Package ‘lsirm12pl’

June 30, 2026

Type Package

Title Latent Space Item Response Model

Version 2.0.3

Date 2026-06-29

Maintainer Gwanghee Kim <musagh08@yonsei.ac.kr>

Description Analysis of dichotomous, ordinal, and continuous response data using latent space item response model ('LSIRM'). Provides 1PL and 2PL 'LSIRM' for binary response data as described in Jeon et al. (2021) <doi:10.1007/s11336-021-09762-5>, graded response models ('GRM') for ordinal data (De Carolis et al., 2025, <doi:10.1080/00273171.2025.2605678>), and extensions for continuous response data. Supports Bayesian model selection with spike-and-slab priors, adaptive MCMC algorithms, and methods for handling missing data under missing at random ('MAR') and missing completely at random ('MCAR') assumptions. Provides various diagnostic plots to inspect the latent space and summaries of estimated parameters.

License GPL-3

Imports Rcpp (>= 1.0.5), MCMCpack, ggplot2, GPArotation, dplyr (>= 1.1.4), grDevices, rlang, pROC, coda, spatstat, spatstat.geom, spatstat.random, plotly, gridExtra, grid, tidyr, fpc, kernlab, plyr, purrr

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

RoxygenNote 7.3.2

Suggests knitr

Depends R (>= 4.0.0)

LazyData true

NeedsCompilation yes

Author Dongyoung Go [aut],
Gwanghee Kim [aut, cre],
Jina Park [aut],
Ickhoon Jin [ctb],
Minjeong Jeon [ctb]

Repository CRAN

Date/Publication 2026-06-30 16:30:02 UTC

Contents

`BFPT` *Big Five Personality Test*

Description

A dataset containing the result of personality test for 50 questions from 1,000 random sampled people.

Usage

```
data(BFPT)
```

Format

A matrix with 1,015,341 rows and 50 columns.

Details

A dataset collected in 2016-2018 through an interactive on-line personality test, containing the result of personality test for 50 questions. 1,000 people are random sampled from the original dataset containing 1,015,341 people. The scale is labeled as 1=Disagree, 3=Neutral and 5=Agree.

Source

https://www.kaggle.com/tunguz/big-five-personality-test `diagnostic` *Diagnostic the result of LSIRM.*

Description

`diagnostic` checks the convergence of MCMC for LSIRM parameters using various diagnostic tools, such as trace plots, posterior density distributions, autocorrelation functions (ACF), and Gelman-Rubin-Brooks plots.

Usage

```
diagnostic(
  object,
  draw.item = list(beta = c(1), theta = c(1)),
  gelman.diag = FALSE
)
```

Arguments

`object` Object of class `lsirm`.

`draw.item` List or Character vector; specifies which parameters to diagnose. Can be:

- A character vector of parameter names to diagnose using default settings, e.g. `"gamma"` or `c("beta", "gamma")`.
- A list combining named index vectors/matrices (e.g., `beta = c(1)`, `z = matrix(c(1,1), ncol=2)`) and unnamed string elements representing scalar parameters (e.g., `"gamma", "sigma"`), or parameters mapped to `TRUE`/`NULL` (e.g., `gamma = TRUE`).

Supported parameters include: "beta" (item difficulty/threshold), "theta" (respondent ability), "gamma" (distance weight), "alpha" (item discrimination), "sigma" (error variance for continuous models), "theta_sd" (standard deviation of theta), "z" (respondent latent positions), "w" (item latent positions), and "zw.dist" (latent distances). Default is `list(beta = c(1), theta = c(1))`.

`gelman.diag` Logical; If TRUE, the Gelman-Rubin convergence diagnostic will be printed. Default is FALSE.

Value

`diagnostic` returns plots for checking MCMC convergence for selected parameters.

Examples

```
# Generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5), ncol=10, nrow=50)

# For 1PL LSIRM
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = FALSE, fixed_gamma = FALSE))
```

```
diagnostic(lsirm_result)

# For 2PL LSIRM
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = FALSE, fixed_gamma = FALSE))
diagnostic(lsirm_result)

# For specific latent positions
diagnostic(lsirm_result, draw.item = list(z = matrix(c(1,1, 2,1, 1,2), ncol=2, byrow=TRUE),
                                       w = matrix(c(1,1, 2,1), ncol=2, byrow=TRUE)))

# Using flexible draw.item formats:
# 1) Character string for a single parameter
diagnostic(lsirm_result, draw.item = "gamma")

# 2) Character vector for multiple parameters with default settings
diagnostic(lsirm_result, draw.item = c("beta", "gamma"))

# 3) List combining named indexes and unnamed strings
diagnostic(lsirm_result, draw.item = list(beta = c(1), "gamma"))
```

---

gof *Goodness-of-fit LSIRM*

Description

gof is goodness-of-fit the latent space of fitted LSIRM.

Usage

```
gof(object, chain.idx = 1)
```

Arguments

object	Object of class lsirm.
chain.idx	Numeric; Index of MCMC chain. Default is 1.

Value

gof returns the boxplot or AUC plot

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)
lsirm_result <- lsirm(data ~ lsirm1pl())
gof(lsirm_result)
```

lsirm — *Fit a LSIRM (Latent Space Item Response Model)*

Description

lsirm is used to fit 1PL LSIRM, 2PL LSIRM, and ordinal GRM LSIRM using Bayesian methods.

Usage

```
lsirm(formula, ...)
```

Arguments

formula	The form of formula is `lsirm(A ~ <term 1>(<term 2>, <term 3> ...))`, where `A` is an item response matrix to be analyzed, `<term1>` is the model you want to fit and has one of the following values: "lsirm1pl", "lsirm2pl", "lsirmgrm", and "lsirmgrm2pl"., and `<term 2>`, `<term 3>`, etc. are each option for the model.
...	Additional arguments for the corresponding function.

Details

The descriptions of options for each model, such as `<term 2>` and `<term 3>`, are included in `lsirm1pl` for 1PL LSIRM, `lsirm2pl` for 2PL LSIRM, and `lsirmgrm` for ordinal GRM LSIRM.

Value

`lsirm` returns an object of class `list`.

See corresponding functions such as `lsirm1pl` for 1PL LSIRM and `lsirm2pl` for 2PL LSIRM.

See Also

`lsirm1pl` for 1PL LSIRM.

`lsirm2pl` for 2PL LSIRM.

`lsirmgrm` for ordinal GRM LSIRM.

Examples

```
# generate example item response matrix
data <- matrix(rbinom(500, size = 1, prob = 0.5), ncol = 10, nrow = 50)

lsirm_result <- lsirm(data ~ lsirm1pl())
lsirm_result <- lsirm(data ~ lsirm2pl())

# Realistic example with BFPT data
data(BFPT)
dat <- BFPT
dat[(dat == 0) | (dat == 6)] <- NA
```

```
reverse <- c(2, 4, 6, 8, 10, 11, 13, 15, 16, 17, 18, 19, 20, 21, 23, 25, 27, 32, 34, 36, 42, 44, 46)
dat[, reverse] <- 6 - dat[, reverse]
dat <- dat[complete.cases(dat), ]
# Fit model (subset for speed)
fit_bfpt <- lsirm(dat[1:50, 1:10] ~ lsirmgrm(niter = 1000, nburn = 500))
summary(fit_bfpt)

# Fit with adaptive MCMC
lsirm_result <- lsirm(data ~ lsirm1pl(adapt = list(use_adapt = TRUE)))
```

`lsirm.formula` *Formula function for LSIRM*

Description

lsirm.formula is formula object.

Usage

```
## S3 method for class 'formula'
lsirm(formula, ...)
```

Arguments

`formula`	The form of formula is `lsirm(A ~ <term 1>(<term 2>, <term 3> ...))`, where `A` is an item response matrix to be analyzed, `<term1>` is the model you want to fit and has one of the following values: "lsirm1pl", "lsirm2pl", "lsirmgrm", and "lsirmgrm2pl"., and `<term 2>`, `<term 3>`, etc., are each option for the model.
`...`	Additional arguments for the corresponding function.

`lsirm12pl` *lsirm12pl-package*

Description

#' Analysis of dichotomous, continuous, and ordinal response data using latent space models. It includes 1PL and 2PL Latent Space Item Response Models (LSIRM) for binary and continuous data, and Graded Response Models (GRM) for ordinal data, as described in Jeon et al. (2021) <doi:10.1007/s11336-021-09762-5>. Bayesian model selection with spike-and-slab prior and method for dealing data with missing value under missing at random (MAR), missing completely at random (MCAR) are also supported. Various diagnostic plots are available to inspect the latent space and summary of estimated parameters.

References

Jeon, M., Jin, I. H., Schweinberger, M., & Baugh, S. (2021). Mapping unobserved item-respondent interactions: A latent space item response model with interaction map. *Psychometrika*, 86(2), 378–403. https://link.springer.com/article/10.1007/s11336-021-09762-5

`lsirm1pl` *Fit a 1PL LSIRM for binary and continuous item response data*

Description

`lsirm1pl` integrates all functions related to 1PL LSIRM. Various 1PL LSIRM function can be used by setting the `spikenslab`, `fixed_gamma`, and `missing_data` arguments.

This function can be used regardless of the data type, providing a unified approach to model fitting.

Usage

```
lsirm1pl(
  data,
  spikenslab = FALSE,
  fixed_gamma = FALSE,
  missing_data = NA,
  chains = 1,
  multicore = 1,
  seed = NA,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL,
  ...
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`spikenslab`	Logical; specifies whether to use a model selection approach. Default is FALSE.
`fixed_gamma`	Logical; indicates whether to fix gamma at 1. Default is FALSE.
`missing_data`	Character; the type of missing data assumed. Options are NA, "mar", or "mcar". Default is NA.
`chains`	Integer; the number of MCMC chains to run. Default is 1.
`multicore`	Integer; the number of cores to use for parallel execution. Default is 1.
`seed`	Integer; the seed number for MCMC fitting. Default is NA.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.
`nburn`	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
`nthin`	Integer; the number of MCMC iterations to thin. Default is 5.
`nprint`	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
`jump_beta`	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
`jump_theta`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_gamma`	Numeric; the jumping rule for the gamma proposal density. Default is 0.2.
`jump_z`	Numeric; the jumping rule for the z proposal density. Default is 0.5.
`jump_w`	Numeric; the jumping rule for the w proposal density. Default is 0.5.
`pr_mean_beta`	Numeric; the mean of the normal prior for beta. Default is 0.
`pr_sd_beta`	Numeric; the standard deviation of the normal prior for beta. Default is 1.
`pr_mean_theta`	Numeric; the mean of the normal prior for theta. Default is 0.
`pr_sd_theta`	Numeric; the standard deviation of the normal prior for theta. Default is 1.
`pr_mean_gamma`	Numeric; mean of log normal prior for gamma. Default is 0.5.
`pr_sd_gamma`	Numeric; standard deviation of log normal prior for gamma. Default is 1.
`pr_a_theta`	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
`pr_b_theta`	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
`verbose`	Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.
`fix_theta_sd`	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

adapt — List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

... — Additional arguments passed to the underlying model-fitting functions.

Details

Additional arguments and return values for each function are documented in the respective function's description.

* For LSIRM with data included missing value are detailed in lsirm1pl_mar and lsirm1pl_mcar.

* For LSIRM using the spike-and-slab model selection approach are detailed in lsirm1pl_ss.

* For continuous version of LSIRM are detailed in lsirm1pl_normal_o.

For 1PL LSIRM with binary item response data, the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term:

$$logit(P(Y_{j,i} = 1|\theta_j, \beta_i, \gamma, z_j, w_i)) = \theta_j + \beta_i - \gamma||z_j - w_i||$$

For 1PL LSIRM with continuous item response data, the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term:

$$Y_{j,i} = \theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$.

Value

`lsirm1pl` returns an object of list. The basic return list containing the following components:

- `data` — A data frame or matrix containing the variables used in the model.
- `bic` — A numeric value representing the Bayesian Information Criterion (BIC).
- `mcmc_inf` — Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
- `map_inf` — The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
- `beta_estimate` — Posterior estimates of the beta parameter.
- `theta_estimate` — Posterior estimates of the theta parameter.
- `sigma_theta_estimate` — Posterior estimates of the standard deviation of theta.
- `z_estimate` — Posterior estimates of the z parameter.
- `w_estimate` — Posterior estimates of the w parameter.
- `beta` — Posterior samples of the beta parameter.
- `theta` — Posterior samples of the theta parameter.
- `theta_sd` — Posterior samples of the standard deviation of theta.
- `z` — Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
- `w` — Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
- `z_raw` — Posterior samples of the z parameter before Procrustes matching.
- `w_raw` — Posterior samples of the w parameter before Procrustes matching.
- `accept_beta` — Acceptance ratio for the beta parameter.
- `accept_theta` — Acceptance ratio for the theta parameter.
- `accept_z` — Acceptance ratio for the z parameter.
- `accept_w` — Acceptance ratio for the w parameter.
- `...` — Additional return values for various settings. Refer to the functions in the Details.

Note

If both spikenslab and fixed_gamma are set TRUE, it returns error because both are related to gamma.

References

Jeon, M., Jin, I. H., Schweinberger, M., & Baugh, S. (2021). Mapping unobserved item-respondent interactions: A latent space item response model with interaction map. *Psychometrika*, 86(2), 378–403. https://link.springer.com/article/10.1007/s11336-021-09762-5

Roberts, G. O., Gelman, A., & Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1), 110–120. doi:10.1214/aoap/1034625254

Roberts, G. O., & Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4), 351–367. doi:10.1214/ss/1015346320

Andrieu, C., & Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4), 343–373. doi:10.1007/s112220089110y

See Also

The LSIRM for 1PL LSIRM for binary item response data as following:

lsirm1pl_o, lsirm1pl_fixed_gamma, lsirm1pl_mar,lsirm1pl_mcar, lsirm1pl_fixed_gamma_mar, lsirm1pl_fixed_gamma_mcar, lsirm1pl_ss, lsirm1pl_mar_ss, and lsirm1pl_mcar_ss

The LSIRM for 1PL LSIRM for continuous item response data as following:

lsirm1pl_normal_o, lsirm1pl_normal_fixed_gamma, lsirm1pl_normal_mar, lsirm1pl_normal_mcar,lsirm1pl_norm
lsirm1pl_normal_fixed_gamma_mcar, lsirm1pl_normal_ss, lsirm1pl_normal_mar_ss, lsirm1pl_normal_mcar_ss

Examples

```
# generate example item response matrix
data <- matrix(rbinom(500, size = 1, prob = 0.5), ncol = 10, nrow = 50)
lsirm_result <- lsirm1pl(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl())

# Fit with adaptive MCMC
lsirm_result <- lsirm1pl(data, adapt = list(use_adapt = TRUE))
```

---

lsirm1pl_fixed_gamma *1PL LSIRM fixing gamma to 1.*

Description

lsirm1pl_fixed_gamma is used to fit 1PL LSIRM with gamma fixed to 1. lsirm1pl_fixed_gamma factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_fixed_gamma(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.
`nburn`	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
`nthin`	Integer; the number of MCMC iterations to thin. Default is 5.
`nprint`	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
`jump_beta`	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
`jump_theta`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_z`	Numeric; the jumping rule for the z proposal density. Default is 0.5.
`jump_w`	Numeric; the jumping rule for the w proposal density. Default is 0.5.

pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
verbose	Logical; If TRUE, MCMC samples are printed for each nprint. default value is FALSE
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

adapt

List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

`lsirm1pl_fixed_gamma` models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space:

$$logit(P(Y_{j,i} = 1|\theta_j, \beta_i, z_j, w_i)) = \theta_j + \beta_i - ||z_j - w_i||$$

Value

`lsirm1pl_fixed_gamma` returns an object of list containing the following components:

`data`	Data frame or matrix containing the variables in the model.
`bic`	Numeric value with the corresponding BIC.
`mcmc_inf`	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
`map_inf`	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
`beta_estimate`	Posterior estimates of the beta parameter.
`theta_estimate`	Posterior estimates of the theta parameter.
`sigma_theta_estimate`	Posterior estimates of the standard deviation of theta.
`z_estimate`	Posterior estimates of the z parameter.
`w_estimate`	Posterior estimates of the w parameter.
`beta`	Posterior samples of the beta parameter.
`theta`	Posterior samples of the theta parameter.
`theta_sd`	Posterior samples of the standard deviation of theta.
`z`	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`w`	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`z_raw`	Posterior samples of the z parameter without procrustes matching.
`w_raw`	Posterior samples of the w parameter without procrustes matching.
`accept_beta`	Acceptance ratio for the beta parameter.
`accept_theta`	Acceptance ratio for the theta parameter.
`accept_z`	Acceptance ratio for the z parameter.
`accept_w`	Acceptance ratio for the w parameter.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

lsirm_result <- lsirm1pl_fixed_gamma(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = FALSE, fixed_gamma = TRUE))
```

lsirm1pl_fixed_gamma_mar *1PL LSIRM fixing gamma to 1 for missing at random data.*

Description

lsirm1pl_fixed_gamma_mar is used to fit LSIRM with gamma fixed to 1 in incomplete data assumed to be missing at random. lsirm1pl_fixed_gamma_mar factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while considering the missing element under the assumption of missing at random. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_fixed_gamma_mar(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.

nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
missing.val	Numeric; a number to replace missing values. Default is 99.
verbose	Logical; If TRUE, MCMC samples are printed for each nprint. default value is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

adapt — List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.

- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm1pl_fixed_gamma_mar models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space:

$$logit(P(Y_{j,i} = 1|\theta_j, \beta_i, z_j, w_i)) = \theta_j + \beta_i - ||z_j - w_i||$$

Under the assumption of missing at random, the model takes the missing element into consideration in the sampling procedure. For the details of missing at random assumption and data augmentation, see References.

Value

lsirm1pl_fixed_gamma_mar returns an object of list containing the following components:

data	Data frame or matrix containing the variables in the model.
missing.val	A number to replace missing values.
bic	Numeric value with the corresponding BIC.
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
imp_estimate	Probability of imputing a missing value with 1.
beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.
theta_sd	Posterior samples of the standard deviation of theta.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.

`w`	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`z_raw`	Posterior samples of the z parameter without procrustes matching.
`w_raw`	Posterior samples of the w parameter without procrustes matching.
`imp`	Imputation for missing Values using posterior samples.
`accept_beta`	Acceptance ratio for the beta parameter.
`accept_theta`	Acceptance ratio for the theta parameter.
`accept_z`	Acceptance ratio for the z parameter.
`accept_w`	Acceptance ratio for the w parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm1pl_fixed_gamma_mar(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = FALSE, fixed_gamma = TRUE,
missing_data = "mar", missing.val = NA))
```

`lsirm1pl_fixed_gamma_mcar`

1PL LSIRM fixing gamma to 1 for missing completely at random data.

Description

lsirm1pl_fixed_gamma_mcar is used to fit LSIRM with gamma fixed to 1 in incomplete data assumed to be missing completely at random. lsirm1pl_fixed_gamma_mcar factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while ignoring the missing element under the assumption of missing completely at random. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_fixed_gamma_mcar(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.
`nburn`	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
`nthin`	Integer; the number of MCMC iterations to thin. Default is 5.
`nprint`	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
`jump_beta`	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
`jump_theta`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_z`	Numeric; the jumping rule for the z proposal density. Default is 0.5.
`jump_w`	Numeric; the jumping rule for the w proposal density. Default is 0.5.
`pr_mean_beta`	Numeric; the mean of the normal prior for beta. Default is 0.
`pr_sd_beta`	Numeric; the standard deviation of the normal prior for beta. Default is 1.
`pr_mean_theta`	Numeric; the mean of the normal prior for theta. Default is 0.
`pr_sd_theta`	Numeric; the standard deviation of the normal prior for theta. Default is 1.

- `pr_a_theta` Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
- `pr_b_theta` Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
- `missing.val` Numeric; a number to replace missing values. Default is 99.
- `verbose` Logical; If TRUE, MCMC samples are printed for each `nprint`. default value is FALSE
- `fix_theta_sd` Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
- `adapt` List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:
  - `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
  - `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
  - `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^α. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
  - `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^α. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.
  - `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
  - `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is `0.44`.
  - `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
  - `target_accept_beta/theta/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

Details

`lsirm1pl_fixed_gamma_mcar` models the probability of correct response by respondent j to item

i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space:

$$logit(P(Y_{j,i} = 1|\theta_j, \beta_i, z_j, w_i)) = \theta_j + \beta_i - ||z_j - w_i||$$

Under the assumption of missing completely at random, the model ignores the missing element in doing inference. For the details of missing completely at random assumption and data augmentation, see References.

