

Package ‘SSLfmm’

September 30, 2026

Type Package

Title Semi-Supervised Learning with Mixed Missingness in Finite Mixture Models

Version 0.2.2

Author Geoffrey J. McLachlan [aut] (ORCID: <https://orcid.org/0000-0002-5921-3145>),
Jinran Wu [aut, cre] (ORCID: <https://orcid.org/0000-0002-2388-3614>)

Maintainer Jinran Wu <jinran.wu@uq.edu.au>

Depends R (>= 3.6.0)

Description Semi-supervised Gaussian finite mixture models for partially labelled data under complete-case, missing completely at random (MCAR), entropy-dependent missing at random (MAR), and mixed MCAR/MAR label-missingness formulations. For the mixed formulation, the source of a missing label may be observed or latent. The package supports equal and component-specific covariance matrices, model fitting, simulation, initialization, prediction, classification performance assessment, and entropy-based diagnostics. A semi-synthetic Blood Transfusion data set is included to illustrate the applied workflow.

License GPL-3

URL <https://github.com/wujrtudou/SSLfmm>

BugReports <https://github.com/wujrtudou/SSLfmm/issues>

VignetteBuilder knitr

Encoding UTF-8

LazyData true

Imports graphics, stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Repository CRAN

Date/Publication 2026-09-30 16:00:10 UTC

Contents

SSLfmm-package	2
blood_transfusion	2
classification_performance	3
fit_sslfmm	4
initialize_sslfmm	6
predict.SSLfmm	7
rmix	8
simulate_mixed_missingness	9
simulate_sslfmm	11

Index	13
-------	----

SSLfmm-package	<i>SSLfmm: Semi-Supervised Finite Mixture Models with Mixed Label Missingness</i>
----------------	---

Description

Semi-supervised Gaussian finite mixture models for partially observed class labels under complete-case, MCAR, entropy-dependent MAR, and mixed MCAR/MAR formulations. The mixed formulation allows the missingness source to be observed or latent. The package supports equal or component-specific covariance matrices, simulation, prediction, classification performance assessment, and entropy-based diagnostics.

Details

The intended user-facing functions are `fit_sslfmm()`, `initialize_sslfmm()`, `rmix()`, `simulate_sslfmm()`, `simulate_mixed_missingness()`, `classification_performance()`, together with standard S3 `predict()`, `print()`, and `summary()` methods for fitted objects. The package also includes the semi-synthetic `blood_transfusion` data used in the software-paper application. Low-level likelihood, parameter-packing, Cholesky, and entropy helper functions are internal and are not exported.

blood_transfusion	<i>Semi-Synthetic Blood Transfusion Data</i>
-------------------	--

Description

A semi-synthetic partially observed version of the Blood Transfusion classification data used to illustrate the missing-label models in **SSLfmm**.

Usage

```
data("blood_transfusion")
```

Format

A data frame with 748 observations and 7 variables:

id Observation identifier.

truth Complete reference class label, coded 1 or 2.

observed Class label supplied during fitting; NA denotes a missing label.

missing_indicator Label status/source: observed, mar, or mcar.

Recency Standardized recency measurement.

Frequency Standardized donation-frequency measurement.

Time Standardized time measurement.

Details

The complete labels in truth are retained only for evaluation in the semi-synthetic case study. The response supplied to fit_sslfmm() is observed.

Examples

```
data("blood_transfusion")
table(blood_transfusion$missing_indicator)
```

classification_performance
<i>Classification Performance Metrics</i>

Description

Computes overall and per-class classification performance after removing observations for which truth or predicted is missing. Macro averages are taken over classes represented in the retained truth labels. If a represented true class is never predicted, its precision and F1 score are defined as zero and remain in the macro average.

Usage

```
classification_performance(truth, predicted, posterior = NULL)
```

Arguments

truth	True class labels.
predicted	Predicted class labels; must have the same length as truth.
posterior	Optional numeric matrix with one row per observation and nonnegative finite class probabilities. If columns are named, names must be unique and include every true class used in the calculation.

