DATA 487 Project:
An Investigation Into the Robustness of Semi-supervised Model-based Clustering Using Binary Data With Incorrect Labels

Jessica Sam
School of Mathematics and Statistics
Victoria University of Wellington, New Zealand

Abstract

Semi-supervised instead of unsupervised clustering methods, can be helpful since prior knowledge of cluster assignments for a subset of data, can inform better clustering for data with unknown assignments. Using finite mixture models with the Expectation-Maximisation algorithm is a way to cluster data, so statistical inference can be used to evaluate clustering performance. However, for many semi-supervised clustering methods, labels are typically assumed to be correct, so it is often unknown how well these clustering methods perform when data may be incorrectly labelled. This research project proposes to explore the impact of incorrectly labelled data on semi-supervised clustering using finite mixture models. Simulation studies using binary response data will be carried out, with the behaviour of parameter estimates analysed under different data scenarios.

Contents

1 Introduction

Clustering is a common, typically unsupervised technique to divide a set of observations into groups, or clusters, with the concept that observations within the same cluster share more similarity compared to observations from two different clusters. Clustering is used for many practical applications such as finance, technology, healthcare and even marketing. An example in marketing is customer segmentation, where customers are grouped by similar traits and behaviour to create product recommendations tailored to customer groupings (Madhulatha, 2012).

Unsupervised clustering groups data that are entirely unlabelled, with no prior information about the true cluster assignments. This differs from semi-supervised clustering, where there exists some proportion of labelled data points that guide the clustering of unlabelled data. Semi-supervised clustering often results in improved performance due to the addition of labelled data that inform the model (Cai et al., 2023).

However, in many semi-supervised clustering methods, the labels are typically assumed to be known with full certainty. It is often unrealistic to expect labels to be certain, as various labelling methods can result in incorrect or unreliable labels to some degree of confidence (Antoine & Labroche, 2018).

Data used in clustering problems tend to be expressed as a matrix, an array of responses arranged in rows and columns. A binary data matrix has entries encoded as a value of 1 or 0.

The objective of this research project is to investigate, through simulation, the robustness of semi-supervised model-based clustering when all three situations below simultaneously occur:

1. When a matrix of binary data is used for row clustering;
2. When data points may be incorrectly labelled;
3. When clustering is performed using the model-based finite-mixtures method (McLachlan & Peel, 2000).

1.1 Literature Review

To understand how using incorrect labels will affect semi-supervised clustering for binary data, we need to explain the methods required for this research.

Finite mixtures, as detailed in McLachlan & Peel (2000), are one type of model used for model-based clustering. The overall idea of mixture models has been used for a long time, Pearson (1894) being one of the commonly cited sources for its origin. The idea of finite mixture models is that data are drawn from a mixture of clusters, where each cluster is associated with its own probability density function.

Maximum Likelihood (ML) estimation, a fundamental statistical estimation method proposed by Fisher (1922), estimates parameter values for a model, such that the selected estimates maximise the probability of observing the data.

Dempster et al. (1977) proposed the Expectation-Maximisation (EM) algorithm, an algorithm which estimates model parameters via ML when there is missing data. This is an iterative algorithm consisting of two steps, the Expectation step (E-step) and the Maximisation step (M-step), which alternate until convergence. The E-step is where expectations are taken over all the incomplete data, using the parameter estimates at the current state, to provide estimates of the missing data. The M-step is where the completed data, being the original data and latent estimates of missing data, are used to maximise the completed data likelihood to provide new parameter estimates. This algorithm is widely applicable to a range of different data scenarios, while maintaining simplicity.

The EM algorithm is good for fitting finite mixture models, as in this case, the cluster memberships are treated as the missing data, and latent variables are estimated to indicate the probability of membership to clusters.

Pledger & Arnold (2014) suggest a group of likelihood-based models for clustering binary or count data in the structure of a matrix, based on Bernoulli or Poisson mixtures. This provides a starting point for the model in this project as a general unsupervised clustering approach for binary data matrices.

Semi-supervised model-based clustering has, in the last number of years, received more interest. Cui et al. (2024) introduced a novel semi-supervised model-based clustering method for ordinal data. This method uses the proportional odds model as the model structure for the ordinal data, and the clustering model is a finite mixture model fitted using the EM algorithm. Their paper demonstrates, through simulation, that the technique can effectively cluster partially labelled data with ordinal responses, given the labels are all correct. This paper provides a more complex model than what we aim to use. Ordinal data with only two levels of response can simplify to binary data, resulting in the use of a logistic binary model instead of a proportional odds model.

Recent studies also exist that investigate how noisy, incorrect labels or pairwise annotations impact semi-supervised clustering. Several of these papers focus on developing new methods that are robust to such data issues. These studies tend to show reasonable performance, each using methods of their respective papers, but tend to focus on distance-based or constraint-based clustering methods, rather than model-based clustering methods. Overall, they show that the addition of labels is still a net positive effect on performance, but this is dependent on the clustering methods used, and how the noisy the labels are.

Gan et al. (2018) propose a method for safe semi-supervised clustering, where the focus is around ensuring model performance is robust to data quality issues from prior knowledge. This method uses a local homogeneous graph to model the relationship between labelled data and unlabelled data, to construct a regularisation term which accounts for the riskiness of labelled samples. The datasets the algorithm is evaluated on, each consist of 20% labelled data, with the rest of the dataset unlabelled. They change the proportion of labelled data that is incorrect from 0 to 30%, using 5% increments. They show that their algorithm, even if the proportion of incorrectly labelled data reaches 30%, still outperforms the respective unsupervised and semi-supervised clustering methods.

Antoine & Labroche (2018) explores the effect of using incorrect labels on variants of semi-supervised fuzzy clustering (c-clustering) and proposes new approaches which are tailored to handle the uncertainty in labelled data. Their study shows using incorrect or uncertain labelling with fuzzy clustering can still provide a reasonable clustering performance, despite the uncertainty. Despite this reasonable performance, noisy labels generated less accurate solutions compared to using correct labels. They also note that if the data has noisy labels, reducing the label certainty for the semi-supervised clustering allows an improvement in accuracy, compared to using unsupervised clustering alone.

Gribel et al. (2022) explores the effect of using inaccurate pairwise annotations for semi-supervised constrained clustering. They state that class labels may be difficult to obtain and instead, relational information can be used, like "must-links" to indicate a pair of samples should be in the same cluster, and "cannot-links" that indicate a pair of samples should not be in the same cluster. Using a maximum likelihood approach, they propose a generative model that models data as coming from Gaussian distributions and has such link relations generated by stochastic block models. They show although there may be inaccurate annotations or a small quantity of annotations, using the links still improves clustering performance compared to excluding the relational information.

An exploration into previous literature indicates a lack of investigation around the implications of using incorrect data within semi-supervised clustering methods in general. The performances of some semi-supervised clustering methods with incorrect data have been evaluated, but most of these methods are not model-based, meaning that statistical inference cannot be made with them. This justifies looking at the robustness of using finite mixture modelling with a matrix of binary responses. This will serve as a binary-data complement to the work of Cui et al. (2024) on ordinal models.

2 Methodology

2.1 Data Structure and Binary Model

We now define the structure of the binary response data matrix used for finite-mixture clustering and the data model.