Value

`lsirm1pl_fixed_gamma_mcar` returns an object of list containing the following components:

`data`	Data frame or matrix containing the variables in the model.
`missing.val`	A number to replace missing values.
`bic`	Numeric value with the corresponding BIC.
`mcmc_inf`	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
`map_inf`	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
`beta_estimate`	Posterior estimates of the beta parameter.
`theta_estimate`	Posterior estimates of the theta parameter.
`sigma_theta_estimate`	Posterior estimates of the standard deviation of theta.
`z_estimate`	Posterior estimates of the z parameter.
`w_estimate`	Posterior estimates of the w parameter.
`beta`	Posterior samples of the beta parameter.
`theta`	Posterior samples of the theta parameter.
`theta_sd`	Posterior samples of the standard deviation of theta.
`z`	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`w`	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`z_raw`	Posterior samples of the z parameter without procrustes matching.
`w_raw`	Posterior samples of the w parameter without procrustes matching.
`accept_beta`	Acceptance ratio for the beta parameter.
`accept_theta`	Acceptance ratio for the theta parameter.
`accept_z`	Acceptance ratio for the z parameter.
`accept_w`	Acceptance ratio for the w parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm1pl_fixed_gamma_mcar(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = FALSE, fixed_gamma = TRUE,
missing_data = "mcar", missing.val = NA))
```

---

lsirm1pl_mar *1PL LSIRM for missing at random data.*

Description

lsirm1pl_mar is used to fit 1PL LSIRM in incomplete data assumed to be missing at random. lsirm1pl_mar factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while considering the missing element under the assumption of missing at random. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_mar(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
```

```
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the gamma proposal density. Default is 0.2
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_mean_gamma	Numeric; mean of log normal prior for gamma. Default is 0.5.
pr_sd_gamma	Numeric; standard deviation of log normal prior for gamma. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
missing.val	Numeric; a number to replace missing values. Default is 99.
verbose	Logical; If TRUE, MCMC samples are printed for each `nprint`. default value is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

adapt
: List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm1pl_mar models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term:

$$logit(P(Y_{j,i} = 1|\theta_j, \beta_i, \gamma, z_j, w_i)) = \theta_j + \beta_i - \gamma||z_j - w_i||$$

Under the assumption of missing at random, the model takes the missing element into consideration in the sampling procedure. For the details of missing at random assumption and data augmentation, see References.

Value

`lsirm1pl_mar` returns an object of list containing the following components:

`data`	Data frame or matrix containing the variables in the model.
`missing.val`	A number to replace missing values.
`bic`	Numeric value with the corresponding BIC.
`mcmc_inf`	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
`map_inf`	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
`beta_estimate`	Posterior estimates of the beta parameter.
`theta_estimate`	Posterior estimates of the theta parameter.
`sigma_theta_estimate`	Posterior estimates of the standard deviation of theta.
`gamma_estimate`	posterior estimates of gamma parameter.
`z_estimate`	Posterior estimates of the z parameter.
`w_estimate`	Posterior estimates of the w parameter.
`imp_estimate`	Probability of imputing a missing value with 1.
`beta`	Posterior samples of the beta parameter.
`theta`	Posterior samples of the theta parameter.
`gamma`	Posterior samples of the gamma parameter.
`theta_sd`	Posterior samples of the standard deviation of theta.
`z`	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`w`	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`z_raw`	Posterior samples of the z parameter without procrustes matching.
`w_raw`	Posterior samples of the w parameter without procrustes matching.
`imp`	Imputation for missing Values using posterior samples.
`accept_beta`	Acceptance ratio for the beta parameter.
`accept_theta`	Acceptance ratio for the theta parameter.
`accept_z`	Acceptance ratio for the z parameter.
`accept_w`	Acceptance ratio for the w parameter.
`accept_gamma`	Acceptance ratio for the gamma parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm1pl_mar(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = FALSE, fixed_gamma = FALSE,
missing_data ='mar', missing.val = NA))
```

`lsirm1pl_mar_ss` *1PL LSIRM with model selection approach for missing at random data.*

Description

lsirm1pl_mar_ss is used to fit 1PL LSIRM with model selection approach based on spike-and-slab priors in incomplete data assumed to be missing at random. lsirm1pl_mar_ss factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while considering the missing element under the assumption of missing at random. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_mar_ss(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
```

```
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_xi_a = 1,
  pr_xi_b = 1,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_spike_mean	Numeric; the mean of spike prior for log gamma. Default is -3.
pr_spike_sd	Numeric; the standard deviation of spike prior for log gamma. Default is 1.
pr_slab_mean	Numeric; the mean of spike prior for log gamma. Default is 0.5.
pr_slab_sd	Numeric; the standard deviation of spike prior for log gamma. Default is is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_xi_a — Numeric; the first shape parameter of beta prior for latent variable xi. Default is 1.

pr_xi_b — Numeric; the second shape parameter of beta prior for latent variable xi. Default is 1.

missing.val — Numeric; a number to replace missing values. Default is 99.

verbose — Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.

fix_theta_sd — Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

adapt — List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm1pl_mar_ss models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term:

$$logit(P(Y_{j,i} = 1|\theta_j, \beta_i, \gamma, z_j, w_i)) = \theta_j + \beta_i - \gamma||z_j - w_i||$$

Under the assumption of missing at random, the model takes the missing element into consideration in the sampling procedure. For the details of missing at random assumption and data augmentation, see References. `lsirm1pl_mar_ss` model include model selection approach based on spike-and-slab priors for log gamma. For detail of spike-and-slab priors, see References.

Value

`lsirm1pl_mar_ss` returns an object of list containing the following components:

`data`	Data frame or matrix containing the variables in the model.
`missing.val`	A number to replace missing values.
`bic`	Numeric value with the corresponding BIC.
`mcmc_inf`	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
`map_inf`	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
`beta_estimate`	Posterior estimates of the beta parameter.
`theta_estimate`	Posterior estimates of the theta parameter.
`sigma_theta_estimate`	Posterior estimates of the standard deviation of theta.
`gamma_estimate`	posterior estimates of gamma parameter.
`z_estimate`	Posterior estimates of the z parameter.
`w_estimate`	Posterior estimates of the w parameter.
`beta`	Posterior samples of the beta parameter.
`theta`	Posterior samples of the theta parameter.
`gamma`	Posterior samples of the gamma parameter.
`theta_sd`	Posterior samples of the standard deviation of theta.
`z`	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`w`	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`z_raw`	Posterior samples of the z parameter without procrustes matching.
`w_raw`	Posterior samples of the w parameter without procrustes matching.
`pi`	Posterior samples of phi which is indicator of spike and slab prior. If phi is 1, log gamma follows the slab prior, otherwise follows the spike prior.
`imp`	Imputation for missing Values using posterior samples.
`accept_beta`	Acceptance ratio for the beta parameter.
`accept_theta`	Acceptance ratio for the theta parameter.

accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.
pi_estimate	Posterior estimation of phi. inclusion probability of gamma. if estimation of phi is less than 0.5, choose Rasch model with gamma = 0, otherwise latent space model with gamma > 0.
imp_estimate	Probability of imputing a missing value with 1.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons. Ishwaran, H., & Rao, J. S. (2005). Spike and slab variable selection: frequentist and Bayesian strategies. The Annals of Statistics, 33(2), 730-773.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm1pl_mar_ss(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = TRUE, fixed_gamma = FALSE,
missing_data = 'mar', missing = 99))
```

`lsirm1pl_mcar` *1PL LSIRM for missing completely at random data.*

Description

lsirm1pl_mcar is used to fit 1PL LSIRM in incomplete data assumed to be missing completely at random. lsirm1pl_mcar factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while ignoring the missing element under the assumption of missing completely at random. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_mcar(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.
`nburn`	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
`nthin`	Integer; the number of MCMC iterations to thin. Default is 5.
`nprint`	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
`jump_beta`	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
`jump_theta`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_gamma`	Numeric; the jumping rule for the gamma proposal density. Default is 0.2.
`jump_z`	Numeric; the jumping rule for the z proposal density. Default is 0.5.
`jump_w`	Numeric; the jumping rule for the w proposal density. Default is 0.5.
`pr_mean_beta`	Numeric; the mean of the normal prior for beta. Default is 0.

`pr_sd_beta` Numeric; the standard deviation of the normal prior for beta. Default is 1.

`pr_mean_theta` Numeric; the mean of the normal prior for theta. Default is 0.

`pr_sd_theta` Numeric; the standard deviation of the normal prior for theta. Default is 1.

`pr_mean_gamma` Numeric; mean of log normal prior for gamma. Default is 0.5.

`pr_sd_gamma` Numeric; standard deviation of log normal prior for gamma. Default is 1.

`pr_a_theta` Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

`pr_b_theta` Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

`missing.val` Numeric; A number to replace missing values. Default is 99.

`verbose` Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.

`fix_theta_sd` Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

`adapt` List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.
- `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is `0.44`.

- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

Details

`lsirm1pl_mcar` models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term:

$$logit(P(Y_{j,i} = 1|\theta_j, \beta_i, \gamma, z_j, w_i)) = \theta_j + \beta_i - \gamma||z_j - w_i||$$

Under the assumption of missing completely at random, the model ignores the missing element in doing inference. For the details of missing completely at random assumption and data augmentation, see References.

Value

`lsirm1pl_mcar` returns an object of list containing the following components:

`data`	A data frame or matrix containing the variables used in the model.
`missing.val`	A number to replace missing values.
`bic`	Numeric value with the corresponding BIC.
`mcmc_inf`	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
`map_inf`	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
`beta_estimate`	Posterior estimates of the beta parameter.
`theta_estimate`	Posterior estimates of the theta parameter.
`sigma_theta_estimate`	Posterior estimates of the standard deviation of theta.
`gamma_estimate`	Posterior estimates of gamma parameter.
`z_estimate`	Posterior estimates of the z parameter.
`w_estimate`	Posterior estimates of the w parameter.
`beta`	Posterior samples of the beta parameter.
`theta`	Posterior samples of the theta parameter.
`theta_sd`	Posterior samples of the standard deviation of theta.
`gamma`	Posterior samples of the gamma parameter.
`z`	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`w`	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`z_raw`	Posterior samples of the z parameter without procrustes matching.

w_raw	Posterior samples of the w parameter without procrustes matching.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm1pl_mcar(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = FALSE, fixed_gamma = FALSE,
missing_data ='mcar', missing.val = NA))
```

---

lsirm1pl_mcar_ss — *1PL LSIRM with model selection approach for missing completely at random data.*

Description

lsirm1pl_mcar_ss is used to fit 1PL LSIRM with model selection approach based on spike-and-slab priors in incomplete data assumed to be missing completely at random. lsirm1pl_mcar_ss factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while ignoring the missing element under the assumption of missing completely at random. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_mcar_ss(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_xi_a = 1,
  pr_xi_b = 1,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.
`nburn`	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
`nthin`	Integer; the number of MCMC iterations to thin. Default is 5.
`nprint`	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
`jump_beta`	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
`jump_theta`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_gamma`	Numeric; the jumping rule for the theta proposal density. Default is 1.

jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_spike_mean	Numeric; the mean of spike prior for log gamma. Default is -3.
pr_spike_sd	Numeric; the standard deviation of spike prior for log gamma. Default is 1.
pr_slab_mean	Numeric; the mean of spike prior for log gamma. Default is 0.5.
pr_slab_sd	Numeric; the standard deviation of spike prior for log gamma. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_xi_a	Numeric; the first shape parameter of beta prior for latent variable xi. Default is 1.
pr_xi_b	Numeric; the second shape parameter of beta prior for latent variable xi. Default is 1.
missing.val	Numeric; a number to replace missing values. Default is 99.
verbose	Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
adapt	List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include: • use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE. • adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100. • adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.

- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm1pl_mcar_ss models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term:

$$logit(P(Y_{j,i} = 1|\theta_j, \beta_i, \gamma, z_j, w_i)) = \theta_j + \beta_i - \gamma||z_j - w_i||$$

Under the assumption of missing completely at random, the model ignores the missing element in doing inference. For the details of missing completely at random assumption and data augmentation, see References. lsirm1pl_mcar_ss model include model selection approach based on spike-and-slab priors for log gamma. For detail of spike-and-slab priors, see References.

Value

lsirm1pl_mcar_ss returns an object of list containing the following components:

data	Data frame or matrix containing the variables in the model.
missing.val	A number to replace missing values.
bic	Numeric value with the corresponding BIC.
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
gamma_estimate	posterior estimates of gamma parameter.
z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
beta	Posterior samples of the beta parameter.

theta	Posterior samples of the theta parameter.
gamma	Posterior samples of the gamma parameter.
theta_sd	Posterior samples of the standard deviation of theta.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
pi	Posterior samples of phi which is indicator of spike and slab prior. If phi is 1, log gamma follows the slab prior, otherwise follows the spike prior.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons. Ishwaran, H., & Rao, J. S. (2005). Spike and slab variable selection: frequentist and Bayesian strategies. The Annals of Statistics, 33(2), 730-773.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm1pl_mcar_ss(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = TRUE, fixed_gamma = FALSE,
missing_data ='mcar', missing.val = NA))
```

`lsirm1pl_normal_fixed_gamma` *1PL LSIRM fixing gamma to 1 with normal likelihood*

Description

lsirm1pl_normal_fixed_gamma is used to fit 1PL LSIRM for continuous variable with gamma fixed to 1. lsirm1pl_normal_fixed_gamma factorizes continuous item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while ignoring the missing element under the assumption of missing at random. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_normal_fixed_gamma(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.

nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_a_eps	Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
pr_b_eps	Numeric; the scale parameter of inverse gamma prior for variance of data likelihood Default is 0.001.
verbose	Logical; If TRUE, MCMC samples are printed for each `nprint`. default value is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
adapt	List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include: • `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`. • `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.

- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm1pl_normal_fixed_gamma models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space:

$$Y_{j,i} = \theta_j + \beta_i - ||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$.

Value

lsirm1pl_normal_fixed_gamma returns an object of list containing the following components:

data	A data frame or matrix containing the variables used in the model.
bic	A numeric value representing the Bayesian Information Criterion (BIC).
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.

theta_sd	Posterior samples of the standard deviation of theta.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
sigma_estimate	Posterior estimates of the standard deviation.
sigma	Posterior samples of the standard deviation.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm1pl_normal_fixed_gamma(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = FALSE, fixed_gamma = TRUE,
                      niter = 1000, nburn = 500))
```

---

lsirm1pl_normal_fixed_gamma_mar

1PL LSIRM fixing gamma to 1 with normal likelihood for missing at random data.

Description

lsirm1pl_normal_fixed_gamma_mar is used to fit 1PL LSIRM for continuous variable with gamma fixed to 1 in incomplete data assumed to be missing at random.

lsirm1pl_normal_fixed_gamma_mar factorizes continuous item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while considering the missing element under the assumption of missing at random. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_normal_fixed_gamma_mar(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.
`nburn`	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
`nthin`	Integer; the number of MCMC iterations to thin. Default is 5.
`nprint`	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
`jump_beta`	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
`jump_theta`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_z`	Numeric; the jumping rule for the z proposal density. Default is 0.5.
`jump_w`	Numeric; the jumping rule for the w proposal density. Default is 0.5.
`pr_mean_beta`	Numeric; the mean of the normal prior for beta. Default is 0.
`pr_sd_beta`	Numeric; the standard deviation of the normal prior for beta. Default is 1.
`pr_mean_theta`	Numeric; the mean of the normal prior for theta. Default is 0.

pr_sd_theta — Numeric; the standard deviation of the normal prior for theta. Default is 1.

pr_a_theta — Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_b_theta — Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_a_eps — Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

pr_b_eps — Numeric; the scale parameter of inverse gamma prior for variance of data likelihood Default is 0.001.

missing.val — Numeric; a number to replace missing values. Default is 99.

verbose — Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.

fix_theta_sd — Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

adapt — List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^α. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^α. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.
- `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is `0.44`.

- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm1pl_normal_fixed_gamma_mar models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space:

$$Y_j, i = \theta_j + \beta_i - ||z_j - w_i|| + e_{ji}$$

where the error $e_j i\ N(0, \sigma^2)$ Under the assumption of missing at random, the model takes the missing element into consideration in the sampling procedure. For the details of missing at random assumption and data augmentation, see References.

Value

lsirm1pl_normal_fixed_gamma_mar returns an object of list containing the following components:

data	Data frame or matrix containing the variables in the model.
missing.val	A number to replace missing values.
bic	Numeric value with the corresponding BIC.
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
imp_estimate	Probability of imputing a missing value with 1.
beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.
theta_sd	Posterior samples of the standard deviation of theta.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.

`w_raw`	Posterior samples of the w parameter without procrustes matching.
`imp`	Imputation for missing Values using posterior samples.
`accept_beta`	Acceptance ratio for the beta parameter.
`accept_theta`	Acceptance ratio for the theta parameter.
`accept_z`	Acceptance ratio for the z parameter.
`accept_w`	Acceptance ratio for the w parameter.
`sigma_estimate`	Posterior estimates of the standard deviation.
`sigma`	Posterior samples of the standard deviation.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm1pl_normal_fixed_gamma_mar(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = FALSE, fixed_gamma = TRUE,
                      niter = 1000, nburn = 500,
missing_data = "mar", missing.val = NA))
```

`lsirm1pl_normal_fixed_gamma_mcar`

1PL LSIRM fixing gamma to 1 with normal likelihood for missing completely at random data.

Description

lsirm1pl_normal_fixed_gamma_mcar is used to fit 1PL LSIRM for continuous variable with gamma fixed to 1 in incomplete data assumed to be missing completely at random.

lsirm1pl_normal_fixed_gamma_mcar factorizes continuous item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while ignoring the missing element under the assumption of missing completely at random. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_normal_fixed_gamma_mcar(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.
`nburn`	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
`nthin`	Integer; the number of MCMC iterations to thin. Default is 5.
`nprint`	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
`jump_beta`	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
`jump_theta`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_z`	Numeric; the jumping rule for the z proposal density. Default is 0.5.
`jump_w`	Numeric; the jumping rule for the w proposal density. Default is 0.5.
`pr_mean_beta`	Numeric; the mean of the normal prior for beta. Default is 0.
`pr_sd_beta`	Numeric; the standard deviation of the normal prior for beta. Default is 1.
`pr_mean_theta`	Numeric; the mean of the normal prior for theta. Default is 0.

pr_sd_theta
: Numeric; the standard deviation of the normal prior for theta. Default is 1.

pr_a_theta
: Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_b_theta
: Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_a_eps
: Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

pr_b_eps
: Numeric; the scale parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

missing.val
: Numeric; a number to replace missing values. Default is 99.

verbose
: Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.

fix_theta_sd
: Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

adapt
: List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.
- `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is `0.44`.

- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm1pl_normal_fixed_gamma_mcar models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space:

$$Y_{j,i} = \theta_j + \beta_i - ||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$ Under the assumption of missing completely at random, the model ignores the missing element in doing inference. For the details of missing completely at random assumption and data augmentation, see References.

Value

lsirm1pl_normal_fixed_gamma_mcar returns an object of list containing the following components:

data	Data frame or matrix containing the variables in the model.
missing.val	A number to replace missing values.
bic	Numeric value with the corresponding BIC.
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.
theta_sd	Posterior samples of the standard deviation of theta.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.

accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
sigma_estimate	Posterior estimates of the standard deviation.
sigma	Posterior samples of the standard deviation.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm1pl_normal_fixed_gamma_mcar(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = FALSE, fixed_gamma = TRUE,
                      niter = 1000, nburn = 500,
missing_data = "mcar", missing.val = NA))
```

---

lsirm1pl_normal_mar *1PL LSIRM with normal likelihood for missing at random data.*

Description

lsirm1pl_normal_mar is used to fit LSIRM for continuous variable with 1pl in incomplete data assumed to be missing at random. lsirm1pl_normal_mar factorizes continuous item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while considering the missing element under the assumption of missing at random. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_normal_mar(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
```

```
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.
`nburn`	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
`nthin`	Integer; the number of MCMC iterations to thin. Default is 5.
`nprint`	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
`jump_beta`	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
`jump_theta`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_gamma`	Numeric; the jumping rule for the gamma proposal density. Default is 0.2
`jump_z`	Numeric; the jumping rule for the z proposal density. Default is 0.5.
`jump_w`	Numeric; the jumping rule for the w proposal density. Default is 0.5.
`pr_mean_beta`	Numeric; the mean of the normal prior for beta. Default is 0.
`pr_sd_beta`	Numeric; the standard deviation of the normal prior for beta. Default is 1.
`pr_mean_theta`	Numeric; the mean of the normal prior for theta. Default is 0.
`pr_sd_theta`	Numeric; the standard deviation of the normal prior for theta. Default is 1.
`pr_mean_gamma`	Numeric; mean of log normal prior for gamma. Default is 0.5.
`pr_sd_gamma`	Numeric; standard deviation of log normal prior for gamma. Default is 1.

pr_a_theta
: Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_b_theta
: Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_a_eps
: Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

pr_b_eps
: Numeric; the scale parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

missing.val
: Numeric; a number to replace missing values. Default is 99.

verbose
: Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.

fix_theta_sd
: Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

adapt
: List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

  - `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
  - `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
  - `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
  - `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.
  - `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
  - `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is `0.44`.
  - `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.

- `target_accept_beta/theta/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

Details

`lsirm1pl_normal_mar` models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term:

$$Y_{j,i} = \theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$ Under the assumption of missing at random, the model takes the missing element into consideration in the sampling procedure. For the details of missing at random assumption and data augmentation, see References.