Value

A list with:

- `metrics`: named numeric vector containing accuracy, error rate, balanced accuracy, macro precision, macro recall, and macro F1; when posterior is supplied, log loss is added;
- `per_class`: data.frame with class support, precision, recall, and F1;
- `confusion_matrix`: truth-by-predicted contingency table.

Posterior rows are normalized internally to unit sum after validation; negative or non-finite values are rejected rather than silently corrected.

Examples

```
truth <- c(1, 1, 2, 2)
pred <- c(1, 2, 2, 2)
post <- matrix(c(.8, .2,
                 .4, .6,
                 .1, .9,
                 .3, .7), ncol = 2, byrow = TRUE)
colnames(post) <- c("1", "2")
classification_performance(truth, pred, post)
```

fit_sslfmm

Fit a Semi-Supervised Gaussian Finite Mixture Model

Description

Fits semi-supervised Gaussian finite-mixture classifiers under complete-case, MCAR, entropy-dependent MAR, or mixed MCAR/MAR label-missingness formulations. For mixed models, the MCAR/MAR source may be known or unknown.

Usage

```
fit_sslfmm(
  x, y, g = NULL,
  method = c("mixed", "mar", "mcar", "cc"),
  covariance_type = c("equal", "unequal"),
  indicator = c("unknown", "known"),
  missing_source = NULL, init = NULL, n_starts = 5L,
  control = list(), entropy_floor = 1e-300,
  alpha_starts = c(0, 0.05, 0.15, 0.30),
  alpha_upper = 1 - 1e-8, ridge = 1e-4, seed = NULL
)
```

Arguments

x	Numeric feature matrix or data.frame with observations in rows and features in columns. Feature values must be fully observed.
y	Class labels with one entry per row of x; use NA for missing labels. Numeric labels must be positive integers in 1:g; factor and character labels are also accepted.
g	Positive integer number of mixture components. It is inferred from observed labels when possible.
method	One of "cc", "mcar", "mar", or "mixed".
covariance_type	"equal" for one covariance matrix shared by all components or "unequal" for one covariance matrix per component.
indicator	For method = "mixed", whether the MCAR/MAR source indicator is "unknown" or "known".
missing_source	Required only for method = "mixed", indicator = "known". Supply "mcar"/"mar" on missing-label rows, or a logical/0-1 MCAR-source indicator. Values on labelled rows are ignored.
init	Optional output from initialize_sslfmm() or a list containing pi, mu, and sigma. The fitting orientation for mu is $g \times p$ (one component per row). A $p \times g$ matrix, matching the simulation input orientation, is also accepted when $p \neq g$ and therefore unambiguous.
n_starts	Positive integer number of fitting starts.
control	A list controlling numerical fitting.
entropy_floor	Single positive finite lower bound used before taking the logarithm of posterior entropy.
alpha_starts	Finite starting values for the latent MCAR-channel probability in the latent mixed model. Every value must lie between zero and alpha_upper.
alpha_upper	Single finite upper bound for alpha, strictly between zero and one.
ridge	Single positive finite covariance regularization value used for initialization and complete-case fitting.
seed	Optional random seed for initialization and perturbed starts.

Details

For method = "mcar", unlabelled observations contribute the Gaussian-mixture marginal density. For method = "mar", label missingness follows a logistic model in log posterior entropy. For method = "mixed", the missing-label probability combines an MCAR channel and the entropy-dependent MAR channel. With a known source indicator, alpha is estimated from the recorded MCAR-source assignments; with an unknown source indicator, alpha is estimated jointly with the other model parameters.

Entropy-dependent "mar" and "mixed" fitting requires $g \geq 2$. With one component posterior entropy is constant and the entropy slope is not identifiable. Simulation with $g = 1$ remains supported as a boundary/API check.

Value

An object of class "SSLfmm". Important components include:

- pi: fitted mixing proportions, named by class;
- mu: fitted $g \times p$ mean matrix, one component per row;
- sigma: one $p \times p$ matrix for equal covariance or a $p \times p \times g$ array for unequal covariance;
- xi and alpha: fitted missingness parameters when applicable;
- posterior: fitted class-posterior matrix with class-labelled columns;
- en and entropy: identical fitted posterior-entropy vectors;
- missing and observed_missing: identical observed label-missingness indicators;
- latent_missing: recorded MCAR-source indicator for known-source mixed fitting, or NA when that source is unknown;
- latent_missing_probability: posterior probability of the MCAR source for each row in the unknown-source mixed model;
- loglik, convergence, optimizer, start_summary, and start_diagnostics: fit and start diagnostics.