We can consider a questionnaire consisting of n respondents who answer p questions, where each question is answered with a binary "Yes" or "No" response. Each respondent will answer p questions, resulting in a total of $n \times p$ responses. The responses to this questionnaire can be represented in the structure of an $n \times p$ matrix denoted by $\boldsymbol{Y}$, where n is the number of rows and p is the number of columns. Each observation $y_{ij} \in \{0, 1\}$ in the matrix is some respondent's answer to a question, with a "Yes" response encoded with 1 and a "No" response encoded as 0, with subscripts taking values $i = 1, 2, ..., n$ and $j = 1, 2, ..., p$. Table 1 illustrates what a questionnaire matrix might look like.

Table 1: Example questionnaire matrix with n respondents and p questions with binary responses.

Respondent	Q_1	Q_2	Q_3	$\cdots$	Q_p
1	1	0	1	$\cdots$	1
2	0	1	1	$\cdots$	1
3	0	0	1	$\cdots$	0
4	1	0	0	$\cdots$	1
5	1	1	0	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	0
n	0	1	1	$\cdots$	1

With this data matrix, before accounting for any clustering patterns, a simple logistic binary model can be made by considering there will be distinct response patterns. In the example of a questionnaire, answering "Yes" to a question depends on which respondent answers and which question is asked. These can be known as row effects, represented by $\{\alpha_i\}$, with $i = 1, 2, ..., n$, and column effects, represented by $\{\beta_j\}$, with $j = 1, 2, ..., p$. This models the log odds of a positive response, or a success, as an overall mean μ with some row effect α_i and some column effect β_j, as follows:

$$\text{logit}[\theta_{ij}] = \mu + \alpha_i + \beta_j, \tag{2.1}$$

where θ_{ij} represents $P(y_{ij} = 1)$, the probability of success, or responding with a 1. Here, sum to zero constraints are used, $\sum_{i=1}^{n} \alpha_i = 0$ and $\sum_{j=1}^{p} \beta_j = 0$.

2.2 Row Clustering Binary Model

The model in (2.1) accounts for each individual row and column effect, as expressed by the α and β parameters. With row clustering, the number of α parameters needed can be reduced by assuming that each row can belong to one of R row clusters. Instead of the row effects being expressed as $\{\alpha_i\}$, where $i = 1, 2, ..., n$, they can instead be expressed as row cluster effects $\{\alpha_r\}$ with $r = 1, 2, ..., R$.

The row clustering model with individual column effects becomes:

$$\begin{gathered} \operatorname{logit}[\theta_{rj} | i \in r)] = \mu + \alpha_r + \beta_j \\ i = 1, 2, ..., n, \quad j = 1, 2, ..., p, \quad r = 1, 2, ..., R. \end{gathered} \tag{2.2}$$

Here, in (2.2), $i \in r$ refers to a particular row i belonging to the cluster r. θ_{rj} represents the probability of success for a particular row $i \in r$ and column j.

2.3 Unsupervised Row Clustering

The exact cluster assignments for the rows in unsupervised clustering are unknown, so this is treated as missing information to compute when using the EM algorithm.

The unknown proportions of rows that belong to each cluster can be defined as $\{\pi_r\}$ where $r = 1, 2, ..., R$, with the constraint $\sum_{r=1}^{R} \pi_r = 1$.

The assumption is made that the rows are independent, given their cluster memberships, and that for every row, the p columns come from independent trials, where an observation y_{ij} has the probability θ_{ij} of success for $i \in r$. With this assumption, the likelihoods for the model in (2.2) can be constructed with the knowledge that values from the matrix come from Bernoulli distributions, with probabilities of success $\{\theta_{rj}\}$.

2.3.1 Likelihoods

The overall likelihood for the non-clustered binary matrix model can be expressed using the probability density function for the Bernoulli distribution:

$$L(\boldsymbol{\theta};\boldsymbol{y}) = \prod_{i=1}^{n}\prod_{j=1}^{p}\theta_{ij}^{y_{ij}}(1-\theta_{ij})^{1-y_{ij}}. \tag{2.3}$$

The typically easier-to-compute log-likelihood for the binary model is expressed as:

$$\ell(\boldsymbol{\theta};\boldsymbol{y}) = \sum_{i=1}^{n}\sum_{j=1}^{p}\left[y_{ij}\log\theta_{ij} + (1-y_{ij})\log(1-\theta_{ij})\right]. \tag{2.4}$$

With row clustering, we say that the rows can be clustered in R groups, where $\pi_1, \pi_2, ...\pi_R$ are the probabilities of belonging to each of the R clusters, with $\sum_{r=1}^{R}\pi_r = 1$.

This changes the overall data likelihood and log-likelihood from (2.3) and (2.4) to:

$$L(\boldsymbol{\Omega},\boldsymbol{\pi};\boldsymbol{y}) = \prod_{i=1}^{n} L_i = \prod_{i=1}^{n}\left[\sum_{r=1}^{R}\left[\pi_r\prod_{j=1}^{p}\theta_{rj}^{y_{ij}}(1-\theta_{rj})^{1-y_{ij}}\right]\right] \tag{2.5}$$

and

$$\ell(\boldsymbol{\Omega},\boldsymbol{\pi};\boldsymbol{y}) = \sum_{i=1}^{n}\log\left[\sum_{r=1}^{R}\left[\pi_r\prod_{j=1}^{p}\theta_{rj}^{y_{ij}}(1-\theta_{rj})^{1-y_{ij}}\right]\right] \tag{2.6}$$

where $\boldsymbol{\Omega} = \{\mu, \{\alpha_r\}, \{\beta_j\}\}$ is the non-redundant parameter vector of (2.2) that we want to obtain.

The equation (2.6) is still not convenient to compute, so a latent variable is introduced to simplify the log-likelihood, under the assumption that we have complete knowledge about the cluster assignments. The missing information about the cluster assignments can be written as an $n \times R$ matrix $\boldsymbol{Z}$, with the latent variable being $z_{ir} = 1$ if a particular row i belongs to the cluster r, and $z_{ir} = 0$ otherwise.

With this additional information from z_{ir}, another type of likelihood, called the complete-data likelihoods, can be constructed.

The complete-data likelihood incorporating the latent variables can be written as L_C, changing (2.5) to:

$$L_C(\boldsymbol{\Omega},\boldsymbol{\pi};\boldsymbol{y},\boldsymbol{z}) = \prod_{i=1}^{n}\prod_{r=1}^{R}\left[\pi_r\prod_{j=1}^{p}\theta_{rj}^{y_{ij}}(1-\theta_{rj})^{1-y_{ij}}\right]^{z_{ir}} \tag{2.7}$$

with the corresponding (2.6) modified to the complete-data log-likelihood ℓ_C:

$$\ell_C(\boldsymbol{\Omega}, \boldsymbol{\pi}; \boldsymbol{y}, \boldsymbol{z}) = \sum_{i=1}^{n} \sum_{r=1}^{R} z_{ir} \log \left[\pi_r \prod_{j=1}^{p} \theta_{rj}^{y_{ij}} (1-\theta_{rj})^{1-y_{ij}} \right]. \tag{2.8}$$

Rearrangement of (2.8) gives an easier-to-compute:

$$\begin{aligned} \ell_C(\boldsymbol{\Omega}, \boldsymbol{\pi}; \boldsymbol{y}, \boldsymbol{z}) = & \sum_{i=1}^{n} \sum_{r=1}^{R} z_{ir} \log \pi_r + \\ & \sum_{i=1}^{n} \sum_{r=1}^{R} \sum_{j=1}^{p} z_{ir} \log \left[y_{ij} \log \theta_{rj} + (1-y_{ij}) \log(1-\theta_{rj}) \right]. \end{aligned} \tag{2.9}$$

2.3.2 EM Algorithm

The EM algorithm lets us find the parameter estimates: $\hat{\boldsymbol{\Omega}}$, $\{\hat{\pi}_r\}, \{\hat{z}_{ir}\}$, using the likelihood computed above. The algorithm will continue alternating between the E and M steps, until the parameters converge.

