ten-dimensional case Finally, we test the robust initialization in a higher-dimensional setting with $d = 10$ and $K = 4$ components. The proportions are balanced ($\omega_j = 0.25$), and the intercepts are $\mathbf{b} = (-0.25, 0.25, -0.25, 0.25)$. The regression matrix is:

$$\boldsymbol{\beta}^\top = \begin{pmatrix} 1 & 2 & -1 & -1 \\ 2 & -3 & 2 & 1 \\ -1 & 0 & -3 & 3 \\ 0 & 1 & 0 & -1 \\ 3 & 0 & 1 & 0 \\ 4 & -1 & 0 & 0 \\ -1 & -4 & -1 & 2 \\ -3 & 3 & -4 & 0 \\ 0 & 2 & 3 & 1 \\ 1 & -1 & 2 & -1 \end{pmatrix}. \tag{11}$$

Figure 6 demonstrates that the spectral initialization effectively places the optimizer in the correct region of the parameter space, avoiding the spurious local maxima that often plague traditional EM-based methods in high-dimensional mixture models.

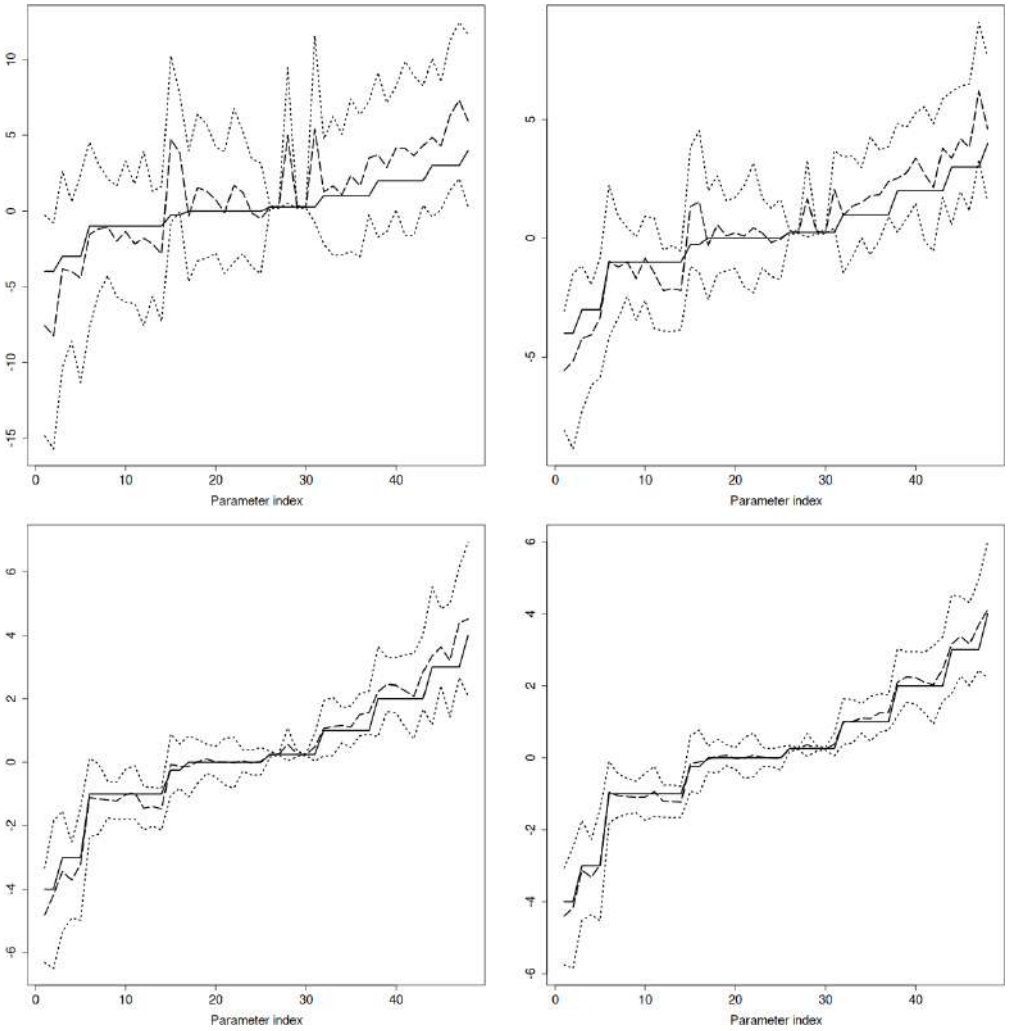

Figure 6: Experiment 3 ($d = 10, K = 4$): True parameter values vs. average estimations and 95% confidence intervals.