Examples

```
set.seed(1)
mu <- matrix(c(-2, 2), nrow = 1, ncol = 2)
sim <- simulate_sslfmm(
  n = 60,
  pi = c(0.5, 0.5),
  mu = mu,
  sigma = matrix(1, 1, 1),
  mechanism = "mcar",
  missing_rate = 0.20,
  seed = 1
)
x <- as.matrix(sim$data["x1"])

# A complete-case fit is deterministic and demonstrates the main interface.
fit <- fit_sslfmm(x, sim$data$label, g = 2,
  method = "cc", covariance_type = "equal")

fit
predict(fit, x[1:5, , drop = FALSE], type = "posterior")
```

initialize_sslfmm

Initialize SSLfmm Model Parameters

Description

Constructs stable starting values for SSLfmm model fitting. Observed class labels anchor component identities; k-means is used only as a fallback for components without adequate labelled observations.

Usage

```
initialize_sslfmm(
  x, y, g = NULL,
  covariance_type = c("equal", "unequal"),
  ridge = 1e-4, seed = NULL
)
```

Arguments

<code>x</code>	Numeric feature matrix or data.frame with observations in rows and features in columns.
<code>y</code>	Class labels with one entry per row of <code>x</code> ; use NA for missing labels.
<code>g</code>	Positive integer number of mixture components. Inferred from observed labels when possible.
<code>covariance_type</code>	"equal" for a shared covariance matrix or "unequal" for component-specific covariance matrices.
<code>ridge</code>	Single positive finite diagonal regularization value for covariance starts.
<code>seed</code>	Optional seed used only for k-means fallback initialization.

Value

An object of class "SSLfmm_init", implemented as a list containing `pi`, `mu`, `sigma`, `g`, `p`, `covariance_type`, and `label_levels`. The mean matrix `mu` is $g \times p$, with one component per row. For equal covariance, `sigma` is $p \times p$; for unequal covariance, it is $p \times p \times g$.

Examples

```
x <- rbind(matrix(rnorm(20, -1), 10, 2),
             matrix(rnorm(20, 1), 10, 2))
y <- c(rep(1, 7), rep(NA, 3), rep(2, 7), rep(NA, 3))
init <- initialize_sslfmm(x, y, g = 2,
                        covariance_type = "unequal", seed = 1)

init$pi
init$mu
dim(init$sigma)
```

predict.SSLfmm

Methods for SSLfmm Fits

Description

Standard prediction, printing, and summary methods for fitted SSLfmm objects.

Usage

```
## S3 method for class 'SSLfmm'
predict(object, newdata,
  type = c("class", "posterior", "entropy", "all"), ...)
## S3 method for class 'SSLfmm'
print(x, ...)
## S3 method for class 'SSLfmm'
summary(object, ...)
```

Arguments

<code>object, x</code>	An object of class "SSLfmm".
<code>newdata</code>	Numeric feature matrix or data.frame with the same number of feature columns as the fitted model.
<code>type</code>	Prediction output: "class", "posterior", "entropy", or "all".
<code>...</code>	Unused.

Value

For `predict()`, "class" returns predicted labels in the same broad label type used for fitting; "posterior" returns an $n \times g$ numeric matrix with class-labelled columns; "entropy" returns a numeric vector; and "all" returns a list with components class, posterior, and entropy. `print()` returns the fit invisibly. `summary()` returns a compact fitted-parameter list.

Examples

```
mu <- matrix(c(-2, 2), nrow = 1, ncol = 2)
s <- simulate_sslfmm(50, c(0.5, 0.5), mu, matrix(1, 1, 1),
  mechanism = "mcar", missing_rate = 0.2, seed = 1)
x <- as.matrix(s$data["x1"])
f <- fit_sslfmm(x, s$data$label, g = 2, method = "cc")
predict(f, x[1:5, , drop = FALSE], type = "posterior")
predict(f, x[1:5, , drop = FALSE], type = "all")
```

rmix

Simulate from a Gaussian Finite Mixture

Description

Generates observations from a Gaussian finite mixture without introducing label missingness.