2.3.3 Expectation Step (E-Step)

The expectation of the latent variable z_{ir} is taken to estimate the missing cluster assignments and is calculated by using the current parameter estimates, $\hat{\boldsymbol{\Omega}}$ and $\{\hat{\pi}_r\}$. It is also the posterior probability of the row i being in the cluster r, given the data in row i. This is calculated by:

$$E(z_{ir}) = P(i \in r | \boldsymbol{y}_i) = \frac{P(i \in r, \boldsymbol{y}_i)}{P(\boldsymbol{y}_i)} = \frac{\pi_r \prod_{j=1}^{p} \theta_{rj}^{y_{ij}} (1-\theta_{rj})^{1-y_{ij}}}{\sum_{h=1}^{R} \pi_h \prod_{j=1}^{p} \theta_{hj}^{y_{ij}} (1-\theta_{hj})^{1-y_{ij}}} \tag{2.10}$$

where $\mathbf{y}_i = \{y_{i1}, ..., y_{ip}\}$.

2.3.4 Maximisation Step (M-step)

The complete-data log-likelihood from (2.9), using the values for z_{ir} calculated in the expectation step, is then maximised to obtain the estimates $\hat{\boldsymbol{\Omega}}$ and $\{\hat{\pi}_r\}$.

$\hat{\pi}_r$ is calculated by the equation:

$$\hat{\pi}_r = \frac{\sum_{i=1}^{n} z_{ir}}{n}. \tag{2.11}$$

Numeric optimisation of the second term of (2.9) finds the parameter estimates, $\hat{\mathbf{\Omega}}$:

$$\hat{\mathbf{\Omega}} = \underset{\mathbf{\Omega}}{\operatorname{argmax}} \left[\sum_{i=1}^{n} \sum_{r=1}^{R} \sum_{j=1}^{p} z_{ir} \log \left[y_{ij} \log \theta_{rj} + (1 - y_{ij}) \log(1 - \theta_{rj}) \right] \right], \tag{2.12}$$

where θ_{rj} is a function of $\mathbf{\Omega}$, derived from equation (2.2).

2.4 Semi-supervised Row Clustering

For the semi-supervised case where there are known cluster assignments for a subset of the data, the data matrix can be considered to have an extra column for the known cluster assignments, compared to the unsupervised case in Table 1:

Table 2: Example data matrix with n rows and p columns with binary responses and some existing cluster labels. NA represents no knowledge of cluster assignment.

Observation	1	2	$\cdots$	p	Cluster
1	1	0	$\cdots$	1	1
2	0	1	$\cdots$	1	2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
n_l	0	0	$\cdots$	0	3
$n_l + 1$	1	0	$\cdots$	1	NA
$n_l + 2$	1	1	$\cdots$	0	NA
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
$n_l + n_u$	1	0	$\cdots$	1	NA

In Table 2, n_l represents the total number of rows with known clustering labels, and n_u represents the total number of rows with unknown clustering labels.

Instead of the unsupervised case that assigns clusters to all data points, the semi-supervised case will only assign clusters to the data points where the cluster memberships are unknown. As a result, values computed in parameter estimation steps for semi-supervised clustering are not the same as the parameters estimated in the unsupervised clustering.

2.4.1 Likelihoods

The overall and complete-data likelihoods associated with the semi-supervised clustering below are obtained from Cui et al. (2024).

The overall data likelihood $L(\boldsymbol{\Omega}, \boldsymbol{\pi}; \boldsymbol{y})$ can be expressed as:

$$L(\boldsymbol{\Omega}, \boldsymbol{\pi}; \boldsymbol{y}) = \prod_{i=1}^{n_l} \prod_{j=1}^{p} \prod_{r=1}^{R} \left[\theta_{rj}^{y_{ij}} (1-\theta_{rj})^{1-y_{ij}}\right]^{I(i \in r)} \times \sum_{i=n_l+1}^{n_l+n_u} \sum_{r=1}^{R} \pi_r \prod_{j=1}^{p} \theta_{rj}^{y_{ij}} (1-\theta_{rj})^{1-y_{ij}}, \quad (2.13)$$

where $I(i \in r) = 1$ if the i^{th} observation is known to be in cluster r, and $I(i \in r) = 0$ otherwise.

The corresponding overall data log-likelihood $\ell(\boldsymbol{\Omega}, \boldsymbol{\pi}; \boldsymbol{y})$ is written as:

$$\begin{aligned} \ell(\boldsymbol{\Omega}, \boldsymbol{\pi}; \boldsymbol{y}) = & \sum_{i=1}^{n_\ell} \sum_{j=1}^{p} \sum_{r=1}^{R} I(i \in r) \left[y_{ij} \log \theta_{rj} + (1-y_{ij}) \log(1-\theta_{rj})\right] + \\ & \sum_{i=n_l+1}^{n_l+n_u} \log \left[\sum_{r=1}^{R} \pi_r \prod_{j=1}^{p} \theta_{rj}^{y_{ij}} (1-\theta_{rj})^{1-y_{ij}}\right]. \end{aligned} \quad (2.14)$$

The complete-data log-likelihood $\ell_C(\boldsymbol{\Omega}, \boldsymbol{\pi}; \boldsymbol{y}, \boldsymbol{z})$ is expressed as:

$$\begin{aligned} \ell_C(\boldsymbol{\Omega}, \boldsymbol{\pi}; \boldsymbol{y}, \boldsymbol{z}) = & \sum_{i=1}^{n_l} \sum_{r=1}^{R} \sum_{j=1}^{p} I(i \in r) \left[y_{ij} \log \theta_{rj} + (1-y_{ij}) \log(1-\theta_{rj})\right] + \\ & \left[\sum_{i=n_l+1}^{n_l+n_u} \sum_{r=1}^{R} z_{ir} \log \pi_r + \sum_{i=n_l+1}^{n_l+n_u} \sum_{j=1}^{p} \sum_{r=1}^{R} z_{ir} \left[y_{ij} \log \theta_{rj} + (1-y_{ij}) \log(1-\theta_{rj})\right]\right]. \end{aligned} \quad (2.15)$$

2.4.2 Expectation Step (E-step)

It is only necessary to calculate $E[z_{ir}]$ for the rows with unknown cluster assignments and not the rows with known cluster assignments, since the values for z_{ir} for the labelled data can be fixed for every iteration of the algorithm.

In this case, using $E[z_{ir}]$ can be rewritten as:

$$E[z_{ir}] = \begin{cases} I(i = r), & i \in \{1, ..., n_l\} \\ \frac{\pi_r \prod_{j=1}^{p} \theta_{rj}^{y_{ij}} (1-\theta_{rj})^{1-y_{ij}}}{\sum_{h=1}^{R} \pi_h \prod_{j=1}^{p} \theta_{hj}^{y_{ij}} (1-\theta_{hj})^{1-y_{ij}}}, & i \in \{n_l + 1, ..., n_l + n_u\}. \end{cases} \tag{2.16}$$

2.4.3 Maximisation Step (M-Step)

With semi-supervised clustering, the overall log-likelihood and the complete-data log-likelihood equations are the same as what was described for unsupervised clustering, except that the labelled data uses exact, known $\{z_{ir}\}$ instead of $\{\hat{z}_{ir}\}$.