5. Summary and discussion

This paper introduced the **morpheus** package for estimating binary regression mixture models using a novel combination of spectral moment methods and weighted least squares optimization. By leveraging tensor-based initialization, the proposed approach overcomes one of the main limitations of classical likelihood-based procedures, namely their sensitivity to local maxima.

Simulation results highlight the strong empirical performance of the method, showing both accuracy and numerical stability even in challenging high-dimensional settings. The spec-

tral initialization plays a key role in guiding the optimization toward meaningful solutions, thereby reducing the need for multiple random restarts.

Overall, **morpheus** provides an efficient and practical alternative to existing approaches for mixture regression models. Future developments may include extensions to multiclass responses, more general latent variable structures, and scalability improvements for large-scale datasets.

Computational details

The results in this paper were obtained using R 3.4.1 with the **MASS** 7.3.47 package. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

Acknowledgments

The authors would like to thank Elisabeth Gassiat for her support and insightful advice during the development and implementation of the package. We also thank the Laboratoire de Mathématiques d'Orsay for providing access to their computing facilities, which greatly facilitated the simulation studies.

References

Afsari B (2008). "Sensitivity Analysis for the Problem of Matrix Joint Diagonalization." *SIAM Journal on Matrix Analysis and Applications*, **30**(3), 1148–1171. doi:10.1137/060655997.

Anandkumar A, Ge R, Hsu D, Kakade SM, Telgarsky M (2014). "Tensor Decompositions for Learning Latent Variable Models." *Journal of Machine Learning Research*, **15**, 2773–2832.

Auder B, Gassiat E, Loum MA (2021). "Least Squares Moment Identification of Binary Regression Mixture Models." *Metrika: International Journal for Theoretical and Applied Statistics*, **84**(4), 561–593. doi:10.1007/s00184-020-00787-x.

Dempster AP, Laird NM, Rubin DB (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society: Series B*, **39**(1), 1–38.

Diebolt J, Robert CP (1994). "Estimation of Finite Mixture Distributions through Bayesian Sampling." *Journal of the Royal Statistical Society: Series B*, **56**(2), 363–375.

Fruhwirth-Schnatter S, Celeux G, Robert CP (2019). *Handbook of mixture analysis*. Chapman and Hall CRC handbooks of modern statistical methods. CRC Press.

Frühwirth-Schnatter S (2006). *Finite Markov Mixture Modeling*, pp. 301–318. Springer New York, New York, NY. ISBN 978-0-387-35768-3. doi:10.1007/978-0-387-35768-3_10. URL https://doi.org/10.1007/978-0-387-35768-3_10.

Gruen B, Leisch F (2025). *flexmix: Flexible Mixture Modeling*. doi:10.32614/CRAN.package.flexmix. R package version 2.3-20, URL https://CRAN.R-project.org/package=flexmix.

Hansen LP (1982). "Large Sample Properties of Generalized Method of Moments Estimators." *Econometrica*, **50**(4), 1029–1054. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1912775.

Wedel M, DeSarbo WS (1995). "A mixture likelihood approach for generalized linear models." *Journal of Classification*, **12**(1), 21–55. doi:10.1007/BF01202266. URL https://doi.org/10.1007/BF01202266.

Wu CFJ (1983). "On the Convergence Properties of the EM Algorithm." *The Annals of Statistics*, **11**(1), 95–103.

Affiliation:

Mor Absa LOUM
Unité de Formation et de Recherche
Sciences de l'Ingénieur
Université Iba Der THIAM de Thiès
Thiès, Sénégal
E-mail: morabsa.loum@univ-thies.sn
Benjamin AUDER
Université Paris-Saclay, CNRS
Laboratoire de Mathématiques d'Orsay
91405 Orsay, France
E-mail: benjamin.auder@universite-paris-saclay.fr