Usage

```
rmix(n = 200L, pi, mu, sigma, seed_number = NULL)
```

Arguments

<code>n</code>	Positive integer number of observations.
<code>pi</code>	Positive finite mixing proportions of length g that sum to one.
<code>mu</code>	Numeric $p \times g$ matrix of component means; each column is one component.
<code>sigma</code>	Either one shared numeric $p \times p$ symmetric positive-definite covariance matrix or a numeric $p \times p \times g$ array containing one symmetric positive-definite covariance matrix per component. For $p = 1$, a length-one scalar is also accepted as a shared variance.
<code>seed_number</code>	Optional random seed.

Value

A data.frame with feature columns `x1`, ..., `xp` followed by integer component label `truth`.

Examples

```
mu <- matrix(c(-1, 1), nrow = 1, ncol = 2)
dat <- rmix(
  n = 20,
  pi = c(0.5, 0.5),
  mu = mu,
  sigma = matrix(1, 1, 1),
  seed_number = 1
)
head(dat)
```

`simulate_mixed_missingness`

Simulate Semi-Supervised Data Under Mixed Label Missingness

Description

Convenience wrapper for `simulate_sslfmm(..., mechanism = "mixed")`. It uses the same mixed missingness mechanism and the same stable return-value contract.

Usage

```
simulate_mixed_missingness(
  n = 500L, pi, mu, sigma,
  alpha = 0.10, xi0 = NULL, xi1 = 3,
  mar_rate = 0.30, seed = NULL,
  entropy_floor = 1e-300
)
```

Arguments

<code>n</code>	Positive integer number of observations.
<code>pi</code>	Positive finite mixing proportions of length g that sum to one.
<code>mu</code>	Numeric $p \times g$ matrix of component means.
<code>sigma</code>	Shared $p \times p$ covariance matrix or $p \times p \times g$ component-specific covariance array.
<code>alpha</code>	MCAR-channel probability in $[0, 1]$.
<code>xi0</code>	MAR intercept; when NULL, calibrated to <code>mar_rate</code> .
<code>xi1</code>	Positive finite entropy slope.
<code>mar_rate</code>	Target mean MAR probability when <code>xi0</code> = NULL.
<code>seed</code>	Optional random seed.
<code>entropy_floor</code>	Positive finite entropy floor.

Value

The same five-component list documented for `simulate_sslfmm()`: `data`, `true_setup`, `groups`, `probs`, and `raw`.

See Also

`simulate_sslfmm()`, `rmix()`

Examples

```
mu <- matrix(c(-1, 1), nrow = 1, ncol = 2)
sim <- simulate_mixed_missingness(
  n = 40,
  pi = c(0.5, 0.5),
  mu = mu,
  sigma = matrix(1, 1, 1),
  alpha = 0.10,
  mar_rate = 0.25,
  seed = 1
)
stopifnot(
  is.list(sim),
  is.data.frame(sim$data),
  identical(sim$data$missing, sim$data$observed_missing)
)
```

simulate_sslfmm	<i>Simulate Semi-Supervised Finite-Mixture Data</i>
-----------------	---

Description

Simulates semi-supervised Gaussian finite-mixture data with class-label missingness under complete-case, MCAR, entropy-dependent MAR, or mixed MCAR/MAR mechanisms.

Usage

```
simulate_sslfmm(
  n = 500L, pi, mu, sigma,
  mechanism = c("mixed", "mar", "mcar", "cc"),
  alpha = 0.10, missing_rate = NULL,
  xi0 = NULL, xi1 = 3, mar_rate = 0.30,
  seed = NULL, entropy_floor = 1e-300
)
```