With the obtained $\hat{z}_{ir}$ values computed from the expectation step, and with the exact z_{ir} values from the labelled data, the complete-data log-likelihood (2.9) is maximised, like in the unsupervised case, to obtain parameter estimates $\hat{\boldsymbol{\Omega}}$ and $\{\hat{\pi}_r\}$.

2.5 Simulations

2.5.1 Data Generation

All of the simulated datasets were created with $p = 5$ columns, $R = 3$ row clusters, but different numbers of rows: $n = (300, 1000, 3000)$. The datasets were generated with equal proportions of rows assigned to each cluster.

Table 3 lists all the different combinations of dataset size, n, proportion of data labelled, m and proportion of labelled data that are incorrect s, to run simulations. For each combination of n, m and s, 100 replicate datasets were simulated.

The labelled data in each of the datasets were selected at random and incorrect labels from the labelled subsets, were also selected at random. As a result, there may not equal proportions of labelled data that belong to each of the three clusters, despite the true cluster proportions created as equal.

Table 3: Simulation combinations of dataset size, n, proportion of data labelled m, and proportion of the $m\%$ labelled data that are incorrect, s.

n	**m (known)**	**s (wrongly labelled)**
300	10%	10%
		30%
		50%
	30%	10%
		30%
		50%
1000	10%	10%
		30%
		50%
	30%	10%
		30%
		50%
3000	10%	10%
		30%
		50%
	30%	10%
		30%
		50%

2.5.2 True Parameter Values

The true value for μ was set at 0.

True values of the $\{\alpha_r\}$ and $\{\beta_j\}$ parameters were set, following the simulation scenarios proposed by Cui (2025) as:as:

- $\{\alpha_1, \alpha_2, \alpha_3\} = \{-2.0, 0.0, 2.0\}$
- $\{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5\} = \{-2.0, -1.5, 0.3, 1.0, 2.2\}$

Both $\{\alpha_r\}$ and $\{\beta_j\}$ parameters were set using sum-to-zero constraints.

2.5.3 Initial Parameter Estimates

Initial parameter estimates $\{\pi_r^{(0)}\}$, were set as the true proportions so that $\{\hat{\pi}_r^{(0)}\} = \{0.333, 0.333, 0.333\}$. These parameters are not as essential for the evaluation of the clustering performance, so it is not as essential to use randomly assigned values.

Initial parameter estimates $\hat{\boldsymbol{\Omega}}^{(0)}$ were generated to be close to the true parameter values. For each parameter in $\boldsymbol{\Omega}$, the initial estimates were generated by randomly sampling from a uniform distribution, where each distribution has a lower limit equal to the true parameter value minus one, and upper limit equal to the true parameter value plus one.

2.6 Clustering Performance Measures

After the model fitting procedure is done for $H = 100$ replicates of each data scenario, the mean and standard deviations for each parameter estimate in $\hat{\boldsymbol{\Omega}}$ and $\{\hat{\alpha}_r\}$ were calculated from the replicates.

These are calculated as follows:

$$\text{Mean}(\hat{\omega}) = \frac{1}{H}\sum_{h=1}^{H} \hat{\omega}^{(h)}, \tag{2.17}$$

$$\text{s.d}(\hat{\omega}) = \text{s.d}\left(\hat{\omega}^{(h=1)}, \hat{\omega}^{(h=2)}, ..., \hat{\omega}^{(h=100)}\right), \tag{2.18}$$

where $\hat{\omega}$ represents some component of the parameter vector $\hat{\boldsymbol{\Omega}}$ or the set of $\{\hat{\pi}_r\}$, such as $\hat{\mu}$ or $\hat{\pi}_1$.

Mean parameter estimates and accuracies were used to judge how good the clustering performance is for each data scenario. The mean parameter estimates from $\hat{\boldsymbol{\Omega}}$ are compared to the true values for the simulated datasets and predicted labels are compared to the true labels to calculate clustering accuracy.

2.7 Statistical Software

All analyses in the study were conducted using the **R** software (version 4.5.1) (R Core Team, 2025). The `optim()` function, using the quasi-Newton method (L-BFGS-B), was chosen to maximise the terms from the complete-data log-likelihood in the M-step.

3 Results

Table 4: Mean and Standard Deviation of Parameter Estimates

Scenario	**Param.**	**Truth**	**n = 300**		**n = 1000**		**n = 3000**	
			mean	s.d	mean	s.d	mean	s.d
Scenario 1: $(m = 10\%, s = 10\%)$	μ	0.000	-0.330	0.030	-0.337	0.016		
	α_1	-2.000	-1.520	0.030	-1.512	0.016		
	α_2	0.000	-0.270	0.030	-0.270	0.016		
	β_1	-2.000	-1.860	0.160	-1.840	0.079		
	β_2	-1.500	-1.380	0.170	-1.387	0.096		
	β_3	0.300	0.230	0.170	0.253	0.092		
	β_4	1.000	0.900	0.220	0.904	0.105		
	π_1	0.333	0.240	0.030	0.244	0.015		
	π_2	0.333	0.290	0.050	0.295	0.022		
Scenario 2: $(m = 10\%, s = 30\%)$	μ	0.000	-0.358	0.031	-0.360	0.018		
	α_1	-2.000	-1.490	0.031	-1.488	0.018		
	α_2	0.000	-0.249	0.031	-0.247	0.018		
	β_1	-2.000	-1.814	0.157	-1.797	0.081		
	β_2	-1.500	-1.339	0.160	-1.348	0.090		
	β_3	0.300	0.237	0.165	0.256	0.086		
	β_4	1.000	0.881	0.216	0.887	0.106		
	π_1	0.333	0.243	0.031	0.245	0.017		
	π_2	0.333	0.289	0.043	0.294	0.025		
Scenario 3: $(m = 10\%, s = 50\%)$	μ	0.000	-0.379	0.031	-0.377	0.021		
	α_1	-2.000	-1.470	0.031	-1.472	0.021		
	α_2	0.000	-0.228	0.031	-0.230	0.021		
	β_1	-2.000	-1.773	0.152	-1.774	0.083		
	β_2	-1.500	-1.306	0.156	-1.324	0.094		
	β_3	0.300	0.243	0.161	0.263	0.094		
	β_4	1.000	0.873	0.195	0.877	0.104		
	π_1	0.333	0.237	0.032	0.244	0.016		
	π_2	0.333	0.296	0.045	0.297	0.023		