Value

`lsirm1pl_normal_mar` returns an object of list containing the following components:

`data`	Data frame or matrix containing the variables in the model.
`missing.val`	A number to replace missing values.
`bic`	Numeric value with the corresponding BIC.
`mcmc_inf`	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
`map_inf`	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
`beta_estimate`	Posterior estimates of the beta parameter.
`theta_estimate`	Posterior estimates of the theta parameter.
`sigma_theta_estimate`	Posterior estimates of the standard deviation of theta.
`gamma_estimate`	posterior estimates of gamma parameter.
`z_estimate`	Posterior estimates of the z parameter.
`w_estimate`	Posterior estimates of the w parameter.
`imp_estimate`	Posterior mean imputation for each missing cell.
`beta`	Posterior samples of the beta parameter.
`theta`	Posterior samples of the theta parameter.
`gamma`	Posterior samples of the gamma parameter.
`theta_sd`	Posterior samples of the standard deviation of theta.
`z`	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`w`	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`z_raw`	Posterior samples of the z parameter without procrustes matching.
`w_raw`	Posterior samples of the w parameter without procrustes matching.

imp	Imputation for missing Values using posterior samples.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.
sigma_estimate	Posterior estimates of the standard deviation.
sigma	Posterior samples of the standard deviation.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm1pl_normal_mar(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = FALSE, fixed_gamma = FALSE,
                      niter = 1000, nburn = 500,
missing_data ='mar', missing.val = NA))
```

---

lsirm1pl_normal_mar_ss

1PL LSIRM with normal likelihood and model selection approach for missing at random data.

Description

lsirm1pl_normal_mar_ss is used to fit 1PL LSIRM with model selection approach based on spike-and-slab priors for continuous variable with 1pl in incomplete data assumed to be missing at random. lsirm1pl_normal_mar_ss factorizes continuous item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while considering the missing element under the assumption of missing at random. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_normal_mar_ss(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  pr_xi_a = 0.001,
  pr_xi_b = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.

jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_spike_mean	Numeric; the mean of spike prior for log gamma. Default is -3.
pr_spike_sd	Numeric; the standard deviation of spike prior for log gamma. Default is 1.
pr_slab_mean	Numeric; the mean of spike prior for log gamma. Default is 0.5.
pr_slab_sd	Numeric; the standard deviation of spike prior for log gamma. Default is is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_a_eps	Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
pr_b_eps	Numeric; the scale parameter of inverse gamma prior for variance of data likelihood Default is 0.001.
pr_xi_a	Numeric; the first shape parameter of beta prior for latent variable xi. Default is 1.
pr_xi_b	Numeric; the second shape parameter of beta prior for latent variable xi. Default is 1.
missing.val	Numeric; a number to replace missing values. Default is 99.
verbose	Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
adapt	List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.
- `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is `0.44`.
- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

Details

`lsirm1pl_normal_mar_ss` models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term:

$$Y_{j,i} = \theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$ Under the assumption of missing at random, the model takes the missing element into consideration in the sampling procedure. For the details of missing at random assumption and data augmentation, see References. `lsirm1pl_normal_mar_ss` model include model selection approach based on spike-and-slab priors for log gamma. For detail of spike-and-slab priors, see References.

Value

`lsirm1pl_normal_mar_ss` returns an object of list containing the following components:

`data`	Data frame or matrix containing the variables in the model.
`missing.val`	A number to replace missing values.
`bic`	Numeric value with the corresponding BIC.
`mcmc_inf`	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
`map_inf`	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.

beta_estimate Posterior estimates of the beta parameter.

theta_estimate Posterior estimates of the theta parameter.

sigma_theta_estimate
  Posterior estimates of the standard deviation of theta.

gamma_estimate posterior estimates of gamma parameter.

z_estimate Posterior estimates of the z parameter.

w_estimate Posterior estimates of the w parameter.

beta Posterior samples of the beta parameter.

theta Posterior samples of the theta parameter.

gamma Posterior samples of the gamma parameter.

theta_sd Posterior samples of the standard deviation of theta.

z Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.

w Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.

z_raw Posterior samples of the z parameter without procrustes matching.

w_raw Posterior samples of the w parameter without procrustes matching.

pi Posterior samples of phi which is indicator of spike and slab prior. If phi is 1, log gamma follows the slab prior, otherwise follows the spike prior.

imp Imputation for missing Values using posterior samples.

accept_beta Acceptance ratio for the beta parameter.

accept_theta Acceptance ratio for the theta parameter.

accept_z Acceptance ratio for the z parameter.

accept_w Acceptance ratio for the w parameter.

accept_gamma Acceptance ratio for the gamma parameter.

pi_estimate Posterior estimation of phi. inclusion probability of gamma. if estimation of phi is less than 0.5, choose Rasch model with gamma = 0, otherwise latent space model with gamma > 0.

imp_estimate Probability of imputing a missing value with 1.

sigma_estimate Posterior estimates of the standard deviation.

sigma Posterior samples of the standard deviation.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons. Ishwaran, H., & Rao, J. S. (2005). Spike and slab variable selection: frequentist and Bayesian strategies. The Annals of Statistics, 33(2), 730-773.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm1pl_normal_mar_ss(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = TRUE, fixed_gamma = FALSE,
                      niter = 1000, nburn = 500,
missing_data = 'mar', missing = 99))
```

`lsirm1pl_normal_mcar` *1PL LSIRM with normal likelihood for missing completely at random data.*

Description

lsirm1pl_normal_mcar is used to fit LSIRM with 1pl in incomplete data assumed to be missing completely at random. lsirm1pl_normal_mcar factorizes continuous item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while ignoring the missing element under the assumption of missing completely at random. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_normal_mcar(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
```

```
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.
`nburn`	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
`nthin`	Integer; the number of MCMC iterations to thin. Default is 5.
`nprint`	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
`jump_beta`	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
`jump_theta`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_gamma`	Numeric; the jumping rule for the gamma proposal density. Default is 0.2
`jump_z`	Numeric; the jumping rule for the z proposal density. Default is 0.5.
`jump_w`	Numeric; the jumping rule for the w proposal density. Default is 0.5.
`pr_mean_beta`	Numeric; the mean of the normal prior for beta. Default is 0.
`pr_sd_beta`	Numeric; the standard deviation of the normal prior for beta. Default is 1.
`pr_mean_theta`	Numeric; the mean of the normal prior for theta. Default is 0.
`pr_sd_theta`	Numeric; the standard deviation of the normal prior for theta. Default is 1.
`pr_mean_gamma`	Numeric; mean of log normal prior for gamma. Default is 0.5.
`pr_sd_gamma`	Numeric; standard deviation of log normal prior for gamma. Default is 1.
`pr_a_theta`	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
`pr_b_theta`	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
`pr_a_eps`	Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
`pr_b_eps`	Numeric; the scale parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

missing.val Numeric; a number to replace missing values. Default is 99.

verbose Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.

fix_theta_sd Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

adapt List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm1pl_normal_mcar models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term:

$$Y_{j,i} = \theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$ Under the assumption of missing completely at random, the model ignores the missing element in doing inference. For the details of missing at random assumption and data augmentation, see References.

Value

`lsirm1pl_normal_mcar` returns an object of list containing the following components:

`data`	A data frame or matrix containing the variables used in the model.
`missing.val`	A number to replace missing values.
`bic`	Numeric value with the corresponding BIC.
`mcmc_inf`	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
`map_inf`	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
`beta_estimate`	Posterior estimates of the beta parameter.
`theta_estimate`	Posterior estimates of the theta parameter.
`sigma_theta_estimate`	Posterior estimates of the standard deviation of theta.
`gamma_estimate`	Posterior estimates of gamma parameter.
`z_estimate`	Posterior estimates of the z parameter.
`w_estimate`	Posterior estimates of the w parameter.
`beta`	Posterior samples of the beta parameter.
`theta`	Posterior samples of the theta parameter.
`theta_sd`	Posterior samples of the standard deviation of theta.
`gamma`	Posterior samples of the gamma parameter.
`z`	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`w`	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`z_raw`	Posterior samples of the z parameter without procrustes matching.
`w_raw`	Posterior samples of the w parameter without procrustes matching.
`accept_beta`	Acceptance ratio for the beta parameter.
`accept_theta`	Acceptance ratio for the theta parameter.
`accept_z`	Acceptance ratio for the z parameter.
`accept_w`	Acceptance ratio for the w parameter.
`accept_gamma`	Acceptance ratio for the gamma parameter.
`sigma_estimate`	Posterior estimates of the standard deviation.
`sigma`	Posterior samples of the standard deviation.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm1pl_normal_mcar(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = FALSE, fixed_gamma = FALSE,
                      niter = 1000, nburn = 500,
missing_data ='mcar', missing.val = NA))
```

---

lsirm1pl_normal_mcar_ss

1PL LSIRM with normal likelihood and model selection approach for missing completely at random data.

Description

lsirm1pl_normal_mcar_ss is used to fit LSIRM with model selection approach based on spike-and-slab priors for continuous variable with 1pl in incomplete data assumed to be missing completely at random. lsirm1pl_normal_mcar_ss factorizes continuous item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while ignoring the missing element under the assumption of missing completely at random. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_normal_mcar_ss(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
```

```
    jump_gamma = 1,
    jump_z = 0.5,
    jump_w = 0.5,
    pr_mean_beta = 0,
    pr_sd_beta = 1,
    pr_mean_theta = 0,
    pr_sd_theta = 1,
    pr_spike_mean = -3,
    pr_spike_sd = 1,
    pr_slab_mean = 0.5,
    pr_slab_sd = 1,
    pr_a_theta = 0.001,
    pr_b_theta = 0.001,
    pr_a_eps = 0.001,
    pr_b_eps = 0.001,
    pr_xi_a = 0.001,
    pr_xi_b = 0.001,
    missing.val = NA,
    verbose = FALSE,
    fix_theta_sd = FALSE,
    adapt = NULL
  )
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.
`nburn`	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
`nthin`	Integer; the number of MCMC iterations to thin. Default is 5.
`nprint`	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
`jump_beta`	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
`jump_theta`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_gamma`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_z`	Numeric; the jumping rule for the z proposal density. Default is 0.5.
`jump_w`	Numeric; the jumping rule for the w proposal density. Default is 0.5.
`pr_mean_beta`	Numeric; the mean of the normal prior for beta. Default is 0.
`pr_sd_beta`	Numeric; the standard deviation of the normal prior for beta. Default is 1.
`pr_mean_theta`	Numeric; the mean of the normal prior for theta. Default is 0.
`pr_sd_theta`	Numeric; the standard deviation of the normal prior for theta. Default is 1.

pr_spike_mean Numeric; the mean of spike prior for log gamma. Default is -3.

pr_spike_sd Numeric; the standard deviation of spike prior for log gamma. Default is 1.

pr_slab_mean Numeric; the mean of spike prior for log gamma. Default is 0.5.

pr_slab_sd Numeric; the standard deviation of spike prior for log gamma. Default is is 1.

pr_a_theta Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_b_theta Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_a_eps Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

pr_b_eps Numeric; the scale parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

pr_xi_a Numeric; the first shape parameter of beta prior for latent variable xi. Default is 1.

pr_xi_b Numeric; the second shape parameter of beta prior for latent variable xi. Default is 1.

missing.val Numeric; a number to replace missing values. Default is 99.

verbose Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.

fix_theta_sd Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

adapt List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.

- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm1pl_normal_mcar_ss models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term:

$$Y_{j,i} = \theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$. Under the assumption of missing completely at random, the model ignores the missing element in doing inference. For the details of missing at random assumption and data augmentation, see References. lsirm1pl_normal_mcar_ss model include model selection approach based on spike-and-slab priors for log gamma. For detail of spike-and-slab priors, see References.

Value

lsirm1pl_normal_mcar_ss returns an object of list containing the following components:

data	Data frame or matrix containing the variables in the model.
missing.val	A number to replace missing values.
bic	Numeric value with the corresponding BIC.
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
gamma_estimate	posterior estimates of gamma parameter.
z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.

beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.
gamma	Posterior samples of the gamma parameter.
theta_sd	Posterior samples of the standard deviation of theta.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
pi	Posterior samples of phi which is indicator of spike and slab prior. If phi is 1, log gamma follows the slab prior, otherwise follows the spike prior.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.
sigma_estimate	Posterior estimates of the standard deviation.
sigma	Posterior samples of the standard deviation.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons. Ishwaran, H., & Rao, J. S. (2005). Spike and slab variable selection: frequentist and Bayesian strategies. The Annals of Statistics, 33(2), 730-773.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm1pl_normal_mcar_ss(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = TRUE, fixed_gamma = FALSE,
                      niter = 1000, nburn = 500,
missing_data ='mcar', missing.val = NA))
```

lsirm1pl_normal_o — *1PL LSIRM with normal likelihood.*

Description

lsirm1pl_normal_o is used to fit LSIRM for continuous variable with 1pl. lsirm1pl_normal_o factorizes continuous item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_normal_o(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.

nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the gamma proposal density. Default is 0.2
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_mean_gamma	Numeric; mean of log normal prior for gamma. Default is 0.5.
pr_sd_gamma	Numeric; standard deviation of log normal prior for gamma. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_a_eps	Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
pr_b_eps	Numeric; the scale parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
verbose	Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
adapt	List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include: • `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.

- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^α. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^α. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm1pl_normal_o models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term:

$$Y_{j,i} = \theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$.

Value

lsirm1pl_normal_o returns an object of list containing the following components:

data	Data frame or matrix containing the variables used in the model.
bic	A numeric value representing the Bayesian Information Criterion (BIC).
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
gamma_estimate	Posterior estimates of gamma parameter.
z_estimate	Posterior estimates of the z parameter.

w_estimate	Posterior estimates of the w parameter.
beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.
theta_sd	Posterior samples of the standard deviation of theta.
gamma	Posterior samples of the gamma parameter.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.
sigma_estimate	Posterior estimates of the standard deviation.
sigma	Posterior samples of the standard deviation.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

lsirm_result <- lsirm1pl_normal_o(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = FALSE, fixed_gamma = FALSE,
                      niter = 1000, nburn = 500))
```

`lsirm1pl_normal_ss` *1PL LSIRM with normal likelihood and model selection approach.*

Description

lsirm1pl_normal_ss is used to fit LSIRM with model selection approach based on spike-and-slab priors for continuous variable with 1pl. LSIRM factorizes continuous item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_normal_ss(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  pr_xi_a = 0.001,
  pr_xi_b = 0.001,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.

jump_gamma — Numeric; the jumping rule for the theta proposal density. Default is 1.

jump_z — Numeric; the jumping rule for the z proposal density. Default is 0.5.

jump_w — Numeric; the jumping rule for the w proposal density. Default is 0.5.

pr_mean_beta — Numeric; the mean of the normal prior for beta. Default is 0.

pr_sd_beta — Numeric; the standard deviation of the normal prior for beta. Default is 1.

pr_mean_theta — Numeric; the mean of the normal prior for theta. Default is 0.

pr_sd_theta — Numeric; the standard deviation of the normal prior for theta. Default is 1.

pr_spike_mean — Numeric; mean of spike prior for log gamma default value is -3.

pr_spike_sd — Numeric; standard deviation of spike prior for log gamma default value is 1.

pr_slab_mean — Numeric; mean of spike prior for log gamma default value is 0.5.

pr_slab_sd — Numeric; standard deviation of spike prior for log gamma default value is 1.

pr_a_theta — Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_b_theta — Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_a_eps — Numeric; shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

pr_b_eps — Numeric; scale parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

pr_xi_a — Numeric; first shape parameter of beta prior for latent variable xi. Default is 1.

pr_xi_b — Numeric; second shape parameter of beta prior for latent variable xi. Default is 1.

verbose — Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.

fix_theta_sd — Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

adapt — List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.

- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm1pl_normal_ss models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term:

$$Y_{j,i} = \theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$. lsrm1pl_noraml_ss model include model selection approach based on spike-and-slab priors for log gamma. For detail of spike-and-slab priors, see References.

Value

lsirm1pl_normal_ss returns an object of list containing the following components:

data	Data frame or matrix containing the variables in the model.
bic	Numeric value with the corresponding BIC.
mcmc_inf	number of mcmc iteration, burn-in periods, and thinning intervals.
map_inf	value of log maximum a posterior and iteration number which have log maximum a posterior.
beta_estimate	posterior estimation of beta.
theta_estimate	posterior estimation of theta.
sigma_theta_estimate	posterior estimation of standard deviation of theta.
sigma_estimate	posterior estimation of standard deviation.
gamma_estimate	posterior estimation of gamma.
z_estimate	posterior estimation of z.
w_estimate	posterior estimation of w.

pi_estimate	posterior estimation of phi. inclusion probability of gamma. if estimation of phi is less than 0.5, choose Rasch model with gamma = 0, otherwise latent space model with gamma > 0.
beta	posterior samples of beta.
theta	posterior samples of theta.
theta_sd	posterior samples of standard deviation of theta.
sigma	posterior samples of standard deviation.
gamma	posterior samples of gamma.
z	posterior samples of z. The output is 3-dimensional matrix with last axis represent the dimension of latent space.
w	posterior samples of w. The output is 3-dimensional matrix with last axis represent the dimension of latent space.
z_raw	posterior samples of z without procrustes matching.
w_raw	posterior samples of w without procrustes matching.
pi	posterior samples of phi which is indicator of spike and slab prior. If phi is 1, log gamma follows the slab prior, otherwise follows the spike prior.
accept_beta	accept ratio of beta.
accept_theta	accept ratio of theta.
accept_w	accept ratio of w.
accept_z	accept ratio of z.
accept_gamma	accept ratio of gamma.

References

Ishwaran, H., & Rao, J. S. (2005). Spike and slab variable selection: Frequentist and Bayesian strategies (Vol. 33). The Annals of Statistics

Examples

```
# generate example (continuous) item response matrix
data <- matrix(rnorm(500, mean = 0, sd = 1), ncol = 10, nrow = 50)

lsirm_result <- lsirm1pl_normal_ss(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = TRUE, fixed_gamma = FALSE,
                      niter = 1000, nburn = 500))
```

lsirm1pl_o *1PL LSIRM.*

Description

lsirm1pl_o is used to fit 1PL LSIRM. lsirm1pl_o factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_o(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.

nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the gamma proposal density. Default is 0.2.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_mean_gamma	Numeric; mean of log normal prior for gamma. Default is 0.5.
pr_sd_gamma	Numeric; standard deviation of log normal prior for gamma. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
verbose	Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
adapt	List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include: • `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`. • `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`. • `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5–2.0`.

- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm1pl_o models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term:

$$logit(P(Y_{j,i} = 1|\theta_j, \beta_i, \gamma, z_j, w_i)) = \theta_j + \beta_i - \gamma||z_j - w_i||$$

Value

lsirm1pl_o returns an object of list containing the following components:

data	Data frame or matrix containing the variables used in the model.
bic	A numeric value representing the Bayesian Information Criterion (BIC).
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
gamma_estimate	Posterior estimates of gamma parameter.
z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.
theta_sd	Posterior samples of the standard deviation of theta.
gamma	Posterior samples of the gamma parameter.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.

w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

lsirm_result <- lsirm1pl_o(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = FALSE, fixed_gamma = FALSE))
```

---

lsirm1pl_ss *1PL LSIRM with model selection approach.*

Description

lsirm1pl_ss is used to fit 1PL LSIRM with model selection approach based on spike-and-slab priors. LSIRM factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm1pl_ss(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
```

```
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_xi_a = 1,
  pr_xi_b = 1,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  adapt = NULL
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.
`nburn`	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
`nthin`	Integer; the number of MCMC iterations to thin. Default is 5.
`nprint`	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
`jump_beta`	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
`jump_theta`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_gamma`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_z`	Numeric; the jumping rule for the z proposal density. Default is 0.5.
`jump_w`	Numeric; the jumping rule for the w proposal density. Default is 0.5.
`pr_mean_beta`	Numeric; the mean of the normal prior for beta. Default is 0.
`pr_sd_beta`	Numeric; the standard deviation of the normal prior for beta. Default is 1.
`pr_mean_theta`	Numeric; the mean of the normal prior for theta. Default is 0.
`pr_sd_theta`	Numeric; the standard deviation of the normal prior for theta. Default is 1.
`pr_spike_mean`	Numeric; the mean of spike prior for log gamma. Default is -3.
`pr_spike_sd`	Numeric; the standard deviation of spike prior for log gamma. Default is 1.
`pr_slab_mean`	Numeric; the mean of spike prior for log gamma. Default is 0.5.
`pr_slab_sd`	Numeric; the standard deviation of spike prior for log gamma. Default is is 1.
`pr_a_theta`	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_b_theta — Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_xi_a — Numeric; the first shape parameter of beta prior for latent variable xi. Default is 1.

pr_xi_b — Numeric; the second shape parameter of beta prior for latent variable xi. Default is 1.

verbose — Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.

fix_theta_sd — Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

adapt — List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.
- `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is `0.44`.
- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

Details

lsirm1pl_ss models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term:

$$logit(P(Y_{j,i} = 1|\theta_j, \beta_i, \gamma, z_j, w_i)) = \theta_j + \beta_i - \gamma||z_j - w_i||$$

lsirm1pl_ss model include model selection approach based on spike-and-slab priors for log gamma. For detail of spike-and-slab priors, see References.

Value

lsirm1pl_ss returns an object of list containing the following components:

data	Data frame or matrix containing the variables used in the model.
bic	A numeric value representing the Bayesian Information Criterion (BIC).
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
gamma_estimate	Posterior estimates of gamma parameter.
z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.
theta_sd	Posterior samples of the standard deviation of theta.
gamma	Posterior samples of the gamma parameter.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.

pi_estimate	Posterior estimation of phi. inclusion probability of gamma. if estimation of phi is less than 0.5, choose Rasch model with gamma = 0, otherwise latent space model with gamma > 0.
pi	Posterior samples of phi which is indicator of spike and slab prior. If phi is 1, log gamma follows the slab prior, otherwise follows the spike prior.

References

Ishwaran, H., & Rao, J. S. (2005). Spike and slab variable selection: Frequentist and Bayesian strategies (Vol. 33). The Annals of Statistics

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

lsirm_result <- lsirm1pl_ss(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm1pl(spikenslab = TRUE, fixed_gamma = FALSE))
```

---

lsirm2pl *Fit a 2pl LSIRM for binary and continuous item response data*

Description

lsirm2pl integrates all functions related to 2PL LSIRM. Various 2PL LSIRM function can be used by setting the spikenslab, fixed_gamma, and missing_data arguments.

This function can be used regardless of the data type, providing a unified approach to model fitting.

Usage

```
lsirm2pl(
  data,
  spikenslab = FALSE,
  fixed_gamma = FALSE,
  missing_data = NA,
  chains = 1,
  multicore = 1,
  seed = NA,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
```

```
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL,
  ...
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
spikenslab	Logical; specifies whether to use a model selection approach. Default is FALSE.
fixed_gamma	Logical; indicates whether to fix gamma at 1. Default is FALSE.
missing_data	Character; the type of missing data assumed. Options are NA, "mar", or "mcar". Default is NA.
chains	Integer; the number of MCMC chains to run. Default is 1.
multicore	Integer; the number of cores to use for parallel execution. Default is 1.
seed	Integer; the seed number for MCMC fitting. Default is NA.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the gamma proposal density. Default is 0.2.

jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_mean_gamma	Numeric; mean of log normal prior for gamma. Default is 0.5.
pr_sd_gamma	Numeric; standard deviation of log normal prior for gamma. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_mean_alpha	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
pr_sd_alpha	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
fix_alpha_1	Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.

adapt

List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, accept_alpha, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.

- `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is `0.44`.
- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/alpha/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

`...` Additional arguments passed to the underlying model-fitting functions.

Details

Additional arguments and return values for each function are documented in the respective function's description.

* For 2PL LSIRM with data included missing value are detailed in lsirm2pl_mar and lsirm2pl_mcar.

* For 2PL LSIRM using the spike-and-slab model selection approach are detailed in lsirm2pl_ss.

* For continuous version of 2PL LSIRM are detailed in lsirm2pl_normal_o.

For 2PL LSIRM with binary item response data, the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$logit(P(Y_{j,i} = 1|\theta_j, \alpha_i, \beta_i, \gamma, z_j, w_i)) = \theta_j * \alpha_i + \beta_i - \gamma||z_j - w_i||$$

For 2PL LSIRM with continuous item response data, the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$Y_{j,i} = \alpha_i\theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$

Value

`lsirm2pl` returns an object of list. The basic return list containing the following components:

`data`	A data frame or matrix containing the variables used in the model.
`bic`	A numeric value representing the Bayesian Information Criterion (BIC).
`mcmc_inf`	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
`map_inf`	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
`beta_estimate`	Posterior estimates of the beta parameter.

theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
alpha_estimate	posterior estimates of alpha parameter..
gamma_estimate	Posterior estimates of gamma parameter.
z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.
theta_sd	Posterior samples of the standard deviation of theta.
alpha	Posterior samples of the alpha parameter.
gamma	Posterior samples of the gamma parameter.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter before Procrustes matching.
w_raw	Posterior samples of the w parameter before Procrustes matching.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_alpha	Acceptance ratio for the alpha parameter.
accept_gamma	Acceptance ratio for the gamma parameter.
...	Additional return values for various settings. Refer to the functions in the Details.

Note

If both spikenslab and fixed_gamma are set TRUE, it returns error because both are related to gamma.