Arguments

n	Positive integer number of observations.
pi	Positive finite mixing proportions of length g that sum to one.
mu	Numeric $p \times g$ matrix of component means; each column is one component.
sigma	Either one shared numeric $p \times p$ symmetric positive-definite covariance matrix or a numeric $p \times p \times g$ array with one covariance matrix per component. For $p = 1$, a length-one scalar is also accepted as a shared variance.
mechanism	One of "cc", "mcar", "mar", or "mixed".
alpha	MCAR-channel probability for mechanism = "mixed". For mechanism = "mcar", this is the missing-label rate when missing_rate is NULL.
missing_rate	Optional MCAR missing-label rate used only for mechanism = "mcar".
xi0	MAR intercept. If NULL, it is calibrated on the generated feature sample so that the mean MAR probability equals mar_rate.
xi1	Positive finite slope multiplying log posterior entropy for MAR or mixed simulation.
mar_rate	Target average MAR probability when xi0 = NULL; must lie strictly between zero and one.
seed	Optional random seed.
entropy_floor	Positive finite lower bound used before taking the logarithm of posterior entropy.

Details

For mixed missingness, let q_i denote the entropy-dependent MAR probability and let the MCAR trigger have probability α . The observed missing-label probability is

$$\alpha + (1 - \alpha)q_i.$$

The simulator records both the true MCAR trigger and a latent potential-MAR trigger. MCAR takes precedence when both occur. This is distributionally equivalent to generating the MAR trigger only among observations not selected by MCAR, while also making the documented group decomposition reproducible.

When $g = 1$, posterior entropy is constant. Simulation remains supported for API/boundary testing, and automatic `xi0` calibration uses the resulting constant-probability analytic solution. Entropy-dependent MAR slope estimation with $g = 1$ is not identifiable and is therefore not supported by `fit_sslfmm()`.

Value

Always a list with exactly the top-level components `data`, `true_setup`, `groups`, `probs`, and `raw`.

- `data`: a `data.frame` whose leading columns are, in order, `x1`, ..., `xp`, `en`, `missing`, `label`, and `truth`. It then contains `observed_missing`, `latent_missing`, `missing_source`, `prob_mar`, and `entropy`. `en` and `entropy` are identical; `missing` and `observed_missing` are identical. `latent_missing` is the true MCAR-channel trigger in the simulation.
- `true_setup`: a list beginning with `pi`, `mu`, and `sigma`, followed by covariance/missingness metadata and the true missingness parameters when applicable.
- `groups`: a list beginning with `mar_group`, `obs_group`, `mcmar_in_mar`, and `mcmar_in_obs`. `mar_group` is the latent potential-MAR stratum, `obs_group` its complement, and the two `mcmar_in_*` entries are intersections with the MCAR trigger. Additional entries `observed`, `mcmar`, `mar`, and `missing` give directly useful row indices.
- `probs`: numeric vector of MAR probabilities q_i ; it is identical to `data$prob_mar`.
- `raw`: the original `rmix()` `data.frame` augmented with `en`, `missing`, `label`, the explicit missingness indicators/source, and MAR probabilities.

Examples

```
mu <- matrix(c(-1.5, 1.5), nrow = 1, ncol = 2)
Sigma <- matrix(1, 1, 1)

s_cc <- simulate_sslfmm(30, c(0.5, 0.5), mu, Sigma,
  mechanism = "cc", seed = 1)
s_mcar <- simulate_sslfmm(30, c(0.5, 0.5), mu, Sigma,
  mechanism = "mcar", missing_rate = 0.25, seed = 2)
s_mar <- simulate_sslfmm(30, c(0.5, 0.5), mu, Sigma,
  mechanism = "mar", mar_rate = 0.25, seed = 3)
s_mixed <- simulate_sslfmm(30, c(0.5, 0.5), mu, Sigma,
  mechanism = "mixed", alpha = 0.10,
  mar_rate = 0.25, seed = 4)

names(s_mixed)
head(s_mixed$data)
```

Index

* **datasets**

blood_transfusion, [2](#)

blood_transfusion, [2](#)

classification_performance, [3](#)

fit_sslfmm, [4](#)

initialize_sslfmm, [6](#)

predict.SSLfmm, [7](#)

print.SSLfmm(predict.SSLfmm), [7](#)

rmix, [8](#)

simulate_mixed_missingness, [9](#)

simulate_sslfmm, [11](#)

SSLfmm (SSLfmm-package), [2](#)

SSLfmm-package, [2](#)

summary.SSLfmm(predict.SSLfmm), [7](#)