Table 5: Mean and Standard Deviation of Parameter Estimates

Scenario	**Param.**	**Truth**	**n = 300**		**n = 1000**		**n = 3000**	
			mean	s.d	mean	s.d	mean	s.d
	μ	0.000	-0.317	0.040	-0.326	0.022		
	α_1	-2.000	-1.531	0.040	-1.522	0.022		
	α_2	0.000	-0.290	0.040	-0.281	0.022		
Scenario 4:	β_1	-2.000	-1.831	0.166	-1.805	0.078		
	β_2	-1.500	-1.351	0.183	-1.346	0.104		
$(m = 30\%, s = 10\%)$	β_3	0.300	0.240	0.188	0.267	0.103		
	β_4	1.000	0.897	0.220	0.903	0.119		
	π_1	0.333	0.268	0.024	0.268	0.013		
	π_2	0.333	0.309	0.031	0.311	0.019		
	μ	0.000	-0.382	0.039	-0.389	0.023		
	α_1	-2.000	-1.467	0.039	-1.460	0.023		
	α_2	0.000	-0.225	0.039	-0.218	0.023		
Scenario 5:	β_1	-2.000	-1.711	0.160	-1.704	0.085		
	β_2	-1.500	-1.245	0.166	-1.265	0.099		
$(m = 30\%, s = 30\%)$	β_3	0.300	0.252	0.172	0.264	0.091		
	β_4	1.000	0.858	0.215	0.855	0.104		
	π_1	0.333	0.263	0.027	0.270	0.014		
	π_2	0.333	0.316	0.034	0.308	0.019		
	μ	0.000	-0.433	0.044	-0.443	0.024		
	α_1	-2.000	-1.415	0.044	-1.406	0.024		
	α_2	0.000	-0.174	0.044	-0.164	0.024		
Scenario 6:	β_1	-2.000	-1.644	0.160	-1.631	0.078		
	β_2	-1.500	-1.196	0.177	-1.191	0.093		
$(m = 30\%, s = 50\%)$	β_3	0.300	0.260	0.166	0.270	0.087		
	β_4	1.000	0.821	0.185	0.808	0.100		
	π_1	0.333	0.267	0.027	0.272	0.017		
	π_2	0.333	0.314	0.033	0.307	0.023		

Table 6: Mean and Standard Deviation of Clustering Accuracy

Scenario	**n = 300**		**n = 1000**		**n = 3000**	
	mean	s.d	mean	s.d	mean	s.d
Scenario 1: $(m = 10\%, s = 10\%)$	0.672	0.052	0.692	0.023		
Scenario 2: $(m = 10\%, s = 30\%)$	0.677	0.049	0.690	0.027		
Scenario 3: $(m = 10\%, s = 50\%)$	0.673	0.049	0.690	0.026		
Scenario 4: $(m = 30\%, s = 10\%)$	0.686	0.041	0.693	0.016		
Scenario 5: $(m = 30\%, s = 30\%)$	0.682	0.047	0.693	0.016		
Scenario 6: $(m = 30\%, s = 50\%)$	0.688	0.039	0.693	0.016		

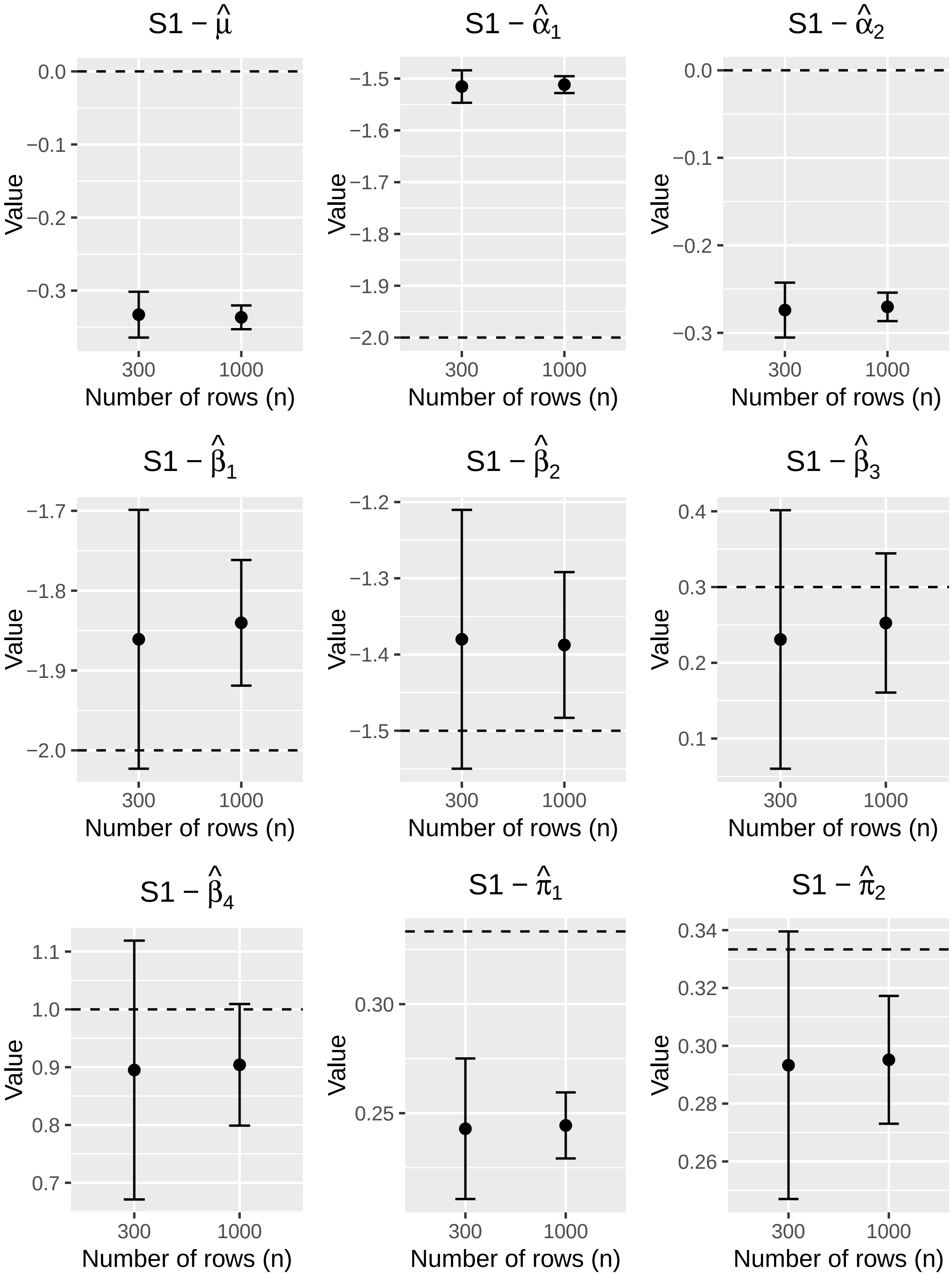

Figure 1: Results for m = 10%, s = 10%.