References

Jeon, M., Jin, I. H., Schweinberger, M., & Baugh, S. (2021). Mapping unobserved item-respondent interactions: A latent space item response model with interaction map. *Psychometrika*, 86(2), 378–403. https://link.springer.com/article/10.1007/s11336-021-09762-5

Roberts, G. O., Gelman, A., & Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1), 110–120. doi:10.1214/aoap/1034625254

Roberts, G. O., & Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4), 351–367. doi:10.1214/ss/1015346320

Andrieu, C., & Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4), 343–373. doi:10.1007/s112220089110y

See Also

The 2PL LSIRM for binary item response data as following:

lsirm2pl_o, lsirm2pl_fixed_gamma, lsirm2pl_mar,lsirm2pl_mcar, lsirm2pl_fixed_gamma_mar, lsirm2pl_fixed_gamma_mcar, lsirm2pl_ss, lsirm2pl_mar_ss, and lsirm2pl_mcar_ss

The 2PL LSIRM for continuous item response data as following:

lsirm2pl_normal_o, lsirm2pl_normal_fixed_gamma, lsirm2pl_normal_mar, lsirm2pl_normal_mcar,lsirm1pl_norm lsirm2pl_normal_fixed_gamma_mcar, lsirm2pl_normal_ss, lsirm2pl_normal_mar_ss, lsirm2pl_normal_mcar_ss

Examples

```
# generate example item response matrix
data <- matrix(rbinom(500, size = 1, prob = 0.5), ncol = 10, nrow = 50)
lsirm_result <- lsirm2pl(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl())

# Fit with adaptive MCMC
lsirm_result <- lsirm2pl(data, adapt = list(use_adapt = TRUE))
```

---

lsirm2pl_fixed_gamma *2PL LSIRM fixing gamma to 1.*

Description

lsirm2pl_fixed_gamma is used to fit 2PL LSIRM fixing gamma to 1. lsirm2pl_fixed_gamma factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_fixed_gamma(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_z = 0.5,
```

```
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_mean_alpha	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
pr_sd_alpha	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

verbose
: Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.

fix_theta_sd
: Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

fix_alpha_1
: Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.

adapt
: List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, accept_alpha, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^α. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^α. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/alpha/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm2pl_fixed_gamma models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$logit(P(Y_{j,i} = 1|\theta_j, \alpha_i, \beta_i, z_j, w_i)) = \theta_j * \alpha_i + \beta_i - ||z_j - w_i||$$

Value

lsirm2pl_fixed_gamma returns an object of list containing the following components:

lsirm1pl_fixed_gamma returns an object of list containing the following components:

- data — Data frame or matrix containing the variables in the model.
- bic — Numeric value with the corresponding BIC.
- mcmc_inf — Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
- map_inf — The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
- beta_estimate — Posterior estimates of the beta parameter.
- theta_estimate — Posterior estimates of the theta parameter.
- sigma_theta_estimate — Posterior estimates of the standard deviation of theta.
- z_estimate — Posterior estimates of the z parameter.
- w_estimate — Posterior estimates of the w parameter.
- beta — Posterior samples of the beta parameter.
- theta — Posterior samples of the theta parameter.
- theta_sd — Posterior samples of the standard deviation of theta.
- z — Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
- w — Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
- z_raw — Posterior samples of the z parameter without procrustes matching.
- w_raw — Posterior samples of the w parameter without procrustes matching.
- accept_beta — Acceptance ratio for the beta parameter.
- accept_theta — Acceptance ratio for the theta parameter.
- accept_z — Acceptance ratio for the z parameter.
- accept_w — Acceptance ratio for the w parameter.
- alpha_estimate — Posterior estimates of the alpha parameter.
- alpha — Posterior estimates of the alpha parameter.
- accept_alpha — Acceptance ratio for the alpha parameter.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

lsirm_result <- lsirm2pl_fixed_gamma(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = FALSE, fixed_gamma = TRUE))
```

`lsirm2pl_fixed_gamma_mar`

2PL LSIRM fixing gamma to 1 for missing at random data.

Description

lsirm2pl_fixed_gamma_mar is used to fit 2PL LSIRM fixing gamma to 1 in incomplete data assumed to be missing at random. lsirm2pl_fixed_gamma_mar factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while considering the missing element under the assumption of missing at random. Unlike 1pl model, 2pl model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_fixed_gamma_mar(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  missing.val = NA,
```

```
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_mean_alpha	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
pr_sd_alpha	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
missing.val	Numeric; A number to replace missing values. Default is 99.
verbose	Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
fix_alpha_1	Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.
adapt	List; optional adaptive MCMC control. If not NULL, proposal SDs are adapted during burn-in using a Robbins-Monro update on the log proposal SD and held fixed afterward. See lsirm2pl for the full list of adaptive tuning settings and their default targets.

Details

`lsirm2pl_fixed_gamma_mar` models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$logit(P(Y_{j,i} = 1|\theta_j, \alpha_i, \beta_i, z_j, w_i)) = \theta_j * \alpha_i + \beta_i - ||z_j - w_i||$$

Under the assumption of missing at random, the model takes the missing element into consideration in the sampling procedure. For the details of missing at random assumption and data augmentation, see References.

Value

`lsirm2pl_fixed_gamma_mar` returns an object of list containing the following components:

`data` Data frame or matrix containing the variables in the model.

`missing.val` A number to replace missing values.

`bic` Numeric value with the corresponding BIC.

`mcmc_inf` Details about the number of MCMC iterations, burn-in periods, and thinning intervals.

`map_inf` The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.

`beta_estimate` Posterior estimates of the beta parameter.

`theta_estimate` Posterior estimates of the theta parameter.

`sigma_theta_estimate` Posterior estimates of the standard deviation of theta.

`z_estimate` Posterior estimates of the z parameter.

`w_estimate` Posterior estimates of the w parameter.

`imp_estimate` Probability of imputing a missing value with 1.

`beta` Posterior samples of the beta parameter.

`theta` Posterior samples of the theta parameter.

`theta_sd` Posterior samples of the standard deviation of theta.

`z` Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.

`w` Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.

`z_raw` Posterior samples of the z parameter without procrustes matching.

`w_raw` Posterior samples of the w parameter without procrustes matching.

`imp` Imputation for missing Values using posterior samples.

`accept_beta` Acceptance ratio for the beta parameter.

`accept_theta` Acceptance ratio for the theta parameter.

`accept_z` Acceptance ratio for the z parameter.

accept_w	Acceptance ratio for the w parameter.
alpha_estimate	Posterior estimates of the alpha parameter.
alpha	Posterior estimates of the alpha parameter.
accept_alpha	Acceptance ratio for the alpha parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm2pl_fixed_gamma_mar(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = FALSE, fixed_gamma = TRUE,
                      missing_data = "mar"))
```

---

lsirm2pl_fixed_gamma_mcar

2PL LSIRM fixing gamma to 1 for missing completely at random data.

Description

lsirm2pl_fixed_gamma_mcar is used to fit 2PL LSIRM fixing gamma to 1 in incomplete data assumed to be missing completely at random. lsirm2pl_fixed_gamma_mcar factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while ignoring the missing element under the assumption of missing completely at random. Unlike 1pl model, 2pl model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_fixed_gamma_mcar(
  data,
  ndim = 2,
  niter = 15000,
```

```
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.

pr_mean_alpha	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
pr_sd_alpha	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
missing.val	Numeric; A number to replace missing values. Default is 99.
verbose	Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
fix_alpha_1	Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.
adapt	List; optional adaptive MCMC control. If not NULL, proposal SDs are adapted during burn-in using a Robbins-Monro update on the log proposal SD and held fixed afterward. See lsirm2pl for the full list of adaptive tuning settings and their default targets.

Details

lsirm2pl_fixed_gamma_mcar models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$logit(P(Y_{j,i} = 1|\theta_j, \alpha_i, \beta_i, z_j, w_i)) = \theta_j * \alpha_i + \beta_i - ||z_j - w_i||$$

Under the assumption of missing completely at random, the model ignores the missing element in doing inference. For the details of missing completely at random assumption and data augmentation, see References.

Value

lsirm2pl_fixed_gamma_mar returns an object of list containing the following components:

data	Data frame or matrix containing the variables in the model.
missing.val	A number to replace missing values.
bic	Numeric value with the corresponding BIC.
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.

`z_estimate`	Posterior estimates of the z parameter.
`w_estimate`	Posterior estimates of the w parameter.
`beta`	Posterior samples of the beta parameter.
`theta`	Posterior samples of the theta parameter.
`theta_sd`	Posterior samples of the standard deviation of theta.
`z`	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`w`	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
`z_raw`	Posterior samples of the z parameter without procrustes matching.
`w_raw`	Posterior samples of the w parameter without procrustes matching.
`accept_beta`	Acceptance ratio for the beta parameter.
`accept_theta`	Acceptance ratio for the theta parameter.
`accept_z`	Acceptance ratio for the z parameter.
`accept_w`	Acceptance ratio for the w parameter.
`alpha_estimate`	Posterior estimates of the alpha parameter.
`alpha`	Posterior estimates of the alpha parameter.
`accept_alpha`	Acceptance ratio for the alpha parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm2pl_fixed_gamma_mcar(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = FALSE, fixed_gamma = TRUE,
                      missing_data = "mcar"))
```

lsirm2pl_mar — *2PL LSIRM for missing at random data.*

Description

lsirm2pl_mar is used to fit 2PL LSIRM in incomplete data assumed to be missing at random. lsirm2pl_mar factorizes item response matrix into column-wise item effect, row-wise respondent effect in a latent space, while considering the missing element under the assumption of missing at random. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_mar(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

`data` Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.

ndim — Integer; the dimension of the latent space. Default is 2.

niter — Integer; the total number of MCMC iterations to run. Default is 15000.

nburn — Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.

nthin — Integer; the number of MCMC iterations to thin. Default is 5.

nprint — Integer; the interval at which MCMC samples are displayed during execution. Default is 500.

jump_beta — Numeric; the jumping rule for the beta proposal density. Default is 0.4.

jump_theta — Numeric; the jumping rule for the theta proposal density. Default is 1.

jump_alpha — Numeric; the jumping rule for the alpha proposal density. Default is 1.

jump_gamma — Numeric; the jumping rule for the gamma proposal density. Default is 0.2.

jump_z — Numeric; the jumping rule for the z proposal density. Default is 0.5.

jump_w — Numeric; the jumping rule for the w proposal density. Default is 0.5.

pr_mean_beta — Numeric; the mean of the normal prior for beta. Default is 0.

pr_sd_beta — Numeric; the standard deviation of the normal prior for beta. Default is 1.

pr_mean_theta — Numeric; the mean of the normal prior for theta. Default is 0.

pr_sd_theta — Numeric; the standard deviation of the normal prior for theta. Default is 1.

pr_mean_gamma — Numeric; mean of log normal prior for gamma. Default is 0.5.

pr_sd_gamma — Numeric; standard deviation of log normal prior for gamma. Default is 1.

pr_mean_alpha — Numeric; the mean of the log normal prior for alpha. Default is 0.5.

pr_sd_alpha — Numeric; the standard deviation of the log normal prior for alpha. Default is 1.

pr_a_theta — Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_b_theta — Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

missing.val — Numeric; A number to replace missing values. Default is 99.

verbose — Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.

fix_theta_sd — Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

fix_alpha_1 — Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.

adapt — List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, `accept_alpha`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.
- `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is `0.44`.
- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/alpha/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

Details

`lsirm2pl_mar` models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j in the shared metric space, with γ represents the weight of the distance term. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$logit(P(Y_{j,i} = 1|\theta_j, \alpha_i, \beta_i, \gamma, z_j, w_i)) = \theta_j * \alpha_i + \beta_i - \gamma||z_j - w_i||$$

Under the assumption of missing at random, the model takes the missing element into consideration in the sampling procedure. For the details of missing at random assumption and data augmentation, see References.

Value

`lsirm2pl_mar` returns an object of list containing the following components:

`data` Data frame or matrix containing the variables in the model.

`missing.val` A number to replace missing values.

bic	Numeric value with the corresponding BIC.
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
gamma_estimate	posterior estimates of gamma parameter.
z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
imp_estimate	Probability of imputing a missing value with 1.
beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.
gamma	Posterior samples of the gamma parameter.
theta_sd	Posterior samples of the standard deviation of theta.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
imp	Imputation for missing Values using posterior samples.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.
alpha_estimate	Posterior estimates of the alpha parameter.
alpha	Posterior estimates of the alpha parameter.
accept_alpha	Acceptance ratio for the alpha parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5), ncol=10, nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2), ncol=10, nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm2pl_mar(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = FALSE, fixed_gamma = FALSE,
                      missing_data = "mar"))
```

---

lsirm2pl_mar_ss — *2PL LSIRM with model selection approach for missing at random data.*

Description

lsirm2pl_mar_ss is used to fit 2PL LSIRM based on spike-and-slab priors in incomplete data assumed to be missing at random. lsirm2pl_mar_ss factorizes item response matrix into column-wise item effect, row-wise respondent effect in a latent space, while considering the missing element under the assumption of missing at random. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_mar_ss(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
```

```
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_xi_a = 1,
  pr_xi_b = 1,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_spike_mean	Numeric; the mean of spike prior for log gamma. Default is -3.
pr_spike_sd	Numeric; the standard deviation of spike prior for log gamma. Default is 1.

pr_slab_mean — Numeric; the mean of spike prior for log gamma. Default is 0.5.

pr_slab_sd — Numeric; the standard deviation of spike prior for log gamma. Default is is 1.

pr_mean_alpha — Numeric; the mean of the log normal prior for alpha. Default is 0.5.

pr_sd_alpha — Numeric; the standard deviation of the log normal prior for alpha. Default is 1.

pr_a_theta — Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_b_theta — Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_xi_a — Numeric; the first shape parameter of beta prior for latent variable xi. Default is 1.

pr_xi_b — Numeric; the second shape parameter of beta prior for latent variable xi. Default is 1.

missing.val — Numeric; a number to replace missing values. Default is 99.

verbose — Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.

fix_theta_sd — Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

fix_alpha_1 — Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.

adapt — List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, `accept_alpha`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^α. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^α. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.

- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/alpha/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm2pl_mar_ss models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j in the shared metric space, with γ represents the weight of the distance term. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$logit(P(Y_{j,i} = 1|\theta_j, \alpha_i, \beta_i, \gamma, z_j, w_i)) = \theta_j * \alpha_i + \beta_i - \gamma||z_j - w_i||$$

Under the assumption of missing at random, the model takes the missing element into consideration in the sampling procedure. For the details of missing at random assumption and data augmentation, see References. lsirm2pl_mar_ss model include model selection approach based on spike-and-slab priors for log gamma. For detail of spike-and-slab priors, see References.

Value

lsirm2pl_mar_ss returns an object of list containing the following components:

data	Data frame or matrix containing the variables in the model.
missing.val	A number to replace missing values.
bic	Numeric value with the corresponding BIC.
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
gamma_estimate	posterior estimates of gamma parameter.
z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
beta	Posterior samples of the beta parameter.

theta	Posterior samples of the theta parameter.
gamma	Posterior samples of the gamma parameter.
theta_sd	Posterior samples of the standard deviation of theta.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
pi	Posterior samples of phi which is indicator of spike and slab prior. If phi is 1, log gamma follows the slab prior, otherwise follows the spike prior.
imp	Imputation for missing Values using posterior samples.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.
pi_estimate	Posterior estimation of phi. inclusion probability of gamma. if estimation of phi is less than 0.5, choose Rasch model with gamma = 0, otherwise latent space model with gamma > 0.
imp_estimate	Probability of imputing a missing value with 1.
alpha_estimate	Posterior estimates of the alpha parameter.
alpha	Posterior estimates of the alpha parameter.
accept_alpha	Acceptance ratio for the alpha parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons. Ishwaran, H., & Rao, J. S. (2005). Spike and slab variable selection: Frequentist and Bayesian strategies. The Annals of Statistics, 33(2), 730-773.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5), ncol=10, nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2), ncol=10, nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm2pl_mar_ss(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = TRUE, fixed_gamma = FALSE,
                      missing_data = "mar"))
```

`lsirm2pl_mcar` *2PL LSIRM for missing completely at random data.*

Description

lsirm2pl_mcar is used to fit 2PL LSIRM in incomplete data assumed to be missing completely at random. lsirm2pl_mcar factorizes item response matrix into column-wise item effect, row-wise respondent effect in a latent space, while ignoring the missing element under the assumption of missing completely at random. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_mcar(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the gamma proposal density. Default is 0.2
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_mean_gamma	Numeric; mean of log normal prior for gamma. Default is 0.5.
pr_sd_gamma	Numeric; standard deviation of log normal prior for gamma. Default is 1.
pr_mean_alpha	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
pr_sd_alpha	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
missing.val	Numeric; A number to replace missing values. Default is 99.
verbose	Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
fix_alpha_1	Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.
adapt	List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta,

accept_theta, accept_alpha, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/alpha/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm2pl_mcar models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j in the shared metric space, with γ represents the weight of the distance term. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$logit(P(Y_{j,i} = 1|\theta_j, \alpha_i, \beta_i, \gamma, z_j, w_i)) = \theta_j * \alpha_i + \beta_i - \gamma||z_j - w_i||$$

Under the assumption of missing completely at random, the model ignores the missing element in doing inference. For the details of missing completely at random assumption and data augmentation, see References.

Value

lsirm2pl_mar returns an object of list containing the following components:

data — A data frame or matrix containing the variables used in the model.

missing.val A number to replace missing values.

bic Numeric value with the corresponding BIC.

mcmc_inf Details about the number of MCMC iterations, burn-in periods, and thinning intervals.

map_inf The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.

beta_estimate Posterior estimates of the beta parameter.

theta_estimate Posterior estimates of the theta parameter.

sigma_theta_estimate Posterior estimates of the standard deviation of theta.

gamma_estimate Posterior estimates of gamma parameter.

z_estimate Posterior estimates of the z parameter.

w_estimate Posterior estimates of the w parameter.

beta Posterior samples of the beta parameter.

theta Posterior samples of the theta parameter.

theta_sd Posterior samples of the standard deviation of theta.

gamma Posterior samples of the gamma parameter.

z Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.

w Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.

z_raw Posterior samples of the z parameter without procrustes matching.

w_raw Posterior samples of the w parameter without procrustes matching.

accept_beta Acceptance ratio for the beta parameter.

accept_theta Acceptance ratio for the theta parameter.

accept_z Acceptance ratio for the z parameter.

accept_w Acceptance ratio for the w parameter.

accept_gamma Acceptance ratio for the gamma parameter.

alpha_estimate Posterior estimates of the alpha parameter.

alpha Posterior estimates of the alpha parameter.

accept_alpha Acceptance ratio for the alpha parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5), ncol=10, nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2), ncol=10, nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm2pl_mcar(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = FALSE, fixed_gamma = FALSE,
                      missing_data = "mcar"))
```

`lsirm2pl_mcar_ss` *2PL LSIRM with model selection approach for missing completely at random data.*

Description

lsirm2pl_mar_ss is used to fit 2PL LSIRM based on spike-and-slab priors in incomplete data assumed to be missing completely at random. lsirm2pl_mar_ss factorizes item response matrix into column-wise item effect, row-wise respondent effect in a latent space, while ignoring the missing element under the assumption of missing completely at random. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_mcar_ss(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
```

```
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_xi_a = 1,
  pr_xi_b = 1,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_spike_mean	Numeric; the mean of spike prior for log gamma. Default is -3.
pr_spike_sd	Numeric; the standard deviation of spike prior for log gamma. Default is 1.

`pr_slab_mean`	Numeric; the mean of spike prior for log gamma. Default is 0.5.
`pr_slab_sd`	Numeric; the standard deviation of spike prior for log gamma. Default is 1.
`pr_mean_alpha`	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
`pr_sd_alpha`	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
`pr_a_theta`	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
`pr_b_theta`	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
`pr_xi_a`	Numeric; the first shape parameter of beta prior for latent variable xi. Default is 1.
`pr_xi_b`	Numeric; the second shape parameter of beta prior for latent variable xi. Default is 1.
`missing.val`	Numeric; a number to replace missing values. Default is 99.
`verbose`	Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.
`fix_theta_sd`	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
`fix_alpha_1`	Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.

`adapt`
: List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, `accept_alpha`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.

- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/alpha/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm2pl_mcar_ss models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j in the shared metric space, with γ represents the weight of the distance term. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$logit(P(Y_{j,i} = 1|\theta_j, \alpha_i, \beta_i, \gamma, z_j, w_i)) = \theta_j * \alpha_i + \beta_i - \gamma||z_j - w_i||$$

Under the assumption of missing at random, the model takes the missing element into consideration in the sampling procedure. For the details of missing at random assumption and data augmentation, see References. lsirm2pl_mcar_ss model include model selection approach based on spike-and-slab priors for log gamma. For detail of spike-and-slab priors, see References.