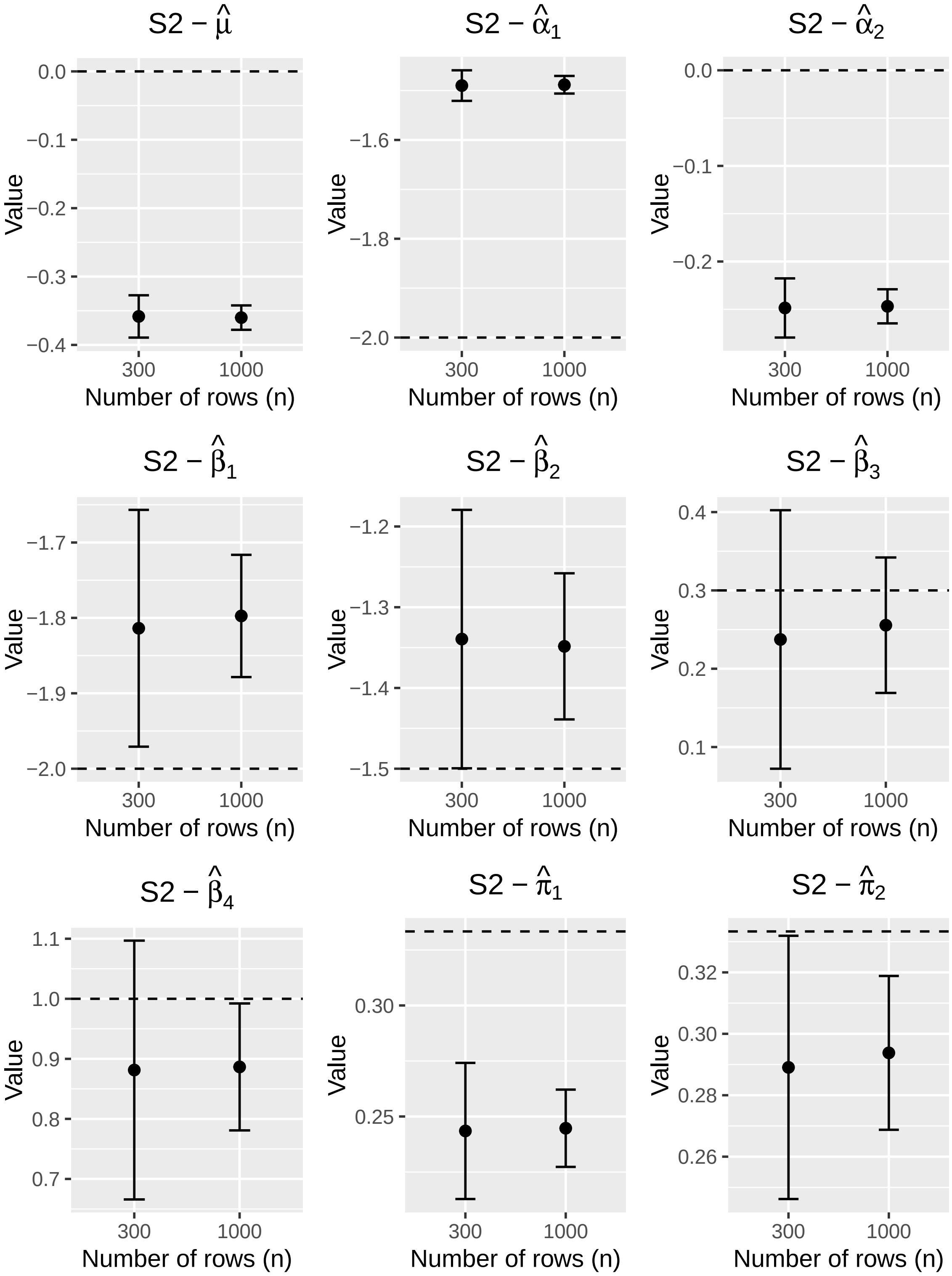

Figure 2: Results for m = 10%, s = 30%.

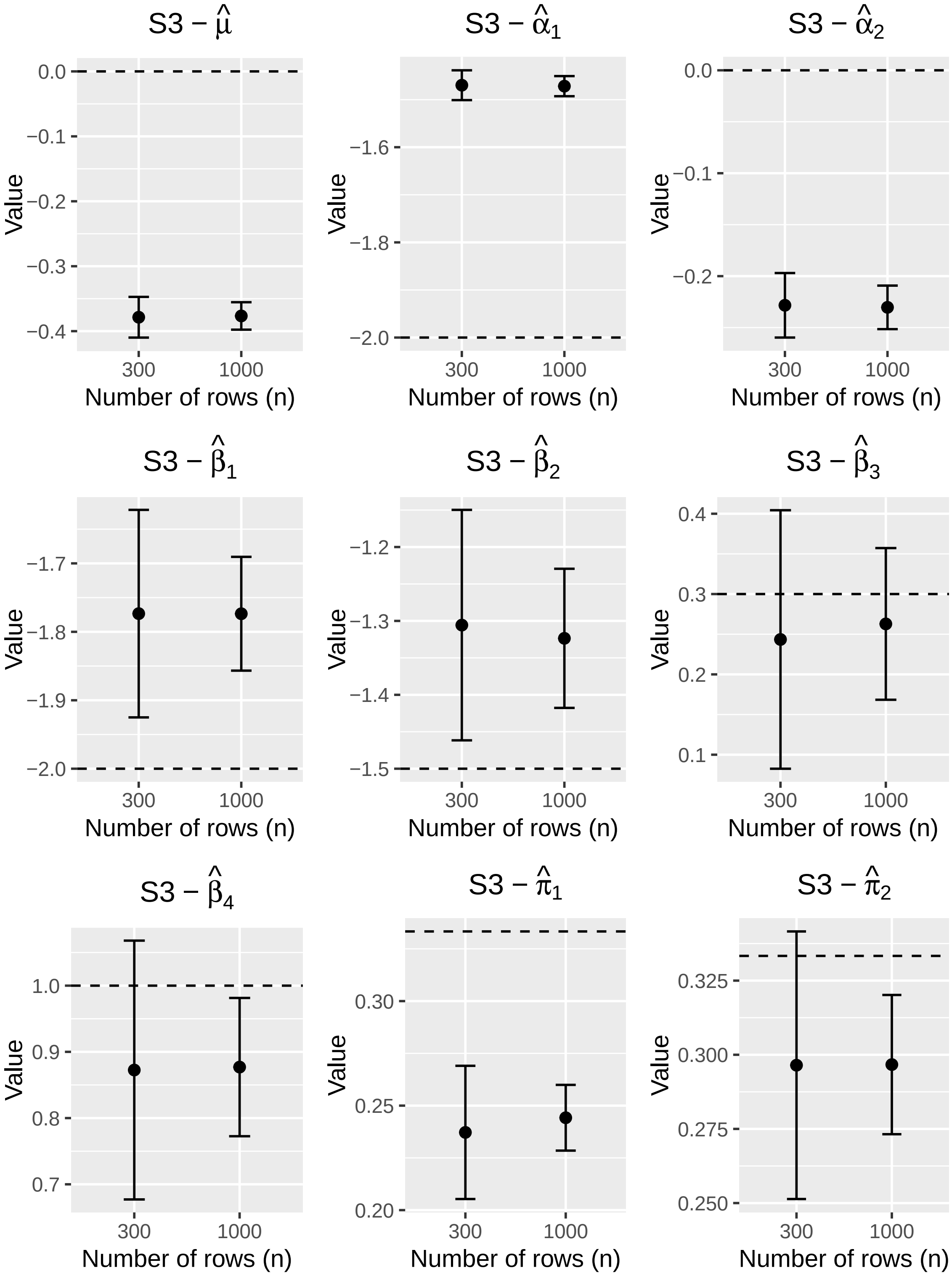

Figure 3: Results for m = 10%, s = 50%.

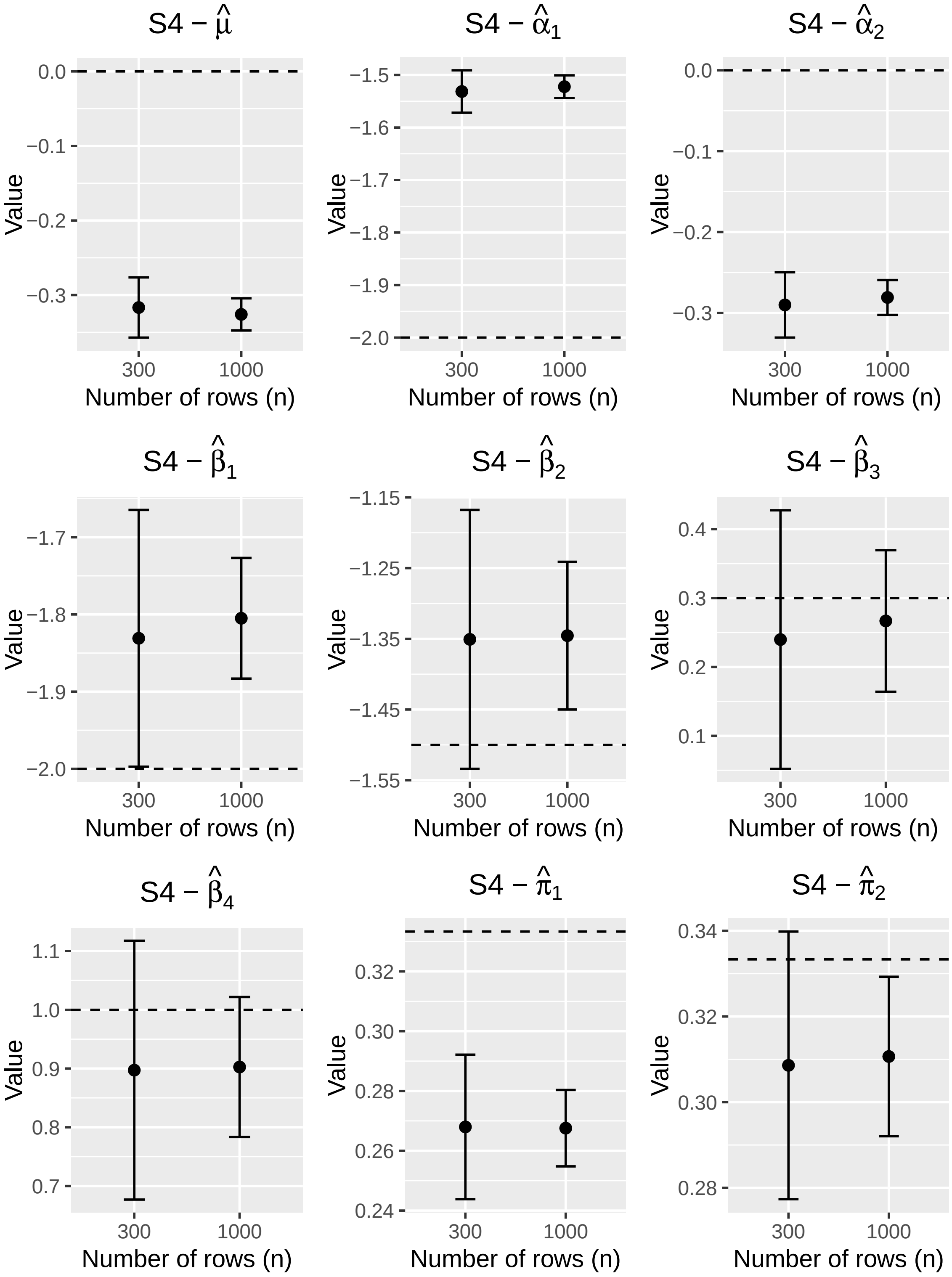

Figure 4: Results for m = 30%, s = 10%.

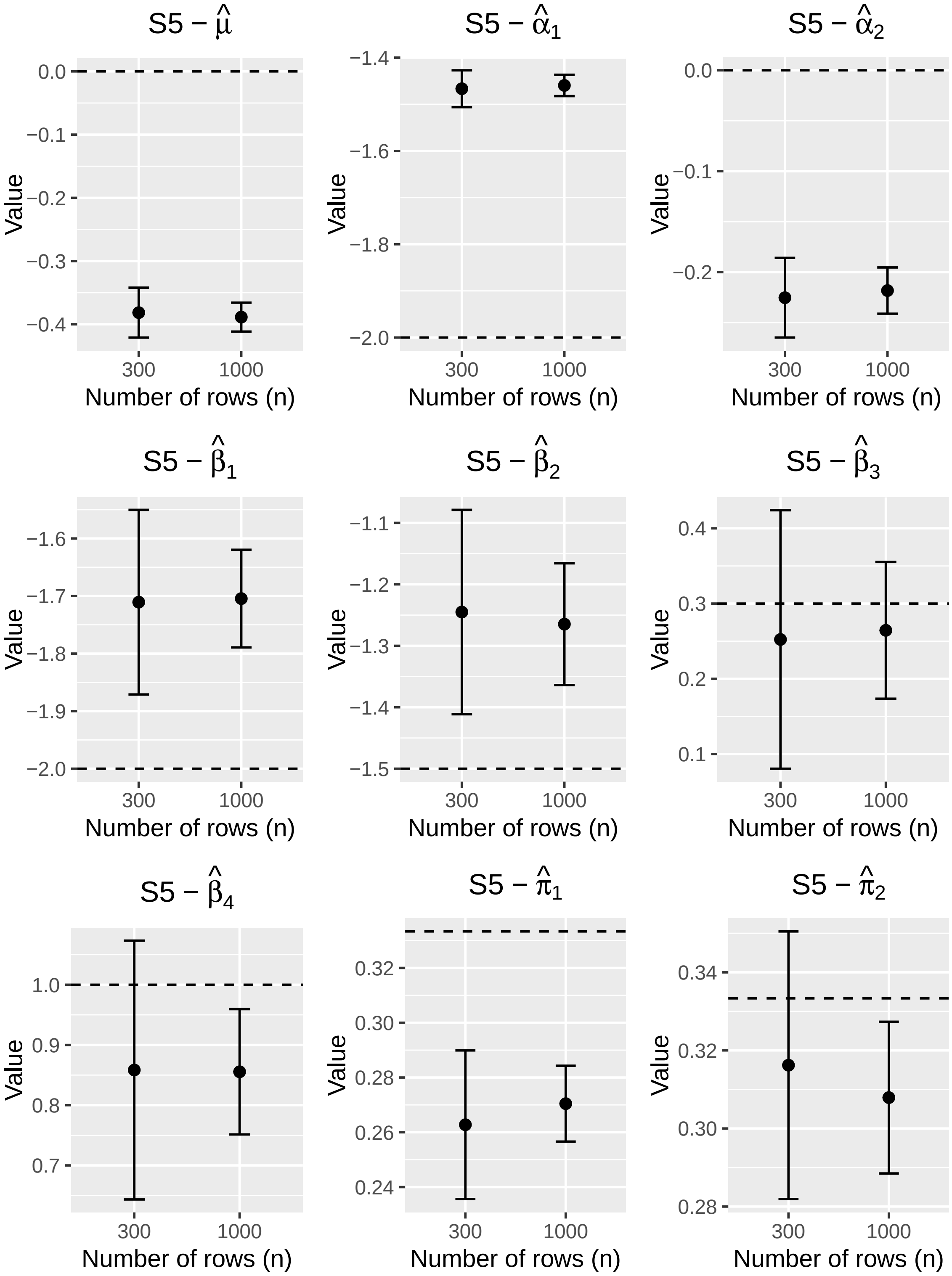

Figure 5: Results for m = 30%, s = 30%.