Value

lsirm2pl_mar_ss returns an object of list containing the following components:

data
: Data frame or matrix containing the variables in the model.

missing.val
: A number to replace missing values.

bic
: Numeric value with the corresponding BIC.

mcmc_inf
: Details about the number of MCMC iterations, burn-in periods, and thinning intervals.

map_inf
: The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.

beta_estimate
: Posterior estimates of the beta parameter.

theta_estimate
: Posterior estimates of the theta parameter.

sigma_theta_estimate
: Posterior estimates of the standard deviation of theta.

gamma_estimate
: posterior estimates of gamma parameter.

z_estimate
: Posterior estimates of the z parameter.

w_estimate
: Posterior estimates of the w parameter.

beta
: Posterior samples of the beta parameter.

theta
: Posterior samples of the theta parameter.

gamma
: Posterior samples of the gamma parameter.

theta_sd	Posterior samples of the standard deviation of theta.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
pi	Posterior samples of phi which is indicator of spike and slab prior. If phi is 1, log gamma follows the slab prior, otherwise follows the spike prior.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.
alpha_estimate	Posterior estimates of the alpha parameter.
alpha	Posterior estimates of the alpha parameter.
accept_alpha	Acceptance ratio for the alpha parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons. Ishwaran, H., & Rao, J. S. (2005). Spike and slab variable selection: Frequentist and Bayesian strategies. The Annals of Statistics, 33(2), 730-773.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5), ncol=10, nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2), ncol=10, nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm2pl_mcar_ss(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = TRUE, fixed_gamma = FALSE,
                      missing_data = "mcar"))
```

---

lsirm2pl_normal_fixed_gamma
2PL LSIRM fixing gamma to 1 with normal likelihood

Description

lsirm2pl_normal_fixed_gamma is used to fit 2PL LSIRM with gamma fixed to 1 for continuous variable. lsirm2pl_normal_fixed_gamma factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_normal_fixed_gamma(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding

item.

ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_mean_alpha	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
pr_sd_alpha	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_a_eps	Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
pr_b_eps	Numeric; the scale parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
verbose	Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
fix_alpha_1	Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.
adapt	List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, accept_alpha, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The

defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.
- `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is `0.44`.
- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/alpha/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

Details

`lsirm2pl_normal_fixed_gamma` models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$Y_{j,i} = \alpha_i\theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$

Value

`lsirm2pl_normal_fixed_gamma` returns an object of list containing the following components:

`data`	A data frame or matrix containing the variables used in the model.
`bic`	A numeric value representing the Bayesian Information Criterion (BIC).
`mcmc_inf`	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.

map_inf — The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.

beta_estimate — Posterior estimates of the beta parameter.

theta_estimate — Posterior estimates of the theta parameter.

sigma_theta_estimate — Posterior estimates of the standard deviation of theta.

z_estimate — Posterior estimates of the z parameter.

w_estimate — Posterior estimates of the w parameter.

beta — Posterior samples of the beta parameter.

theta — Posterior samples of the theta parameter.

theta_sd — Posterior samples of the standard deviation of theta.

z — Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.

w — Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.

z_raw — Posterior samples of the z parameter without procrustes matching.

w_raw — Posterior samples of the w parameter without procrustes matching.

accept_beta — Acceptance ratio for the beta parameter.

accept_theta — Acceptance ratio for the theta parameter.

accept_z — Acceptance ratio for the z parameter.

accept_w — Acceptance ratio for the w parameter.

sigma_estimate — Posterior estimates of the standard deviation.

sigma — Posterior samples of the standard deviation.

alpha_estimate — Posterior estimates of the alpha parameter.

alpha — Posterior estimates of the alpha parameter.

accept_alpha — Acceptance ratio for the alpha parameter.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

lsrm_result <- lsirm2pl_normal_fixed_gamma(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = FALSE, fixed_gamma = TRUE,
                      niter = 1000, nburn = 500))
```

`lsirm2pl_normal_fixed_gamma_mar` *2PL LSIRM fixing gamma to 1 with normal likelihood for missing at random data.*

Description

lsirm2pl_normal_fixed_gamma_mar is used to fit 2PL LSIRM with gamma fixed to 1 for continuous variable in incomplete data assumed to be missing at random.

lsirm2pl_normal_fixed_gamma_mar factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while considering the missing element under the assumption of missing at random. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_normal_fixed_gamma_mar(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_mean_alpha	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
pr_sd_alpha	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_a_eps	Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
pr_b_eps	Numeric; the scale parameter of inverse gamma prior for variance of data likelihood Default is 0.001.
missing.val	Numeric; a number to replace missing values. Default is 99.
verbose	Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
fix_alpha_1	Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.

`adapt` List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, `accept_alpha`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5–2.0`.
- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6–0.8`.
- `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is `0.44`.
- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/alpha/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

Details

`lsirm2pl_normal_fixed_gamma_mar` models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$Y_{j,i} = \alpha_i\theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$ Under the assumption of missing at random, the model takes the missing element into consideration in the sampling procedure. For the details of missing at random assumption and data augmentation, see References.

Value

lsirm2pl_normal_fixed_gamma_mar returns an object of list containing the following components:

data	Data frame or matrix containing the variables in the model.
missing.val	A number to replace missing values.
bic	Numeric value with the corresponding BIC.
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
imp_estimate	Probability of imputing a missing value with 1.
beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.
theta_sd	Posterior samples of the standard deviation of theta.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
imp	Imputation for missing Values using posterior samples.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
sigma_estimate	Posterior estimates of the standard deviation.
sigma	Posterior samples of the standard deviation.
alpha_estimate	Posterior estimates of the alpha parameter.
alpha	Posterior estimates of the alpha parameter.
accept_alpha	Acceptance ratio for the alpha parameter.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm2pl_normal_fixed_gamma_mar(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = FALSE, fixed_gamma = TRUE,
                      niter = 1000, nburn = 500,
                      missing_data = "mar"))
```

`lsirm2pl_normal_fixed_gamma_mcar` *2PL LSIRM fixing gamma to 1 with normal likelihood for missing completely at random data.*

Description

lsirm2pl_normal_fixed_gamma_mcar is used to fit 2PL LSIRM with gamma fixed to 1 for continuous variable in incomplete data assumed to be missing completely at random.

lsirm2pl_normal_fixed_gamma_mcar factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while ignoring the missing element under the assumption of missing completely at random. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_normal_fixed_gamma_mcar(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_z = 0.5,
```

```
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_mean_alpha	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
pr_sd_alpha	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_b_theta — Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_a_eps — Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

pr_b_eps — Numeric; the scale parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

missing.val — Numeric; a number to replace missing values. Default is 99.

verbose — Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.

fix_theta_sd — Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

fix_alpha_1 — Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.

adapt — List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, accept_alpha, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.

- target_accept_beta/theta/alpha/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm2pl_normal_fixed_gamma_mcar models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$Y_{j,i} = \alpha_i\theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$ Under the assumption of missing completely at random, the model ignores the missing element in doing inference. For the details of missing completely at random assumption and data augmentation, see References.

Value

lsirm2pl_normal_fixed_gamma_mcar returns an object of list containing the following components:

data	Data frame or matrix containing the variables in the model.
missing.val	A number to replace missing values.
bic	Numeric value with the corresponding BIC.
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.
theta_sd	Posterior samples of the standard deviation of theta.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
accept_beta	Acceptance ratio for the beta parameter.

accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
sigma_estimate	Posterior estimates of the standard deviation.
sigma	Posterior samples of the standard deviation.
alpha_estimate	Posterior estimates of the alpha parameter.
alpha	Posterior estimates of the alpha parameter.
accept_alpha	Acceptance ratio for the alpha parameter.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm2pl_normal_fixed_gamma_mcar(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = FALSE, fixed_gamma = TRUE,
                      niter = 1000, nburn = 500,
                      missing_data = "mcar"))
```

---

lsirm2pl_normal_mar *2PL LSIRM with normal likelihood and missing at random data.*

Description

lsirm2pl_normal_mar is used to fit 2PL LSIRM for continuous variable in incomplete data assumed to be missing at random. lsirm2pl_normal_mar factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while considering the missing element under the assumption of missing at random. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_normal_mar(
  data,
  ndim = 2,
```

```
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the gamma proposal density. Default is 0.2
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.

jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_mean_gamma	Numeric; mean of log normal prior for gamma. Default is 0.5.
pr_sd_gamma	Numeric; standard deviation of log normal prior for gamma. Default is 1.
pr_mean_alpha	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
pr_sd_alpha	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_a_eps	Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
pr_b_eps	Numeric; the scale parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
missing.val	Numeric; a number to replace missing values. Default is 99.
verbose	Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
fix_alpha_1	Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.

adapt — List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, accept_alpha, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.

- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/alpha/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm2pl_normal_mar models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$Y_{j,i} = \alpha_i\theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$ Under the assumption of missing at random, the model takes the missing element into consideration in the sampling procedure. For the details of missing at random assumption and data augmentation, see References.

Value

lsirm2pl_normal_mar returns an object of list containing the following components:

data	Data frame or matrix containing the variables in the model.
missing.val	A number to replace missing values.
bic	Numeric value with the corresponding BIC.
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
gamma_estimate	posterior estimates of gamma parameter.

z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
imp_estimate	Posterior mean imputation for each missing cell.
beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.
gamma	Posterior samples of the gamma parameter.
theta_sd	Posterior samples of the standard deviation of theta.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
imp	Imputation for missing Values using posterior samples.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.
sigma_estimate	Posterior estimates of the standard deviation.
sigma	Posterior samples of the standard deviation.
alpha_estimate	Posterior estimates of the alpha parameter.
alpha	Posterior samples of the alpha parameter.
accept_alpha	Acceptance ratio for the alpha parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm2pl_normal_mar(data, niter = 1000, nburn = 500)
```

```
# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = FALSE, fixed_gamma = FALSE,
                      niter = 1000, nburn = 500,
                      missing_data = "mar"))
```

---

lsirm2pl_normal_mar_ss   *2pl LSIRM with normal likelihood and model selection approach for missing at random data.*

Description

lsirm2pl_normal_mar_ss is used to fit 2pl LSIRM with model selection approach based on spike-and-slab priors for continuous variable in incomplete data assumed to be missing at random. lsirm2pl_normal_mar_ss factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while considering the missing element under the assumption of missing at random. Unlike 1pl model, 2pl model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_normal_mar_ss(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_eps = 0.001,
```

```
    pr_b_eps = 0.001,
    pr_a_theta = 0.001,
    pr_b_theta = 0.001,
    pr_xi_a = 0.001,
    pr_xi_b = 0.001,
    missing.val = NA,
    verbose = FALSE,
    fix_theta_sd = FALSE,
    fix_alpha_1 = TRUE,
    adapt = NULL
  )
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_spike_mean	Numeric; the mean of spike prior for log gamma. Default is -3.
pr_spike_sd	Numeric; the standard deviation of spike prior for log gamma. Default is 1.
pr_slab_mean	Numeric; the mean of spike prior for log gamma. Default is 0.5.
pr_slab_sd	Numeric; the standard deviation of spike prior for log gamma. Default is is 1.
pr_mean_alpha	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
pr_sd_alpha	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
pr_a_eps	Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

pr_b_eps — Numeric; the scale parameter of inverse gamma prior for variance of data likelihood Default is 0.001.

pr_a_theta — Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_b_theta — Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_xi_a — Numeric; the first shape parameter of beta prior for latent variable xi. Default is 1.

pr_xi_b — Numeric; the second shape parameter of beta prior for latent variable xi. Default is 1.

missing.val — Numeric; a number to replace missing values. Default is 99.

verbose — Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.

fix_theta_sd — Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

fix_alpha_1 — Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.

adapt — List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, `accept_alpha`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.
- `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.

- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is `0.44`.
- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/alpha/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

Details

`lsirm2pl_normal_mar_ss` models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$Y_{j,i} = \alpha_i\theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$ Under the assumption of missing at random, the model takes the missing element into consideration in the sampling procedure. For the details of missing at random assumption and data augmentation, see References. `lsirm2pl_normal_mcar_ss` model include model selection approach based on spike-and-slab priors for log gamma. For detail of spike-and-slab priors, see References.

Value

`lsirm2pl_normal_mar_ss` returns an object of list containing the following components:

`data`	Data frame or matrix containing the variables in the model.
`missing.val`	A number to replace missing values.
`bic`	Numeric value with the corresponding BIC.
`mcmc_inf`	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
`map_inf`	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
`beta_estimate`	Posterior estimates of the beta parameter.
`theta_estimate`	Posterior estimates of the theta parameter.
`sigma_theta_estimate`	Posterior estimates of the standard deviation of theta.
`gamma_estimate`	posterior estimates of gamma parameter.
`z_estimate`	Posterior estimates of the z parameter.
`w_estimate`	Posterior estimates of the w parameter.
`z_raw`	Posterior samples of the z parameter without procrustes matching.
`w_raw`	Posterior samples of the w parameter without procrustes matching.
`beta`	Posterior samples of the beta parameter.
`theta`	Posterior samples of the theta parameter.
`gamma`	Posterior samples of the gamma parameter.

theta_sd	Posterior samples of the standard deviation of theta.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
pi	Posterior samples of phi which is indicator of spike and slab prior. If phi is 1, log gamma follows the slab prior, otherwise follows the spike prior.
imp	Imputation for missing Values using posterior samples.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.
pi_estimate	Posterior estimation of phi. inclusion probability of gamma. if estimation of phi is less than 0.5, choose Rasch model with gamma = 0, otherwise latent space model with gamma > 0.
imp_estimate	Probability of imputing a missing value with 1.
sigma_estimate	Posterior estimates of the standard deviation.
sigma	Posterior samples of the standard deviation.
alpha_estimate	Posterior estimates of the alpha parameter.
alpha	Posterior estimates of the alpha parameter.
accept_alpha	Acceptance ratio for the alpha parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons. Ishwaran, H., & Rao, J. S. (2005). Spike and slab variable selection: Frequentist and Bayesian strategies. The Annals of Statistics, 33(2), 730-773.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm2pl_normal_mar_ss(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = TRUE, fixed_gamma = FALSE,
                      niter = 1000, nburn = 500,
                      missing_data = "mar"))
```

lsirm2pl_normal_mcar — *2PL LSIRM with normal likelihood and missing completely at random data.*

Description

lsirm2pl_normal_mcar is used to fit 2PL LSIRM for continuous variable in incomplete data assumed to be missing completely at random. lsirm2pl_normal_mcar factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while ignoring the missing element under the assumption of missing completely at random. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_normal_mcar(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the gamma proposal density. Default is 0.2
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_mean_gamma	Numeric; mean of log normal prior for gamma. Default is 0.5.
pr_sd_gamma	Numeric; standard deviation of log normal prior for gamma. Default is 1.
pr_mean_alpha	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
pr_sd_alpha	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_a_eps	Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
pr_b_eps	Numeric; the scale parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
missing.val	Numeric; a number to replace missing values. Default is 99.
verbose	Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
fix_alpha_1	Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.

adapt: List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, accept_alpha, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/alpha/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm2pl_normal_mcar models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$Y_{j,i} = \alpha_i\theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$ Under the assumption of missing completely at random, the model ignores the missing element in doing inference. For the details of missing completely at random assumption and data augmentation, see References.

Value

lsirm2pl_normal_mcar returns an object of list containing the following components:

- `data` A data frame or matrix containing the variables used in the model.
- `missing.val` A number to replace missing values.
- `bic` Numeric value with the corresponding BIC.
- `mcmc_inf` Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
- `map_inf` The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
- `beta_estimate` Posterior estimates of the beta parameter.
- `theta_estimate` Posterior estimates of the theta parameter.
- `sigma_theta_estimate` Posterior estimates of the standard deviation of theta.
- `gamma_estimate` Posterior estimates of gamma parameter.
- `z_estimate` Posterior estimates of the z parameter.
- `w_estimate` Posterior estimates of the w parameter.
- `beta` Posterior samples of the beta parameter.
- `theta` Posterior samples of the theta parameter.
- `theta_sd` Posterior samples of the standard deviation of theta.
- `gamma` Posterior samples of the gamma parameter.
- `z` Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
- `w` Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
- `z_raw` Posterior samples of the z parameter without procrustes matching.
- `w_raw` Posterior samples of the w parameter without procrustes matching.
- `accept_beta` Acceptance ratio for the beta parameter.
- `accept_theta` Acceptance ratio for the theta parameter.
- `accept_z` Acceptance ratio for the z parameter.
- `accept_w` Acceptance ratio for the w parameter.
- `accept_gamma` Acceptance ratio for the gamma parameter.
- `sigma_estimate` Posterior estimates of the standard deviation.
- `sigma` Posterior samples of the standard deviation.
- `alpha_estimate` Posterior estimates of the alpha parameter.
- `alpha` Posterior estimates of the alpha parameter.
- `accept_alpha` Acceptance ratio for the alpha parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm2pl_normal_mcar(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = FALSE, fixed_gamma = FALSE,
                      niter = 1000, nburn = 500,
                      missing_data = "mcar"))
```

---

lsirm2pl_normal_mcar_ss

2PL LSIRM with normal likelihood and model selection approach for missing completely at random data.

Description

lsirm2pl_normal_mcar_ss is used to fit 2PL LSIRM with model selection approach based on spike-and-slab priors for continuous variable in incomplete data assumed to be missing completely at random. lsirm2pl_normal_mcar_ss factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space, while ignoring the missing element under the assumption of missing completely at random. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_normal_mcar_ss(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
```

```
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_xi_a = 0.001,
  pr_xi_b = 0.001,
  missing.val = NA,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.
`nburn`	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
`nthin`	Integer; the number of MCMC iterations to thin. Default is 5.
`nprint`	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
`jump_beta`	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
`jump_theta`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_alpha`	Numeric; the jumping rule for the alpha proposal density. Default is 1.
`jump_gamma`	Numeric; the jumping rule for the theta proposal density. Default is 1.
`jump_z`	Numeric; the jumping rule for the z proposal density. Default is 0.5.

jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_spike_mean	Numeric; the mean of spike prior for log gamma. Default is -3.
pr_spike_sd	Numeric; the standard deviation of spike prior for log gamma. Default is 1.
pr_slab_mean	Numeric; the mean of spike prior for log gamma. Default is 0.5.
pr_slab_sd	Numeric; the standard deviation of spike prior for log gamma. Default is is 1.
pr_mean_alpha	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
pr_sd_alpha	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
pr_a_eps	Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
pr_b_eps	Numeric; the scale parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_xi_a	Numeric; the first shape parameter of beta prior for latent variable xi. Default is 1.
pr_xi_b	Numeric; the second shape parameter of beta prior for latent variable xi. Default is 1.
missing.val	Numeric; a number to replace missing values. Default is 99.
verbose	Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
fix_alpha_1	Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.
adapt	List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, `accept_alpha`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/alpha/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm2pl_normal_mcar_ss models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$Y_{j,i} = \alpha_i\theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$ Under the assumption of missing completely at random, the model ignores the missing element in doing inference. For the details of missing completely at random assumption and data augmentation, see References. lsirm2pl_normal_mcar_ss model include model selection approach based on spike-and-slab priors for log gamma. For detail of spike-and-slab priors, see References.

Value

lsirm2pl_normal_mcar_ss returns an object of list containing the following components:

data	Data frame or matrix containing the variables in the model.
missing.val	A number to replace missing values.
bic	Numeric value with the corresponding BIC.
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.

beta_estimate Posterior estimates of the beta parameter.

theta_estimate Posterior estimates of the theta parameter.

sigma_theta_estimate Posterior estimates of the standard deviation of theta.

gamma_estimate posterior estimates of gamma parameter.

z_estimate Posterior estimates of the z parameter.

w_estimate Posterior estimates of the w parameter.

beta Posterior samples of the beta parameter.

theta Posterior samples of the theta parameter.

gamma Posterior samples of the gamma parameter.

theta_sd Posterior samples of the standard deviation of theta.

z Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.

w Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.

z_raw Posterior samples of the z parameter without procrustes matching.

w_raw Posterior samples of the w parameter without procrustes matching.

pi Posterior samples of phi which is indicator of spike and slab prior. If phi is 1, log gamma follows the slab prior, otherwise follows the spike prior.

accept_beta Acceptance ratio for the beta parameter.

accept_theta Acceptance ratio for the theta parameter.

accept_z Acceptance ratio for the z parameter.

accept_w Acceptance ratio for the w parameter.

accept_gamma Acceptance ratio for the gamma parameter.

sigma_estimate Posterior estimates of the standard deviation.

sigma Posterior samples of the standard deviation.

alpha_estimate Posterior estimates of the alpha parameter.

alpha Posterior estimates of the alpha parameter.

accept_alpha Acceptance ratio for the alpha parameter.

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons. Ishwaran, H., & Rao, J. S. (2005). Spike and slab variable selection: Frequentist and Bayesian strategies. The Annals of Statistics, 33(2), 730-773.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

# generate example missing indicator matrix
missing_mat     <- matrix(rbinom(500, size = 1, prob = 0.2),ncol=10,nrow=50)

# make missing value with missing indicator matrix
data[missing_mat==1] <- 99

lsirm_result <- lsirm2pl_normal_mcar_ss(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = TRUE, fixed_gamma = FALSE,
                      niter = 1000, nburn = 500,
                      missing_data = "mcar"))
```

---

lsirm2pl_normal_o   *2PL LSIRM with normal likelihood*

Description

lsirm2pl_normal_o is used to fit 2PL LSIRM for continuous variable. lsirm2pl_normal_o factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_normal_o(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
```

```
    pr_sd_theta = 1,
    pr_mean_gamma = 0.5,
    pr_sd_gamma = 1,
    pr_mean_alpha = 0.5,
    pr_sd_alpha = 1,
    pr_a_theta = 0.001,
    pr_b_theta = 0.001,
    pr_a_eps = 0.001,
    pr_b_eps = 0.001,
    verbose = FALSE,
    fix_theta_sd = FALSE,
    fix_alpha_1 = TRUE,
    adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the gamma proposal density. Default is 0.2
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_mean_gamma	Numeric; mean of log normal prior for gamma. Default is 0.5.
pr_sd_gamma	Numeric; standard deviation of log normal prior for gamma. Default is 1.
pr_mean_alpha	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
pr_sd_alpha	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_a_eps	Numeric; the shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
pr_b_eps	Numeric; the scale parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.
verbose	Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
fix_alpha_1	Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.

adapt

List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, `accept_alpha`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^α. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^α. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.
- `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is `0.44`.
- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/alpha/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

Details

lsirm2pl_normal_o models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$Y_{j,i} = \alpha_i\theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$

Value

lsirm2pl_normal_o returns an object of list containing the following components:

data	Data frame or matrix containing the variables used in the model.
bic	A numeric value representing the Bayesian Information Criterion (BIC).
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
gamma_estimate	Posterior estimates of gamma parameter.
z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.
theta_sd	Posterior samples of the standard deviation of theta.
gamma	Posterior samples of the gamma parameter.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.

`sigma_estimate`	Posterior estimates of the standard deviation.
`sigma`	Posterior samples of the standard deviation.
`alpha_estimate`	Posterior estimates of the alpha parameter.
`alpha`	Posterior estimates of the alpha parameter.
`accept_alpha`	Acceptance ratio for the alpha parameter.

Examples

```
# generate example (continuous) item response matrix
data <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)
lsirm_result <- lsirm2pl_normal_o(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = FALSE, fixed_gamma = FALSE,
                      niter = 1000, nburn = 500))
```

`lsirm2pl_normal_ss` *2PL LSIRM with normal likelihood and model selection approach.*

Description

lsirm2pl_normal_ss is used to fit 2PL LSIRM for continuous variable with model selection approach. lsirm2pl_normal_ss factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_normal_ss(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
```

```
  pr_sd_theta = 1,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_eps = 0.001,
  pr_b_eps = 0.001,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_xi_a = 0.001,
  pr_xi_b = 0.001,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

data	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_spike_mean	Numeric; mean of spike prior for log gamma default value is -3.
pr_spike_sd	Numeric; standard deviation of spike prior for log gamma default value is 1.

pr_slab_mean
: Numeric; mean of spike prior for log gamma default value is 0.5.

pr_slab_sd
: Numeric; standard deviation of spike prior for log gamma default value is 1.

pr_mean_alpha
: Numeric; the mean of the log normal prior for alpha. Default is 0.5.

pr_sd_alpha
: Numeric; the standard deviation of the log normal prior for alpha. Default is 1.

pr_a_eps
: Numeric; shape parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

pr_b_eps
: Numeric; scale parameter of inverse gamma prior for variance of data likelihood. Default is 0.001.

pr_a_theta
: Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_b_theta
: Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_xi_a
: Numeric; first shape parameter of beta prior for latent variable xi. Default is 1.

pr_xi_b
: Numeric; second shape parameter of beta prior for latent variable xi. Default is 1.

verbose
: Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.

fix_theta_sd
: Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

fix_alpha_1
: Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.

adapt
: List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, `accept_alpha`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

  - `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
  - `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
  - `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5–2.0`.

- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/alpha/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm2pl_normal_ss models the continuous value of response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$Y_{j,i} = \alpha_i\theta_j + \beta_i - \gamma||z_j - w_i|| + e_{j,i}$$

where the error $e_{j,i} \sim N(0, \sigma^2)$. lsrm2pl_noraml_ss model include model selection approach based on spike-and-slab priors for log gamma. For detail of spike-and-slab priors, see References.

Value

lsirm2pl_normal_ss returns an object of list containing the following components:

data	Data frame or matrix containing the variables in the model.
bic	Numeric value with the corresponding BIC.
mcmc_inf	number of mcmc iteration, burn-in periods, and thinning intervals.
map_inf	value of log maximum a posterior and iteration number which have log maximum a posterior.
beta_estimate	posterior estimation of beta.
theta_estimate	posterior estimation of theta.
sigma_theta_estimate	posterior estimation of standard deviation of theta.
sigma_estimate	posterior estimation of standard deviation.
gamma_estimate	posterior estimation of gamma.
z_estimate	posterior estimation of z.
w_estimate	posterior estimation of w.
pi_estimate	posterior estimation of phi. inclusion probability of gamma. if estimation of phi is less than 0.5, choose Rasch model with gamma = 0, otherwise latent space model with gamma > 0.

beta	posterior samples of beta.
theta	posterior samples of theta.
theta_sd	posterior samples of standard deviation of theta.
sigma	posterior samples of standard deviation.
gamma	posterior samples of gamma.
z	posterior samples of z. The output is 3-dimensional matrix with last axis represent the dimension of latent space.
w	posterior samples of w. The output is 3-dimensional matrix with last axis represent the dimension of latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
pi	posterior samples of phi which is indicator of spike and slab prior. If phi is 1, log gamma follows the slab prior, otherwise follows the spike prior.
accept_beta	accept ratio of beta.
accept_theta	accept ratio of theta.
accept_w	accept ratio of w.
accept_z	accept ratio of z.
accept_gamma	accept ratio of gamma.
alpha_estimate	Posterior estimates of the alpha parameter.
alpha	Posterior estimates of the alpha parameter.
accept_alpha	Acceptance ratio for the alpha parameter.

References

Ishwaran, H., & Rao, J. S. (2005). Spike and slab variable selection: frequentist and Bayesian strategies. The Annals of Statistics, 33(2), 730-773.

Examples

```
# generate example (continuous) item response matrix
data     <- matrix(rnorm(500, mean = 0, sd = 1),ncol=10,nrow=50)

lsirm_result <- lsirm2pl_normal_ss(data, niter = 1000, nburn = 500)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = TRUE, fixed_gamma = FALSE,
                      niter = 1000, nburn = 500))
```

lsirm2pl_o *2PL LSIRM.*

Description

lsirm2pl_o is used to fit 2PL LSIRM. lsirm2pl_o factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_o(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.

ndim	Integer; the dimension of the latent space. Default is 2.
niter	Integer; the total number of MCMC iterations to run. Default is 15000.
nburn	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
nthin	Integer; the number of MCMC iterations to thin. Default is 5.
nprint	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
jump_beta	Numeric; the jumping rule for the beta proposal density. Default is 0.4.
jump_theta	Numeric; the jumping rule for the theta proposal density. Default is 1.
jump_alpha	Numeric; the jumping rule for the alpha proposal density. Default is 1.
jump_gamma	Numeric; the jumping rule for the gamma proposal density. Default is 0.2.
jump_z	Numeric; the jumping rule for the z proposal density. Default is 0.5.
jump_w	Numeric; the jumping rule for the w proposal density. Default is 0.5.
pr_mean_beta	Numeric; the mean of the normal prior for beta. Default is 0.
pr_sd_beta	Numeric; the standard deviation of the normal prior for beta. Default is 1.
pr_mean_theta	Numeric; the mean of the normal prior for theta. Default is 0.
pr_sd_theta	Numeric; the standard deviation of the normal prior for theta. Default is 1.
pr_mean_gamma	Numeric; mean of log normal prior for gamma. Default is 0.5.
pr_sd_gamma	Numeric; standard deviation of log normal prior for gamma. Default is 1.
pr_mean_alpha	Numeric; the mean of the log normal prior for alpha. Default is 0.5.
pr_sd_alpha	Numeric; the standard deviation of the log normal prior for alpha. Default is 1.
pr_a_theta	Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
pr_b_theta	Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.
verbose	Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.
fix_alpha_1	Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.
adapt	List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, `accept_alpha`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/alpha/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm2pl_o models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$logit(P(Y_{j,i} = 1|\theta_j, \alpha_i, \beta_i, \gamma, z_j, w_i)) = \theta_j * \alpha_i + \beta_i - \gamma||z_j - w_i||$$

Value

lsirm2pl_o returns an object of list containing the following components:

data	Data frame or matrix containing the variables used in the model.
bic	A numeric value representing the Bayesian Information Criterion (BIC).
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.
sigma_theta_estimate	Posterior estimates of the standard deviation of theta.
gamma_estimate	Posterior estimates of gamma parameter.

z_estimate	Posterior estimates of the z parameter.
w_estimate	Posterior estimates of the w parameter.
beta	Posterior samples of the beta parameter.
theta	Posterior samples of the theta parameter.
theta_sd	Posterior samples of the standard deviation of theta.
gamma	Posterior samples of the gamma parameter.
z	Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
w	Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.
z_raw	Posterior samples of the z parameter without procrustes matching.
w_raw	Posterior samples of the w parameter without procrustes matching.
accept_beta	Acceptance ratio for the beta parameter.
accept_theta	Acceptance ratio for the theta parameter.
accept_z	Acceptance ratio for the z parameter.
accept_w	Acceptance ratio for the w parameter.
accept_gamma	Acceptance ratio for the gamma parameter.
alpha_estimate	Posterior estimates of the alpha parameter.
alpha	Posterior estimates of the alpha parameter.
accept_alpha	Acceptance ratio for the alpha parameter.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

lsirm_result <- lsirm2pl_o(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = FALSE, fixed_gamma = FALSE))
```

---

lsirm2pl_ss *2PL LSIRM with model selection approach.*

Description

lsirm2pl_ss is used to fit 2PL LSIRM with model selection approach based on spike-and-slab priors. lsirm2pl_ss factorizes item response matrix into column-wise item effect, row-wise respondent effect and further embeds interaction effect in a latent space. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect. The resulting latent space provides an interaction map that represents interactions between respondents and items.

Usage

```
lsirm2pl_ss(
  data,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_xi_a = 1,
  pr_xi_b = 1,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE,
  adapt = NULL
)
```

Arguments

`data`	Matrix; a binary or continuous item response matrix for analysis. Each row represents a respondent, and each column contains responses to the corresponding item.
`ndim`	Integer; the dimension of the latent space. Default is 2.
`niter`	Integer; the total number of MCMC iterations to run. Default is 15000.
`nburn`	Integer; the number of initial MCMC iterations to discard as burn-in. Default is 2500.
`nthin`	Integer; the number of MCMC iterations to thin. Default is 5.
`nprint`	Integer; the interval at which MCMC samples are displayed during execution. Default is 500.
`jump_beta`	Numeric; the jumping rule for the beta proposal density. Default is 0.4.

jump_theta Numeric; the jumping rule for the theta proposal density. Default is 1.

jump_alpha Numeric; the jumping rule for the alpha proposal density. Default is 1.

jump_gamma Numeric; the jumping rule for the theta proposal density. Default is 1.

jump_z Numeric; the jumping rule for the z proposal density. Default is 0.5.

jump_w Numeric; the jumping rule for the w proposal density. Default is 0.5.

pr_mean_beta Numeric; the mean of the normal prior for beta. Default is 0.

pr_sd_beta Numeric; the standard deviation of the normal prior for beta. Default is 1.

pr_mean_theta Numeric; the mean of the normal prior for theta. Default is 0.

pr_sd_theta Numeric; the standard deviation of the normal prior for theta. Default is 1.

pr_spike_mean Numeric; the mean of spike prior for log gamma. Default is -3.

pr_spike_sd Numeric; the standard deviation of spike prior for log gamma. Default is 1.

pr_slab_mean Numeric; the mean of spike prior for log gamma. Default is 0.5.

pr_slab_sd Numeric; the standard deviation of spike prior for log gamma. Default is is 1.

pr_mean_alpha Numeric; the mean of the log normal prior for alpha. Default is 0.5.

pr_sd_alpha Numeric; the standard deviation of the log normal prior for alpha. Default is 1.

pr_a_theta Numeric; the shape parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_b_theta Numeric; the scale parameter of the inverse gamma prior for the variance of theta. Default is 0.001.

pr_xi_a Numeric; the first shape parameter of beta prior for latent variable xi. Default is 1.

pr_xi_b Numeric; the second shape parameter of beta prior for latent variable xi. Default is 1.

verbose Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.

fix_theta_sd Logical; If TRUE, the standard deviation of the theta parameter is fixed. Default is FALSE.

fix_alpha_1 Logical; If TRUE, the first element of the alpha parameter is fixed to 1. Default is TRUE.

adapt List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, accept_alpha, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/alpha/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

Details

lsirm2pl_ss models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j and the distance between latent position w_i of item i and latent position z_j of respondent j in the shared metric space, with γ represents the weight of the distance term. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$logit(P(Y_{j,i} = 1|\theta_j, \alpha_i, \beta_i, z_j, w_i)) = \theta_j * \alpha_i + \beta_i - \gamma||z_j - w_i||$$

lsirm2pl_ss model include model selection approach based on spike-and-slab priors for log gamma. For detail of spike-and-slab priors, see References.

Value

lsirm2pl_ss returns an object of list containing the following components:

data	Data frame or matrix containing the variables used in the model.
bic	A numeric value representing the Bayesian Information Criterion (BIC).
mcmc_inf	Details about the number of MCMC iterations, burn-in periods, and thinning intervals.
map_inf	The log maximum a posteriori (MAP) value and the iteration number at which this MAP value occurs.
beta_estimate	Posterior estimates of the beta parameter.
theta_estimate	Posterior estimates of the theta parameter.

sigma_theta_estimate
: Posterior estimates of the standard deviation of theta.

gamma_estimate
: Posterior estimates of gamma parameter.

z_estimate
: Posterior estimates of the z parameter.

w_estimate
: Posterior estimates of the w parameter.

beta
: Posterior samples of the beta parameter.

theta
: Posterior samples of the theta parameter.

theta_sd
: Posterior samples of the standard deviation of theta.

gamma
: Posterior samples of the gamma parameter.

z
: Posterior samples of the z parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.

w
: Posterior samples of the w parameter, represented as a 3-dimensional matrix where the last axis denotes the dimension of the latent space.

z_raw
: Posterior samples of the z parameter without procrustes matching.

w_raw
: Posterior samples of the w parameter without procrustes matching.

accept_beta
: Acceptance ratio for the beta parameter.

accept_theta
: Acceptance ratio for the theta parameter.

accept_z
: Acceptance ratio for the z parameter.

accept_w
: Acceptance ratio for the w parameter.

accept_gamma
: Acceptance ratio for the gamma parameter.

pi_estimate
: Posterior estimation of phi. inclusion probability of gamma. if estimation of phi is less than 0.5, choose Rasch model with gamma = 0, otherwise latent space model with gamma > 0.

pi
: Posterior samples of phi which is indicator of spike and slab prior. If phi is 1, log gamma follows the slab prior, otherwise follows the spike prior.

alpha_estimate
: Posterior estimates of the alpha parameter.

alpha
: Posterior estimates of the alpha parameter.

accept_alpha
: Acceptance ratio for the alpha parameter.

References

Ishwaran, H., & Rao, J. S. (2005). Spike and slab variable selection: Frequentist and Bayesian strategies. The Annals of Statistics, 33(2), 730-773.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

lsirm_result <- lsirm2pl_ss(data)

# The code following can achieve the same result.
lsirm_result <- lsirm(data ~ lsirm2pl(spikenslab = TRUE, fixed_gamma = FALSE))
```

lsirmgrm *Fit an ordinal LSIRM with the graded response model*

Description

lsirmgrm fits an ordinal latent space item response model for Likert-scale (ordered categorical) responses using the graded response model (GRM). This approach extends the traditional GRM by incorporating a latent space representation of respondent-item interactions, providing a spatial interpretation of response patterns.

The model captures interactions between respondents and items through the distance between latent respondent positions z_j and item positions w_i in a shared latent space, allowing for the visualization and interpretation of complex response patterns in Likert-scale assessments.

Usage

```
lsirmgrm(
  data,
  ncat = NULL,
  missing_data = NA,
  missing.val = NA,
  chains = 1,
  multicore = 1,
  seed = NA,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  fixed_gamma = FALSE,
  spikenslab = FALSE,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
```

```
  pr_slab_sd = 1,
  pr_xi_a = 1,
  pr_xi_b = 1,
  adapt = NULL,
  verbose = FALSE,
  fix_theta_sd = FALSE
)
```

Arguments

data	Matrix; an ordinal (ordered categorical) item response matrix. Each row represents a respondent, and each column represents an item. Values can be either 0:(K-1) or 1:K. Missing values can be NA.
ncat	Integer; number of categories K. If NULL, it is inferred from the observed data.
missing_data	Character; the type of missing data assumed. Options are NA or "mcar". For GRM models, missing values are currently supported only under MCAR. If data contains NA and missing_data is NA, it is set to "mcar" internally with a warning.
missing.val	Numeric; numeric code used to represent missing values in the C++ sampler. Default is 99.
chains	Integer; number of MCMC chains. Default is 1.
multicore	Integer; number of cores for parallel execution when chains > 1. Default is 1.
seed	Integer; RNG seed. Default is NA.
ndim	Integer; latent space dimension. Default is 2.
niter	Integer; total MCMC iterations. Default is 15000.
nburn	Integer; burn-in iterations. Default is 2500.
nthin	Integer; thinning interval. Default is 5.
nprint	Integer; print interval if verbose=TRUE. Default is 500.
jump_beta	Numeric; proposal SD for GRM thresholds. Default is 0.4. During MCMC sampling, threshold proposals are constrained to maintain the ordering $\beta_{i,1} > \beta_{i,2} > \cdots > \beta_{i,K-1}$ for each item.
jump_theta	Numeric; proposal SD for theta. Default is 1.
jump_gamma	Numeric; proposal SD on log-scale for gamma. Default is 0.2.
jump_z	Numeric; proposal SD for z. Default is 0.5.
jump_w	Numeric; proposal SD for w. Default is 0.5.
pr_mean_beta	Numeric; prior mean for thresholds. Default is 0.
pr_sd_beta	Numeric; prior SD for thresholds. Default is 1.
pr_mean_theta	Numeric; prior mean for theta. Default is 0.
pr_sd_theta	Numeric; prior SD for theta. Default is 1.
pr_mean_gamma	Numeric; log-normal prior mean for gamma. Default is 0.5.
pr_sd_gamma	Numeric; log-normal prior SD for gamma. Default is 1.
pr_a_theta	Numeric; shape for inverse-gamma prior on var(theta). Default is 0.001.

pr_b_theta	Numeric; scale for inverse-gamma prior on var(theta). Default is 0.001.
fixed_gamma	Logical; if TRUE, fixes $\gamma = 1$ (no sampling). Default is FALSE.
spikenslab	Logical; if TRUE, uses spike-and-slab priors for γ. Default is FALSE.
pr_spike_mean	Numeric; prior mean for the spike component (on log-scale). Default is -3.
pr_spike_sd	Numeric; prior SD for the spike component (on log-scale). Default is 1.
pr_slab_mean	Numeric; prior mean for the slab component (on log-scale). Default is 0.5.
pr_slab_sd	Numeric; prior SD for the slab component (on log-scale). Default is 1.
pr_xi_a	Numeric; Beta prior shape a for mixing weight ξ. Default is 1.
pr_xi_b	Numeric; Beta prior shape b for mixing weight ξ. Default is 1.

adapt

List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^α. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^α. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

verbose

Logical; If TRUE, MCMC progress and parameter samples are printed to the console during execution. Default is FALSE.

`fix_theta_sd` Logical; If TRUE, the standard deviation of the respondent latent positions θ is fixed at 1 instead of being sampled. Default is FALSE.

Details

`lsirmgrm` implements the Graded Response Model (GRM) in a latent space framework. Let $Y_{j,i} \in \{0, \ldots, K-1\}$ be the ordered categorical response of respondent j to item i. The model is defined via cumulative logits:

$$\Pr(Y_{j,i} \geq k | \theta_j, \beta_{i,k}, \gamma, z_j, w_i) = \text{logit}^{-1}(\theta_j + \beta_{i,k} - \gamma \|z_j - w_i\|)$$

for $k = 1, \ldots, K-1$, where $\beta_{i,k}$ are item-specific thresholds (difficulty levels) that satisfy the ordering constraint $\beta_{i,1} > \beta_{i,2} > \cdots > \beta_{i,K-1}$ for identifiability.

Missing data can be handled as follows:

- `"mcar"`: Missing responses are assumed to be Missing Completely At Random. They are ignored in the likelihood calculation.

For models with `spikenslab = TRUE`, a spike-and-slab prior is placed on $\log(\gamma)$ to perform model selection between a standard Rasch-type model ($\gamma \approx 0$) and a latent space model ($\gamma > 0$).

Value

An object of class `lsirm`. For multi-chain fits, it returns a list where each element (`chain1`, `chain2`, etc.) is a single-chain fit of class `lsirm`. The single-chain fit contains posterior summaries and MCMC draws, including:

- `z`, `w`: Procrustes-matched posterior samples of respondent and item latent positions.
- `z_raw`, `w_raw`: posterior samples of respondent and item latent positions before Procrustes matching.
- `z_estimate`, `w_estimate`: posterior mean latent positions after Procrustes matching.

References

De Carolis, L., Kang, I., & Jeon, M. (2025). A Latent Space Graded Response Model for Likert-Scale Psychological Assessments. *Multivariate Behavioral Research*. doi:10.1080/00273171.2025.2605678

Roberts, G. O., Gelman, A., & Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1), 110–120. doi:10.1214/aoap/1034625254

Roberts, G. O., & Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4), 351–367. doi:10.1214/ss/1015346320

Andrieu, C., & Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4), 343–373. doi:10.1007/s112220089110y

Examples

```
# generate example ordinal item response matrix
set.seed(123)
nsample <- 50
nitem <- 10
data <- matrix(sample(1:5, nsample * nitem, replace = TRUE), nrow = nsample)

# Fit GRM LSIRM using direct function call
fit <- lsirmgrm(data, niter = 1000, nburn = 500, nthin = 2)
summary(fit)
```

`lsirmgrm2pl` *Fit an ordinal LSIRM with the graded response model (2PL)*

Description

`lsirmgrm2pl` fits a two-parameter logistic (2PL) extension of the ordinal latent space item response model using the graded response model (GRM) with item discrimination parameters α_i. The model factorizes the item response matrix into item thresholds, respondent ability, and item discrimination, while embedding interaction effects in a latent space. The resulting latent space provides an interaction map that visualizes the complex relationships between respondents and items beyond traditional IRT models.

Usage

```
lsirmgrm2pl(
  data,
  ncat = NULL,
  missing_data = NA,
  missing.val = NA,
  chains = 1,
  multicore = 1,
  seed = NA,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
```

```
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  fixed_gamma = FALSE,
  spikenslab = FALSE,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_xi_a = 1,
  pr_xi_b = 1,
  adapt = NULL,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE
)
```

Arguments

`data`	Matrix; an ordinal (ordered categorical) item response matrix. Each row represents a respondent, and each column represents an item. Values can be either `0:(K-1)` or `1:K`. Missing values can be `NA`.
`ncat`	Integer; number of categories K. If `NULL`, it is inferred from the observed data.
`missing_data`	Character; the type of missing data assumed. Options are `NA` or `"mcar"`. For GRM models, missing values are currently supported only under MCAR. If `data` contains `NA` and `missing_data` is `NA`, it is set to `"mcar"` internally with a warning.
`missing.val`	Numeric; numeric code used to represent missing values in the C++ sampler. Default is 99.
`chains`	Integer; number of MCMC chains. Default is 1.
`multicore`	Integer; number of cores for parallel execution when `chains > 1`. Default is 1.
`seed`	Integer; RNG seed. Default is `NA`.
`ndim`	Integer; latent space dimension. Default is 2.
`niter`	Integer; total MCMC iterations. Default is 15000.
`nburn`	Integer; burn-in iterations. Default is 2500.
`nthin`	Integer; thinning interval. Default is 5.
`nprint`	Integer; print interval if `verbose=TRUE`. Default is 500.
`jump_beta`	Numeric; proposal SD for GRM thresholds. Default is 0.4. During MCMC sampling, threshold proposals are constrained to maintain the ordering $\beta_{i,1} > \beta_{i,2} > \cdots > \beta_{i,K-1}$ for each item.

jump_theta Numeric; proposal SD for theta. Default is 1.

jump_alpha Numeric; proposal SD on log-scale for α. Default is 1.

jump_gamma Numeric; proposal SD on log-scale for gamma. Default is 0.2.

jump_z Numeric; proposal SD for z. Default is 0.5.

jump_w Numeric; proposal SD for w. Default is 0.5.

pr_mean_beta Numeric; prior mean for thresholds. Default is 0.

pr_sd_beta Numeric; prior SD for thresholds. Default is 1.

pr_mean_theta Numeric; prior mean for theta. Default is 0.

pr_sd_theta Numeric; prior SD for theta. Default is 1.

pr_mean_alpha Numeric; log-normal prior mean for α. Default is 0.5.

pr_sd_alpha Numeric; log-normal prior SD for α. Default is 1.

pr_mean_gamma Numeric; log-normal prior mean for gamma. Default is 0.5.

pr_sd_gamma Numeric; log-normal prior SD for gamma. Default is 1.

pr_a_theta Numeric; shape for inverse-gamma prior on var(theta). Default is 0.001.

pr_b_theta Numeric; scale for inverse-gamma prior on var(theta). Default is 0.001.

fixed_gamma Logical; if TRUE, fixes $\gamma = 1$ (no sampling). Default is FALSE.

spikenslab Logical; if TRUE, uses spike-and-slab priors for γ. Default is FALSE.

pr_spike_mean Numeric; prior mean for the spike component (on log-scale). Default is -3.

pr_spike_sd Numeric; prior SD for the spike component (on log-scale). Default is 1.

pr_slab_mean Numeric; prior mean for the slab component (on log-scale). Default is 0.5.

pr_slab_sd Numeric; prior SD for the slab component (on log-scale). Default is 1.

pr_xi_a Numeric; Beta prior shape a for mixing weight ξ. Default is 1.

pr_xi_b Numeric; Beta prior shape b for mixing weight ξ. Default is 1.

adapt List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, accept_alpha, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.

- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/alpha/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

verbose: Logical; If TRUE, MCMC progress and parameter samples are printed to the console during execution. Default is FALSE.

fix_theta_sd: Logical; If TRUE, the standard deviation of the respondent latent positions θ is fixed at 1 instead of being sampled. Default is FALSE.

fix_alpha_1: Logical; if TRUE, fixes $\alpha_1 = 1$. Default is TRUE.

Details

lsirmgrm2pl implements the 2PL extension of the Graded Response Model (GRM) in a latent space framework. Let $Y_{j,i} \in \{0, \ldots, K-1\}$ be the ordered categorical response of respondent j to item i. The model is defined via cumulative logits:

$$\Pr(Y_{j,i} \geq k | \theta_j, \alpha_i, \beta_{i,k}, \gamma, z_j, w_i) = \text{logit}^{-1}(\alpha_i \theta_j + \beta_{i,k} - \gamma \left\| z_j - w_i \right\|)$$

for $k = 1, \ldots, K-1$, where α_i is the item discrimination parameter and $\beta_{i,k}$ are item-specific thresholds that satisfy the ordering constraint $\beta_{i,1} > \beta_{i,2} > \cdots > \beta_{i,K-1}$ for identifiability.

Missing data handling:

- "mcar": missing responses are excluded from the likelihood.

Value

An object of class lsirm. For multi-chain fits, a list where each element (chain1, chain2, etc.) is a single-chain fit of class lsirm. The single-chain fit contains posterior summaries and MCMC draws, including:

- z, w: Procrustes-matched posterior samples of respondent and item latent positions.
- z_raw, w_raw: posterior samples of respondent and item latent positions before Procrustes matching.
- z_estimate, w_estimate: posterior mean latent positions after Procrustes matching.

References

De Carolis, L., Kang, I., & Jeon, M. (2025). A Latent Space Graded Response Model for Likert-Scale Psychological Assessments. *Multivariate Behavioral Research*. doi:10.1080/00273171.2025.2605678

Roberts, G. O., Gelman, A., & Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1), 110–120. doi:10.1214/aoap/1034625254

Roberts, G. O., & Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4), 351–367. doi:10.1214/ss/1015346320

Andrieu, C., & Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4), 343–373. doi:10.1007/s112220089110y

Examples

```
# Generate example ordinal item response matrix
set.seed(123)
nsample <- 50
nitem <- 10
data <- matrix(sample(1:5, nsample * nitem, replace = TRUE), nrow = nsample)

# Fit 2PL GRM LSIRM using direct function call
fit <- lsirmgrm2pl(data, niter = 1000, nburn = 500, nthin = 2)
summary(fit)
```

---

lsirmgrm2pl_fixed_gamma

2PL GRM LSIRM fixing gamma to 1.

Description

lsirmgrm2pl_fixed_gamma is used to fit 2PL GRM LSIRM with gamma fixed to 1.

Usage

```
lsirmgrm2pl_fixed_gamma(
  data,
  ncat = NULL,
  missing_data = NA,
  missing.val = NA,
  chains = 1,
  multicore = 1,
  seed = NA,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
```

```
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  adapt = NULL,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE
)
```

Arguments

data	Matrix; an ordinal (ordered categorical) item response matrix. Each row represents a respondent, and each column represents an item. Values can be either `0:(K-1)` or `1:K`. Missing values can be `NA`.
ncat	Integer; number of categories K. If `NULL`, it is inferred from the observed data.
missing_data	Character; the type of missing data assumed. Options are `NA` or `"mcar"`. For GRM models, missing values are currently supported only under MCAR. If `data` contains `NA` and `missing_data` is `NA`, it is set to `"mcar"` internally with a warning.
missing.val	Numeric; numeric code used to represent missing values in the C++ sampler. Default is 99.
chains	Integer; number of MCMC chains. Default is 1.
multicore	Integer; number of cores for parallel execution when `chains` > 1. Default is 1.
seed	Integer; RNG seed. Default is `NA`.
ndim	Integer; latent space dimension. Default is 2.
niter	Integer; total MCMC iterations. Default is 15000.
nburn	Integer; burn-in iterations. Default is 2500.
nthin	Integer; thinning interval. Default is 5.
nprint	Integer; print interval if `verbose=TRUE`. Default is 500.
jump_beta	Numeric; proposal SD for GRM thresholds. Default is 0.4. During MCMC sampling, threshold proposals are constrained to maintain the ordering $\beta_{i,1} > \beta_{i,2} > \cdots > \beta_{i,K-1}$ for each item.
jump_theta	Numeric; proposal SD for theta. Default is 1.
jump_alpha	Numeric; proposal SD on log-scale for α. Default is 1.

jump_z Numeric; proposal SD for z. Default is 0.5.

jump_w Numeric; proposal SD for w. Default is 0.5.

pr_mean_beta Numeric; prior mean for thresholds. Default is 0.

pr_sd_beta Numeric; prior SD for thresholds. Default is 1.

pr_mean_theta Numeric; prior mean for theta. Default is 0.

pr_sd_theta Numeric; prior SD for theta. Default is 1.

pr_mean_alpha Numeric; log-normal prior mean for α. Default is 0.5.

pr_sd_alpha Numeric; log-normal prior SD for α. Default is 1.

pr_a_theta Numeric; shape for inverse-gamma prior on var(theta). Default is 0.001.

pr_b_theta Numeric; scale for inverse-gamma prior on var(theta). Default is 0.001.

adapt List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, accept_alpha, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/alpha/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

verbose — Logical; If TRUE, MCMC samples are printed for each `nprint`. default value is FALSE

fix_theta_sd — Logical; If TRUE, the standard deviation of the respondent latent positions θ is fixed at 1 instead of being sampled. Default is FALSE.

fix_alpha_1 — Logical; if TRUE, fixes $\alpha_1 = 1$. Default is TRUE.

Value

`lsirmgrm2pl_fixed_gamma` returns an object of list containing the same components as `lsirmgrm2pl`.

Examples

```
# generate example ordinal item response matrix
set.seed(123)
nsample <- 50
nitem <- 10
data <- matrix(sample(1:5, nsample * nitem, replace = TRUE), nrow = nsample)

# Fit 2PL GRM LSIRM with fixed gamma = 1
fit <- lsirmgrm2pl_fixed_gamma(data, niter = 1000, nburn = 500)
summary(fit)
```

---

lsirmgrm2pl_mcar — *2PL GRM LSIRM for missing completely at random data.*

Description

lsirmgrm2pl_mcar is used to fit 2PL GRM LSIRM in incomplete data assumed to be missing completely at random.

Usage

```
lsirmgrm2pl_mcar(
  data,
  ncat = NULL,
  missing.val = NA,
  chains = 1,
  multicore = 1,
  seed = NA,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
```

```
  jump_alpha = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  adapt = NULL,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE
)
```

Arguments

`data`	Matrix; an ordinal (ordered categorical) item response matrix. Each row represents a respondent, and each column represents an item. Values can be either `0:(K-1)` or `1:K`. Missing values can be `NA`.
`ncat`	Integer; number of categories K. If `NULL`, it is inferred from the observed data.
`missing.val`	Numeric; numeric code used to represent missing values in the C++ sampler. Default is 99.
`chains`	Integer; number of MCMC chains. Default is 1.
`multicore`	Integer; number of cores for parallel execution when `chains` > `1`. Default is 1.
`seed`	Integer; RNG seed. Default is `NA`.
`ndim`	Integer; latent space dimension. Default is 2.
`niter`	Integer; total MCMC iterations. Default is 15000.
`nburn`	Integer; burn-in iterations. Default is 2500.
`nthin`	Integer; thinning interval. Default is 5.
`nprint`	Integer; print interval if `verbose=TRUE`. Default is 500.
`jump_beta`	Numeric; proposal SD for GRM thresholds. Default is 0.4. During MCMC sampling, threshold proposals are constrained to maintain the ordering $\beta_{i,1} > \beta_{i,2} > \cdots > \beta_{i,K-1}$ for each item.
`jump_theta`	Numeric; proposal SD for theta. Default is 1.
`jump_alpha`	Numeric; proposal SD on log-scale for α. Default is 1.
`jump_gamma`	Numeric; proposal SD on log-scale for gamma. Default is 0.2.
`jump_z`	Numeric; proposal SD for z. Default is 0.5.
`jump_w`	Numeric; proposal SD for w. Default is 0.5.

pr_mean_beta Numeric; prior mean for thresholds. Default is 0.

pr_sd_beta Numeric; prior SD for thresholds. Default is 1.

pr_mean_theta Numeric; prior mean for theta. Default is 0.

pr_sd_theta Numeric; prior SD for theta. Default is 1.

pr_mean_alpha Numeric; log-normal prior mean for α. Default is 0.5.

pr_sd_alpha Numeric; log-normal prior SD for α. Default is 1.

pr_mean_gamma Numeric; log-normal prior mean for gamma. Default is 0.5.

pr_sd_gamma Numeric; log-normal prior SD for gamma. Default is 1.

pr_a_theta Numeric; shape for inverse-gamma prior on var(theta). Default is 0.001.

pr_b_theta Numeric; scale for inverse-gamma prior on var(theta). Default is 0.001.

adapt List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, accept_alpha, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^α. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^α. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is 0.44.
- target_accept_zw: Numeric; target acceptance rate for latent position block updates z and w. Default is 0.234.
- target_accept_beta/theta/alpha/gamma: Numeric; (optional) parameter-specific target acceptance rates to override target_accept.

verbose	Logical; If TRUE, MCMC samples are printed for each nprint. Default is FALSE.
fix_theta_sd	Logical; If TRUE, the standard deviation of the respondent latent positions θ is fixed at 1 instead of being sampled. Default is FALSE.
fix_alpha_1	Logical; if TRUE, fixes $\alpha_1 = 1$. Default is TRUE.

Examples

```
# generate example ordinal item response matrix
set.seed(123)
nsample <- 50
nitem <- 10
data <- matrix(sample(1:5, nsample * nitem, replace = TRUE), nrow = nsample)

# generate missing value (MCAR)
data[sample(1:500, 50)] <- NA

# Fit 2PL GRM LSIRM with MCAR
fit <- lsirmgrm2pl_mcar(data, niter = 1000, nburn = 500)
summary(fit)
```

---

lsirmgrm2pl_o *Fit a single-chain ordinal LSIRM (GRM 2PL)*

Description

Fit a single-chain ordinal LSIRM (GRM 2PL)

Usage

```
lsirmgrm2pl_o(
  data,
  ncat = NULL,
  missing_data = NA,
  missing.val = NA,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
```

```
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  fixed_gamma = FALSE,
  spikenslab = FALSE,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_xi_a = 1,
  pr_xi_b = 1,
  adapt = NULL,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE
)
```

Arguments

data	Matrix; an ordinal (ordered categorical) item response matrix. Each row represents a respondent, and each column represents an item. Values can be either 0:(K-1) or 1:K. Missing values can be NA.
ncat	Integer; number of categories K. If NULL, it is inferred from the observed data.
missing_data	Character; the type of missing data assumed. Options are NA or "mcar". For GRM models, missing values are currently supported only under MCAR. If data contains NA and missing_data is NA, it is set to "mcar" internally with a warning.
missing.val	Numeric; numeric code used to represent missing values in the C++ sampler. Default is 99.
ndim	Integer; latent space dimension. Default is 2.
niter	Integer; total MCMC iterations. Default is 15000.
nburn	Integer; burn-in iterations. Default is 2500.
nthin	Integer; thinning interval. Default is 5.
nprint	Integer; print interval if verbose=TRUE. Default is 500.
jump_beta	Numeric; proposal SD for GRM thresholds. Default is 0.4. During MCMC sampling, threshold proposals are constrained to maintain the ordering $\beta_{i,1} > \beta_{i,2} > \cdots > \beta_{i,K-1}$ for each item.
jump_theta	Numeric; proposal SD for theta. Default is 1.
jump_alpha	Numeric; proposal SD on log-scale for α. Default is 1.

jump_gamma	Numeric; proposal SD on log-scale for gamma. Default is 0.2.
jump_z	Numeric; proposal SD for z. Default is 0.5.
jump_w	Numeric; proposal SD for w. Default is 0.5.
pr_mean_beta	Numeric; prior mean for thresholds. Default is 0.
pr_sd_beta	Numeric; prior SD for thresholds. Default is 1.
pr_mean_theta	Numeric; prior mean for theta. Default is 0.
pr_sd_theta	Numeric; prior SD for theta. Default is 1.
pr_mean_alpha	Numeric; log-normal prior mean for α. Default is 0.5.
pr_sd_alpha	Numeric; log-normal prior SD for α. Default is 1.
pr_mean_gamma	Numeric; log-normal prior mean for gamma. Default is 0.5.
pr_sd_gamma	Numeric; log-normal prior SD for gamma. Default is 1.
pr_a_theta	Numeric; shape for inverse-gamma prior on var(theta). Default is 0.001.
pr_b_theta	Numeric; scale for inverse-gamma prior on var(theta). Default is 0.001.
fixed_gamma	Logical; if TRUE, fixes $\gamma = 1$ (no sampling). Default is FALSE.
spikenslab	Logical; if TRUE, uses spike-and-slab priors for γ. Default is FALSE.
pr_spike_mean	Numeric; prior mean for the spike component (on log-scale). Default is -3.
pr_spike_sd	Numeric; prior SD for the spike component (on log-scale). Default is 1.
pr_slab_mean	Numeric; prior mean for the slab component (on log-scale). Default is 0.5.
pr_slab_sd	Numeric; prior SD for the slab component (on log-scale). Default is 1.
pr_xi_a	Numeric; Beta prior shape a for mixing weight ξ. Default is 1.
pr_xi_b	Numeric; Beta prior shape b for mixing weight ξ. Default is 1.

adapt

List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, `accept_alpha`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5–2.0`.

- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.
- `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is `0.44`.
- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/alpha/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

`verbose` Logical; If TRUE, MCMC progress and parameter samples are printed to the console during execution. Default is FALSE.

`fix_theta_sd` Logical; If TRUE, the standard deviation of the respondent latent positions θ is fixed at 1 instead of being sampled. Default is FALSE.

`fix_alpha_1` Logical; if TRUE, fixes $\alpha_1 = 1$. Default is TRUE.

Value

A list containing MCMC draws and posterior summaries, including:

- `beta`, `theta`, `gamma`, `alpha`: MCMC draws.
- `z`, `w`: Procrustes-matched MCMC draws of respondent and item latent positions.
- `z_raw`, `w_raw`: MCMC draws of respondent and item latent positions before Procrustes matching.
- `beta_estimate`, `theta_estimate`, `gamma_estimate`, `alpha_estimate`, `z_estimate`, `w_estimate`: posterior means, with `z_estimate` and `w_estimate` computed after Procrustes matching.

`lsirmgrm2pl_ss` *2PL GRM LSIRM with spike-and-slab prior.*

Description

lsirmgrm2pl_ss is used to fit 2PL GRM LSIRM with spike-and-slab prior.

Usage

```
lsirmgrm2pl_ss(
  data,
  ncat = NULL,
  missing_data = NA,
  missing.val = NA,
  chains = 1,
```

```
  multicore = 1,
  seed = NA,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_xi_a = 1,
  pr_xi_b = 1,
  adapt = NULL,
  verbose = FALSE,
  fix_theta_sd = FALSE,
  fix_alpha_1 = TRUE
)
```

Arguments

data: Matrix; an ordinal (ordered categorical) item response matrix. Each row represents a respondent, and each column represents an item. Values can be either `0:(K-1)` or `1:K`. Missing values can be `NA`.

ncat: Integer; number of categories K. If `NULL`, it is inferred from the observed data.

missing_data: Character; the type of missing data assumed. Options are `NA` or `"mcar"`. For GRM models, missing values are currently supported only under MCAR. If `data` contains `NA` and `missing_data` is `NA`, it is set to `"mcar"` internally with a warning.

missing.val: Numeric; numeric code used to represent missing values in the C++ sampler. Default is 99.

chains: Integer; number of MCMC chains. Default is 1.

multicore: Integer; number of cores for parallel execution when `chains` > `1`. Default is 1.

seed	Integer; RNG seed. Default is NA.
ndim	Integer; latent space dimension. Default is 2.
niter	Integer; total MCMC iterations. Default is 15000.
nburn	Integer; burn-in iterations. Default is 2500.
nthin	Integer; thinning interval. Default is 5.
nprint	Integer; print interval if verbose=TRUE. Default is 500.
jump_beta	Numeric; proposal SD for GRM thresholds. Default is 0.4. During MCMC sampling, threshold proposals are constrained to maintain the ordering $\beta_{i,1} > \beta_{i,2} > \cdots > \beta_{i,K-1}$ for each item.
jump_theta	Numeric; proposal SD for theta. Default is 1.
jump_alpha	Numeric; proposal SD on log-scale for α. Default is 1.
jump_gamma	Numeric; proposal SD on log-scale for gamma. Default is 0.2.
jump_z	Numeric; proposal SD for z. Default is 0.5.
jump_w	Numeric; proposal SD for w. Default is 0.5.
pr_mean_beta	Numeric; prior mean for thresholds. Default is 0.
pr_sd_beta	Numeric; prior SD for thresholds. Default is 1.
pr_mean_theta	Numeric; prior mean for theta. Default is 0.
pr_sd_theta	Numeric; prior SD for theta. Default is 1.
pr_mean_alpha	Numeric; log-normal prior mean for α. Default is 0.5.
pr_sd_alpha	Numeric; log-normal prior SD for α. Default is 1.
pr_a_theta	Numeric; shape for inverse-gamma prior on var(theta). Default is 0.001.
pr_b_theta	Numeric; scale for inverse-gamma prior on var(theta). Default is 0.001.
pr_spike_mean	Numeric; prior mean for the spike component (on log-scale). Default is -3.
pr_spike_sd	Numeric; prior SD for the spike component (on log-scale). Default is 1.
pr_slab_mean	Numeric; prior mean for the slab component (on log-scale). Default is 0.5.
pr_slab_sd	Numeric; prior SD for the slab component (on log-scale). Default is 1.
pr_xi_a	Numeric; Beta prior shape a for mixing weight ξ. Default is 1.
pr_xi_b	Numeric; Beta prior shape b for mixing weight ξ. Default is 1.
adapt	List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, accept_alpha, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.
- `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma, alpha). Default is `0.44`.
- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/alpha/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

`verbose` Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.

`fix_theta_sd` Logical; If TRUE, the standard deviation of the respondent latent positions θ is fixed at 1 instead of being sampled. Default is FALSE.

`fix_alpha_1` Logical; if TRUE, fixes $\alpha_1 = 1$. Default is TRUE.

Examples

```
# generate example ordinal item response matrix
set.seed(123)
nsample <- 50
nitem <- 10
data <- matrix(sample(1:5, nsample * nitem, replace = TRUE), nrow = nsample)

# Fit 2PL GRM LSIRM with Spike-and-Slab
fit <- lsirmgrm2pl_ss(data, niter = 1000, nburn = 500)
summary(fit)
```

`lsirmgrm_fixed_gamma` *1PL GRM LSIRM fixing gamma to 1.*

Description

lsirmgrm_fixed_gamma is used to fit 1PL GRM LSIRM with gamma fixed to 1.

Usage

```
lsirmgrm_fixed_gamma(
  data,
  ncat = NULL,
  missing_data = NA,
  missing.val = NA,
  chains = 1,
  multicore = 1,
  seed = NA,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  adapt = NULL,
  verbose = FALSE,
  fix_theta_sd = FALSE
)
```

Arguments

data	Matrix; an ordinal (ordered categorical) item response matrix. Each row represents a respondent, and each column represents an item. Values can be either `0:(K-1)` or `1:K`. Missing values can be `NA`.
ncat	Integer; number of categories K. If `NULL`, it is inferred from the observed data.
missing_data	Character; the type of missing data assumed. Options are `NA` or `"mcar"`. For GRM models, missing values are currently supported only under MCAR. If `data` contains `NA` and `missing_data` is `NA`, it is set to `"mcar"` internally with a warning.
missing.val	Numeric; numeric code used to represent missing values in the C++ sampler. Default is 99.
chains	Integer; number of MCMC chains. Default is 1.
multicore	Integer; number of cores for parallel execution when `chains` > `1`. Default is 1.
seed	Integer; RNG seed. Default is `NA`.
ndim	Integer; latent space dimension. Default is 2.
niter	Integer; total MCMC iterations. Default is 15000.

nburn	Integer; burn-in iterations. Default is 2500.
nthin	Integer; thinning interval. Default is 5.
nprint	Integer; print interval if verbose=TRUE. Default is 500.
jump_beta	Numeric; proposal SD for GRM thresholds. Default is 0.4. During MCMC sampling, threshold proposals are constrained to maintain the ordering $\beta_{i,1} > \beta_{i,2} > \cdots > \beta_{i,K-1}$ for each item.
jump_theta	Numeric; proposal SD for theta. Default is 1.
jump_z	Numeric; proposal SD for z. Default is 0.5.
jump_w	Numeric; proposal SD for w. Default is 0.5.
pr_mean_beta	Numeric; prior mean for thresholds. Default is 0.
pr_sd_beta	Numeric; prior SD for thresholds. Default is 1.
pr_mean_theta	Numeric; prior mean for theta. Default is 0.
pr_sd_theta	Numeric; prior SD for theta. Default is 1.
pr_a_theta	Numeric; shape for inverse-gamma prior on var(theta). Default is 0.001.
pr_b_theta	Numeric; scale for inverse-gamma prior on var(theta). Default is 0.001.

adapt — List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^α. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^α. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.

- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is `0.44`.
- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

`verbose` Logical; If TRUE, MCMC samples are printed for each `nprint`. default value is FALSE

`fix_theta_sd` Logical; If TRUE, the standard deviation of the respondent latent positions θ is fixed at 1 instead of being sampled. Default is FALSE.

Value

`lsirmgrm_fixed_gamma` returns an object of list containing the same components as `lsirmgrm`.

Examples

```
# generate example ordinal item response matrix
set.seed(123)
nsample <- 50
nitem <- 10
data <- matrix(sample(1:5, nsample * nitem, replace = TRUE), nrow = nsample)

# Fit 1PL GRM LSIRM with fixed gamma = 1
fit <- lsirmgrm_fixed_gamma(data, niter = 1000, nburn = 500)
summary(fit)
```

`lsirmgrm_mcar` *1PL GRM LSIRM for missing completely at random data.*

Description

lsirmgrm_mcar is used to fit 1PL GRM LSIRM in incomplete data assumed to be missing completely at random.