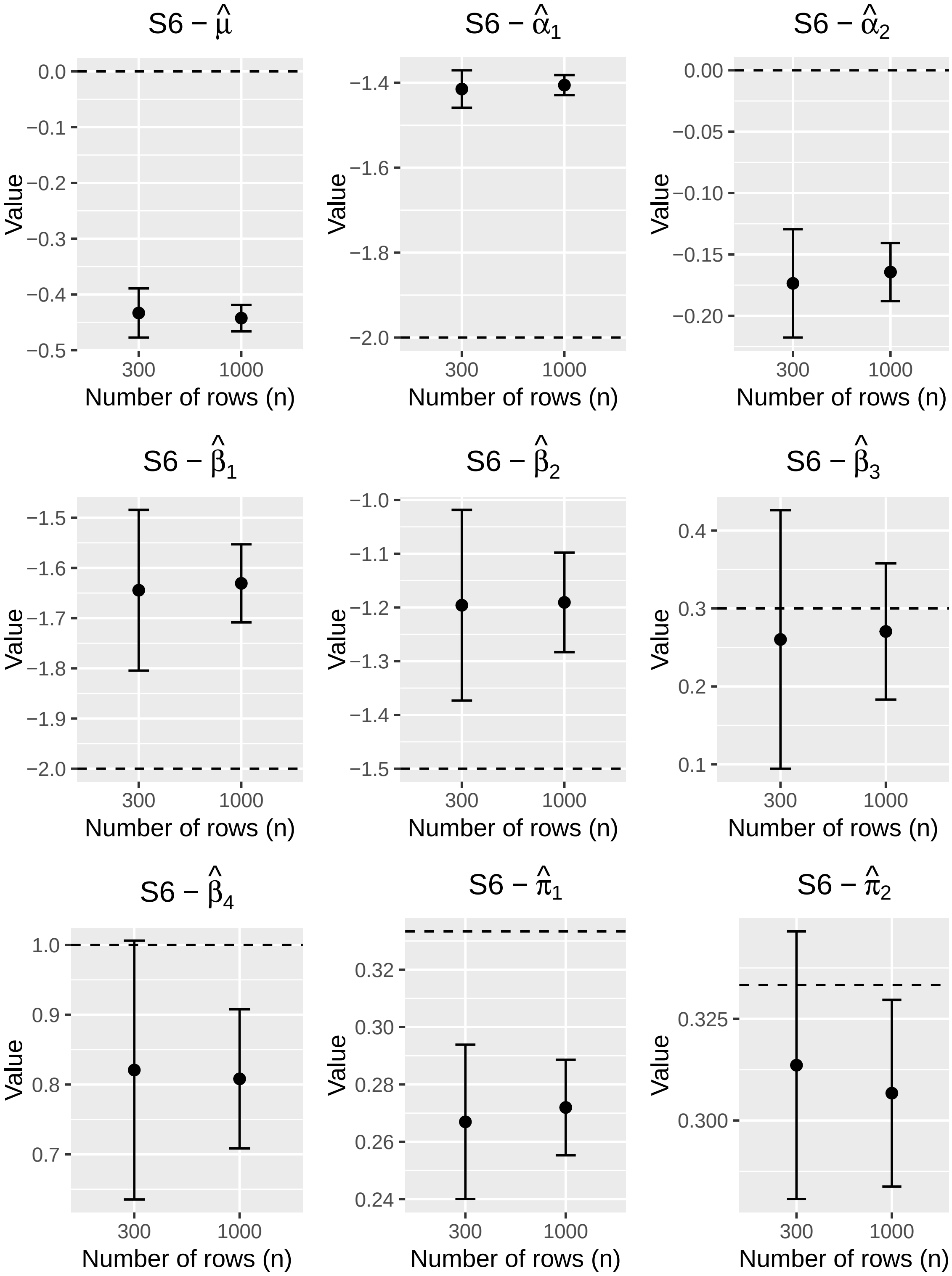

Figure 6: Results for m = 30%, s = 50%.

4 Discussion

- Brief summary from results

- scenario where there is not an observation from each cluster present in the dataset
- if the clustering manages to pick up on if incorrect labels should actually be recorrected -> another Expectation step
- setting initial parameter estimates to random values instead of using algorithm to set randomly
- kept a balanced design for simplicity, assumed that a third of the data is assigned to each cluster
- precision of EM algorithm
- effect of controls

5 Appendix

5.1 Table

Table 7: Mean time in seconds spent to run the EM algorithm for each data scenario based on 100 replicated datasets.

n	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**
300	45.2	43.4	44.0	29.0	26.7	26.6
1000	130.6	125.8	126.6	90.6	85.3	88.8
3000						

References

Antoine, V., & Labroche, N. (2018). Semi-supervised Fuzzy c-Means Variants: A Study on Noisy Label Supervision. In J. Medina, M. Ojeda-Aciego, J. L. Verdegay, D. A. Pelta, I. P. Cabrera, B. Bouchon-Meunier, & R. R. Yager (Eds.), *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations* (Vol. 854, pp. 51–62). Springer International Publishing. https://doi.org/10.1007/978-3-319-91476-3_5

Cai, J., Hao, J., Yang, H., Zhao, X., & Yang, Y. (2023). A review on semi-supervised clustering. *Information Sciences*, *632*, 164–200. https://doi.org/10.1016/j.ins.2023.02.088

Cui, Y. (2025). *Semi-supervised model-based clustering via finite-mixtures using proportional odds models for ordinal data* [PhD thesis, Victoria University of Wellington]. https://doi.org/10.26686/f21e-7fc1

Cui, Y., McMillan, L., & Liu, I. (2024). Semi-supervised Model-Based Clustering for Ordinal Data. In D. Benavides-Prado, S. Erfani, P. Fournier-Viger, Y. L. Boo, & Y. S. Koh (Eds.), *Data Science and Machine Learning* (Vol. 1943, pp. 34–47). Springer Nature Singapore. https://doi.org/10.1007/978-981-99-8696-5_3

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data Via the *EM* Algorithm. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *39*(1), 1–22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London, Series A: Containing Papers of a Mathematical or Physical Character*, *222*(594-604), 309–368. https://doi.org/10.1098/rsta.1922.0009

Gan, H., Fan, Y., Luo, Z., & Zhang, Q. (2018). Local homogeneous consistent safe semi-supervised clustering. *Expert Systems with Applications*, *97*, 384–393. https://doi.org/10.1016/j.eswa.2017.12.046

Gribel, D., Gendreau, M., & Vidal, T. (2022). Semi-supervised clustering with inaccurate pairwise annotations. *Information Sciences*, *607*, 441–457. https://doi.org/10.1016/j.ins.2022.05.035

Madhulatha, T. S. (2012). *An overview on clustering methods*. https://doi.org/10.48550/arXiv.1205.1117

McLachlan, G., & Peel, D. (2000). *Finite Mixture Models* (1st ed.). Wiley. https://doi.org/10.1002/0471721182

Pearson, K. (1894). III. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. (A.)*, (185), 71–110. https://doi.org/10.1098/rsta.1894.0003

Pledger, S., & Arnold, R. (2014). Multivariate methods using mixtures: Correspondence analysis, scaling and pattern-detection. *Computational Statistics & Data Analysis*, *71*, 241–261. https://doi.org/10.1016/j.csda.2013.05.013

R Core Team. (2025). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing. https://www.R-project.org/