Usage

```
lsirmgrm_mcar(
  data,
  ncat = NULL,
  missing.val = NA,
  chains = 1,
  multicore = 1,
  seed = NA,
  ndim = 2,
```

```
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  adapt = NULL,
  verbose = FALSE,
  fix_theta_sd = FALSE
)
```

Arguments

data	Matrix; an ordinal (ordered categorical) item response matrix. Each row represents a respondent, and each column represents an item. Values can be either `0:(K-1)` or `1:K`. Missing values can be `NA`.
ncat	Integer; number of categories K. If `NULL`, it is inferred from the observed data.
missing.val	Numeric; numeric code used to represent missing values in the C++ sampler. Default is 99.
chains	Integer; number of MCMC chains. Default is 1.
multicore	Integer; number of cores for parallel execution when `chains > 1`. Default is 1.
seed	Integer; RNG seed. Default is `NA`.
ndim	Integer; latent space dimension. Default is 2.
niter	Integer; total MCMC iterations. Default is 15000.
nburn	Integer; burn-in iterations. Default is 2500.
nthin	Integer; thinning interval. Default is 5.
nprint	Integer; print interval if `verbose=TRUE`. Default is 500.
jump_beta	Numeric; proposal SD for GRM thresholds. Default is 0.4. During MCMC sampling, threshold proposals are constrained to maintain the ordering $\beta_{i,1} > \beta_{i,2} > \cdots > \beta_{i,K-1}$ for each item.
jump_theta	Numeric; proposal SD for theta. Default is 1.
jump_gamma	Numeric; proposal SD on log-scale for gamma. Default is 0.2.
jump_z	Numeric; proposal SD for z. Default is 0.5.

`jump_w` Numeric; proposal SD for w. Default is 0.5.

`pr_mean_beta` Numeric; prior mean for thresholds. Default is 0.

`pr_sd_beta` Numeric; prior SD for thresholds. Default is 1.

`pr_mean_theta` Numeric; prior mean for theta. Default is 0.

`pr_sd_theta` Numeric; prior SD for theta. Default is 1.

`pr_mean_gamma` Numeric; log-normal prior mean for gamma. Default is 0.5.

`pr_sd_gamma` Numeric; log-normal prior SD for gamma. Default is 1.

`pr_a_theta` Numeric; shape for inverse-gamma prior on var(theta). Default is 0.001.

`pr_b_theta` Numeric; scale for inverse-gamma prior on var(theta). Default is 0.001.

`adapt` List; optional adaptive MCMC control. If not `NULL`, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (`accept_beta`, `accept_theta`, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of `0.44`, while latent position block proposals use `0.234`. The `0.44` target is commonly used for one-dimensional random-walk proposals, and `0.234` is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- `use_adapt`: Logical; if `TRUE`, adaptive MCMC is used. Default is `FALSE`.
- `adapt_interval`: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is `100`.
- `adapt_rate`: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^α. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is `1.0`. Recommended starting range is `0.5`–`2.0`.
- `decay_rate`: Numeric; Robbins-Monro decay exponent α in c/t^α. Values above `0.5` give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is `0.6`; recommended range is `0.6`–`0.8`.
- `jump_min`, `jump_max`: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- `target_accept`: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is `0.44`.
- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

`verbose` Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.

fix_theta_sd — Logical; If TRUE, the standard deviation of the respondent latent positions θ is fixed at 1 instead of being sampled. Default is FALSE.

Examples

```
# generate example ordinal item response matrix
set.seed(123)
nsample <- 50
nitem <- 10
data <- matrix(sample(1:5, nsample * nitem, replace = TRUE), nrow = nsample)

# generate missing value (MCAR)
data[sample(1:500, 50)] <- NA

# Fit 1PL GRM LSIRM with MCAR
fit <- lsirmgrm_mcar(data, niter = 1000, nburn = 500)
summary(fit)
```

---

lsirmgrm_o — *Fit a single-chain ordinal LSIRM (GRM)*

Description

lsirmgrm_o is the single-chain fitting function used internally by lsirmgrm.

Usage

```
lsirmgrm_o(
  data,
  ncat = NULL,
  missing_data = NA,
  missing.val = NA,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
```

```
  pr_mean_gamma = 0.5,
  pr_sd_gamma = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  fixed_gamma = FALSE,
  spikenslab = FALSE,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_xi_a = 1,
  pr_xi_b = 1,
  adapt = NULL,
  verbose = FALSE,
  fix_theta_sd = FALSE
)
```

Arguments

data	Matrix; an ordinal (ordered categorical) item response matrix. Each row represents a respondent, and each column represents an item. Values can be either `0:(K-1)` or `1:K`. Missing values can be `NA`.
ncat	Integer; number of categories K. If `NULL`, it is inferred from the observed data.
missing_data	Character; the type of missing data assumed. Options are `NA` or `"mcar"`. For GRM models, missing values are currently supported only under MCAR. If `data` contains `NA` and `missing_data` is `NA`, it is set to `"mcar"` internally with a warning.
missing.val	Numeric; numeric code used to represent missing values in the C++ sampler. Default is 99.
ndim	Integer; latent space dimension. Default is 2.
niter	Integer; total MCMC iterations. Default is 15000.
nburn	Integer; burn-in iterations. Default is 2500.
nthin	Integer; thinning interval. Default is 5.
nprint	Integer; print interval if `verbose=TRUE`. Default is 500.
jump_beta	Numeric; proposal SD for GRM thresholds. Default is 0.4. During MCMC sampling, threshold proposals are constrained to maintain the ordering $\beta_{i,1} > \beta_{i,2} > \cdots > \beta_{i,K-1}$ for each item.
jump_theta	Numeric; proposal SD for theta. Default is 1.
jump_gamma	Numeric; proposal SD on log-scale for gamma. Default is 0.2.
jump_z	Numeric; proposal SD for z. Default is 0.5.
jump_w	Numeric; proposal SD for w. Default is 0.5.
pr_mean_beta	Numeric; prior mean for thresholds. Default is 0.
pr_sd_beta	Numeric; prior SD for thresholds. Default is 1.
pr_mean_theta	Numeric; prior mean for theta. Default is 0.

pr_sd_theta Numeric; prior SD for theta. Default is 1.

pr_mean_gamma Numeric; log-normal prior mean for gamma. Default is 0.5.

pr_sd_gamma Numeric; log-normal prior SD for gamma. Default is 1.

pr_a_theta Numeric; shape for inverse-gamma prior on var(theta). Default is 0.001.

pr_b_theta Numeric; scale for inverse-gamma prior on var(theta). Default is 0.001.

fixed_gamma Logical; if TRUE, fixes $\gamma = 1$ (no sampling). Default is FALSE.

spikenslab Logical; if TRUE, uses spike-and-slab priors for γ. Default is FALSE.

pr_spike_mean Numeric; prior mean for the spike component (on log-scale). Default is -3.

pr_spike_sd Numeric; prior SD for the spike component (on log-scale). Default is 1.

pr_slab_mean Numeric; prior mean for the slab component (on log-scale). Default is 0.5.

pr_slab_sd Numeric; prior SD for the slab component (on log-scale). Default is 1.

pr_xi_a Numeric; Beta prior shape a for mixing weight ξ. Default is 1.

pr_xi_b Numeric; Beta prior shape b for mixing weight ξ. Default is 1.

adapt List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.

- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

`verbose` Logical; If TRUE, MCMC progress and parameter samples are printed to the console during execution. Default is FALSE.

`fix_theta_sd` Logical; If TRUE, the standard deviation of the respondent latent positions θ is fixed at 1 instead of being sampled. Default is FALSE.

Value

A list containing MCMC draws and posterior summaries, including:

- `beta`, `theta`, `gamma`: MCMC draws.
- `z`, `w`: Procrustes-matched MCMC draws of respondent and item latent positions.
- `z_raw`, `w_raw`: MCMC draws of respondent and item latent positions before Procrustes matching.
- `beta_estimate`, `theta_estimate`, `gamma_estimate`, `z_estimate`, `w_estimate`: posterior means, with `z_estimate` and `w_estimate` computed after Procrustes matching.

---

lsirmgrm_ss *1PL GRM LSIRM with spike-and-slab prior.*

Description

lsirmgrm_ss is used to fit 1PL GRM LSIRM with spike-and-slab prior.

Usage

```
lsirmgrm_ss(
  data,
  ncat = NULL,
  missing_data = NA,
  missing.val = NA,
  chains = 1,
  multicore = 1,
  seed = NA,
  ndim = 2,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_gamma = 0.2,
  jump_z = 0.5,
```

```
  jump_w = 0.5,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_sd_theta = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001,
  pr_spike_mean = -3,
  pr_spike_sd = 1,
  pr_slab_mean = 0.5,
  pr_slab_sd = 1,
  pr_xi_a = 1,
  pr_xi_b = 1,
  adapt = NULL,
  verbose = FALSE,
  fix_theta_sd = FALSE
)
```

Arguments

data	Matrix; an ordinal (ordered categorical) item response matrix. Each row represents a respondent, and each column represents an item. Values can be either `0:(K-1)` or `1:K`. Missing values can be `NA`.
ncat	Integer; number of categories K. If `NULL`, it is inferred from the observed data.
missing_data	Character; the type of missing data assumed. Options are `NA` or `"mcar"`. For GRM models, missing values are currently supported only under MCAR. If `data` contains `NA` and `missing_data` is `NA`, it is set to `"mcar"` internally with a warning.
missing.val	Numeric; numeric code used to represent missing values in the C++ sampler. Default is 99.
chains	Integer; number of MCMC chains. Default is 1.
multicore	Integer; number of cores for parallel execution when `chains` > `1`. Default is 1.
seed	Integer; RNG seed. Default is `NA`.
ndim	Integer; latent space dimension. Default is 2.
niter	Integer; total MCMC iterations. Default is 15000.
nburn	Integer; burn-in iterations. Default is 2500.
nthin	Integer; thinning interval. Default is 5.
nprint	Integer; print interval if `verbose=TRUE`. Default is 500.
jump_beta	Numeric; proposal SD for GRM thresholds. Default is 0.4. During MCMC sampling, threshold proposals are constrained to maintain the ordering $\beta_{i,1} > \beta_{i,2} > \cdots > \beta_{i,K-1}$ for each item.
jump_theta	Numeric; proposal SD for theta. Default is 1.
jump_gamma	Numeric; proposal SD on log-scale for gamma. Default is 0.2.
jump_z	Numeric; proposal SD for z. Default is 0.5.

jump_w	Numeric; proposal SD for w. Default is 0.5.
pr_mean_beta	Numeric; prior mean for thresholds. Default is 0.
pr_sd_beta	Numeric; prior SD for thresholds. Default is 1.
pr_mean_theta	Numeric; prior mean for theta. Default is 0.
pr_sd_theta	Numeric; prior SD for theta. Default is 1.
pr_a_theta	Numeric; shape for inverse-gamma prior on var(theta). Default is 0.001.
pr_b_theta	Numeric; scale for inverse-gamma prior on var(theta). Default is 0.001.
pr_spike_mean	Numeric; prior mean for the spike component (on log-scale). Default is -3.
pr_spike_sd	Numeric; prior SD for the spike component (on log-scale). Default is 1.
pr_slab_mean	Numeric; prior mean for the slab component (on log-scale). Default is 0.5.
pr_slab_sd	Numeric; prior SD for the slab component (on log-scale). Default is 1.
pr_xi_a	Numeric; Beta prior shape a for mixing weight ξ. Default is 1.
pr_xi_b	Numeric; Beta prior shape b for mixing weight ξ. Default is 1.

adapt

List; optional adaptive MCMC control. If not NULL, proposal standard deviations are adapted during burn-in using a Robbins-Monro update on the log proposal SD and are held fixed during the main MCMC sampling. When adaptation is enabled, the reported acceptance ratios in the output (accept_beta, accept_theta, etc.) are computed only from iterations after burn-in, reflecting the performance of the adapted proposal distributions. The defaults are tuning heuristics for random-walk Metropolis proposals: scalar parameters use a target acceptance rate of 0.44, while latent position block proposals use 0.234. The 0.44 target is commonly used for one-dimensional random-walk proposals, and 0.234 is the high-dimensional optimal-scaling benchmark discussed by Roberts, Gelman and Gilks (1997) and Roberts and Rosenthal (2001). These values are proposal-tuning targets, not convergence diagnostics. Elements of the list can include:

- use_adapt: Logical; if TRUE, adaptive MCMC is used. Default is FALSE.
- adapt_interval: Integer; the number of iterations between proposal SD updates. Smaller values react more quickly but can be noisy; larger values are smoother but adapt more slowly. Default is 100.
- adapt_rate: Numeric; Robbins-Monro scaling constant c in the step-size formula c/t^{α}. Larger values adapt faster but can oscillate; smaller values are more conservative. Default is 1.0. Recommended starting range is 0.5–2.0.
- decay_rate: Numeric; Robbins-Monro decay exponent α in c/t^{α}. Values above 0.5 give diminishing adaptation consistent with the usual stochastic-approximation square-summability condition. Default is 0.6; recommended range is 0.6–0.8.
- jump_min, jump_max: Numeric; advanced lower and upper bounds for adapted proposal SDs, used to prevent proposal scales from collapsing to zero or becoming excessively large. These are normally left unchanged.
- target_accept: Numeric; default target acceptance rate for scalar random-walk updates (beta, theta, gamma). Default is 0.44.

- `target_accept_zw`: Numeric; target acceptance rate for latent position block updates z and w. Default is `0.234`.
- `target_accept_beta/theta/gamma`: Numeric; (optional) parameter-specific target acceptance rates to override `target_accept`.

`verbose` Logical; If TRUE, MCMC samples are printed for each `nprint`. Default is FALSE.

`fix_theta_sd` Logical; If TRUE, the standard deviation of the respondent latent positions θ is fixed at 1 instead of being sampled. Default is FALSE.

Examples

```
# generate example ordinal item response matrix
set.seed(123)
nsample <- 50
nitem <- 10
data <- matrix(sample(1:5, nsample * nitem, replace = TRUE), nrow = nsample)

# Fit 1PL GRM LSIRM with Spike-and-Slab
fit <- lsirmgrm_ss(data, niter = 1000, nburn = 500)
summary(fit)
```

---

onepl *1PL Rasch model.*

Description

onepl is used to fit 1PL Rasch model.

Usage

```
onepl(
  data,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001
)
```

Arguments

- `data` — Matrix; binary item response matrix to be analyzed. Each row is assumed to be respondent and its column values are assumed to be response to the corresponding item.
- `niter` — Numeric; number of iterations to run MCMC sampling. default value is 15000.
- `nburn` — Numeric; number of initial, pre-thinning, MCMC iterations to discard. default value is 2500.
- `nthin` — Numeric;number of thinning, MCMC iterations to discard. default value is 5.
- `nprint` — Numeric; MCMC samples is displayed during execution of MCMC chain for each `nprint`. default value is 500.
- `jump_beta` — Numeric; jumping rule of the proposal density for beta. default value is 0.4.
- `jump_theta` — Numeric; jumping rule of the proposal density for theta. default value is 1.0.
- `pr_mean_beta` — Numeric; mean of normal prior for beta. default value is 0.
- `pr_sd_beta` — Numeric; standard deviation of normal prior for beta. default value is 1.0.
- `pr_mean_theta` — Numeric; mean of normal prior for theta. default value is 0.
- `pr_a_theta` — Numeric; shape parameter of inverse gamma prior for variance of theta. default value is 0.001.
- `pr_b_theta` — Numeric; scale parameter of inverse gamma prior for variance of theta. default value is 0.001.

Details

`onepl` models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j:

$$logit(P(Y_{j,i} = 1|\theta_j, \beta_i)) = \theta_j + \beta_i$$

Value

`onepl` returns an object of list containing the following components:

- `beta_estimate` — posterior estimation of beta.
- `theta_estimate` — posterior estimation of theta.
- `sigma_theta_estimate` — posterior estimation of standard deviation of theta.
- `beta` — posterior samples of beta.
- `theta` — posterior samples of theta.
- `theta_sd` — posterior samples of standard deviation of theta.
- `accept_beta` — accept ratio of beta.
- `accept_theta` — accept ratio of theta.

Examples

```
## Not run:
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

result <- onepl(data)

## End(Not run)
```

---

plot *Plotting the interaction map or summarizing the parameter estimate of fitted LSIRM with box plot.*

Description

plot is used to plot the interaction map of fitted LSIRM or summarizing the parameter estimate of fitted LSIRM with box plot.

Usage

```
plot(
  object,
  ...,
  option = "interaction",
  rotation = FALSE,
  cluster = NA,
  which.clust = "item",
  interact = FALSE,
  chain.idx = 1,
  xlim = NULL,
  ylim = NULL,
  resp.size = 0.7,
  item.size = 4,
  xlab = NULL,
  ylab = NULL,
  title = NULL
)
```

Arguments

object	Object of class lsirm.
...	Additional arguments for the corresponding function.
option	Character; If value is "interaction", draw the interaction map that represents interactions between respondents and items. If value is "beta", draw the boxplot for the posterior samples of beta. If value is "theta", draw the distribution of the theta estimates per total test score for the data. If value is "alpha", draw the boxplot for the posterior samples of alpha. The "alpha" is only available for 2PL LSIRM.

`rotation`	Logical; If TRUE the latent positions are visualized after oblique (oblimin) rotation.
`cluster`	Character; clustering method for the interaction map. Use `"neyman"` for Neyman-Scott/Thomas point-process clustering or `"spectral"` for spectral clustering. If `NA`, no clustering is applied. Default is `NA`.
`which.clust`	Character; latent positions to cluster when `cluster` is `"neyman"` or `"spectral"`. Use `"item"` to cluster item positions (`w_estimate`) or `"resp"` to cluster respondent positions (`z_estimate`). Default is `"item"`.
`interact`	Logical; If TRUE, draw the interaction map interactively.
`chain.idx`	Numeric; Index of MCMC chain. Default is 1.
`xlim`	Numeric vector of length 2; x-axis range for interaction plots. Default is NULL (auto).
`ylim`	Numeric vector of length 2; y-axis range for interaction plots. Default is NULL (auto).
`resp.size`	Numeric; respondent point size for interaction plots. Default is 0.7.
`item.size`	Numeric; item text size for interaction plots. Default is 4.
`xlab`	Character; x-axis label. Default is NULL.
`ylab`	Character; y-axis label. Default is NULL.
`title`	Character; plot title. Default is NULL.

Value

`plot` returns the interaction map or boxplot for parameter estimate.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5), ncol=10, nrow=50)
lsirm_result <- lsirm(data ~ lsirm1pl())
plot(lsirm_result)

# use oblique rotation
plot(lsirm_result, rotation = TRUE)

# interaction map interactively
plot(lsirm_result, interact = TRUE)

# clustering the respondents or items
plot(lsirm_result, cluster = "spectral", which.clust = "item")
plot(lsirm_result, cluster = "spectral", which.clust = "resp")
```

print.summary.lsirm *Print the summary the result of LSIRM*

Description

print.summary.lsirm is used to print summary the result of LSIRM.

Usage

```
## S3 method for class 'summary.lsirm'
print(x, items = NULL, respondents = NULL, which = NULL, ...)
```

Arguments

x	List; summary of LSIRM with `summary.lsirm`.
items	Numeric or Character vector; the items to display in the parameter summaries. Either a vector of item indices (e.g. `1:5`) or item names (e.g. `c("item 1", "item 3")`). Default is `NULL` (displays all items).
respondents	Numeric or Character vector; the respondents to display in the respondent summaries (theta). Either a vector of respondent indices (e.g. `1:10`) or respondent names (e.g. `c("respondent 1", "respondent 2")`). Default is `NULL` (displays all respondents if theta is requested).
which	Character vector; specifies which parameters to print in the console. Options are `"beta"` (item difficulty/threshold), `"alpha"` (item discrimination), and `"theta"` (respondent ability). Or set to `"all"` to print all. Default is `NULL` (dynamically prints `"beta"` for 1PL models, and both `"beta"` and `"alpha"` for 2PL models).
...	Additional arguments.

Value

`print.summary.lsirm` return a summary of LSIRM.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)
lsirm_result <- lsirm(data ~ lsirm1pl())
summary(lsirm_result)
```

summary.lsirm *Summary the result of LSIRM*

Description

summary is used to summary the result of LSIRM.

Usage

```
## S3 method for class 'lsirm'
summary(
  object,
  chain.idx = 1,
  estimate = "mean",
  CI = 0.95,
  items = NULL,
  respondents = NULL,
  which = NULL,
  ...
)
```

Arguments

object	Object of class lsirm.
chain.idx	Numeric; Index of MCMC chain. Default is 1.
estimate	Character; Specifies the type of posterior estimate to provide for beta parameters. Options are "mean", "median", or "mode". Default is "mean".
CI	Numeric; The significance level for the highest posterior density interval (HPD) for the beta parameters. Default is 0.95.
items	Numeric or Character vector; the items to display in the parameter summaries. Either a vector of item indices (e.g. 1:5) or item names (e.g. c("item 1", "item 3")). Default is NULL (displays all items).
respondents	Numeric or Character vector; the respondents to display in the respondent summaries (theta). Either a vector of respondent indices (e.g. 1:10) or respondent names (e.g. c("respondent 1", "respondent 2")). Default is NULL (displays all respondents if theta is requested).
which	Character vector; specifies which parameters to print in the console. Options are "beta" (item difficulty/threshold), "alpha" (item discrimination), and "theta" (respondent ability). Or set to "all" to print all. Default is NULL (dynamically prints "beta" for 1PL models, and both "beta" and "alpha" for 2PL models).
...	Additional arguments.

Value

summary.lsirm contains following elements. A print method is available.

call	R call used to fit the model.
coef	Covariate coefficients posterior means.
mcmc.opt	The number of mcmc iteration, burn-in periods, and thinning intervals.
map.inf	Value of log maximum a posterior and iteration number which have log maximum a posterior.
BIC	Numeric value with the corresponding Bayesian information criterion (BIC).
method	Which model is fitted.
missing	The assumed missing type. One of NA, "mar" and "mcar".
dtype	Type of input data (Binary or Continuous).
ss	Whether a model selection approach using the spike-slab prior is applied.

Examples

```
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

# 1PL LSIRM object
lsirm_result <- lsirm(data ~ lsirm1pl())
summary(lsirm_result)
```

TDRI *Inductive Reasoning Developmental Test*

Description

TDRI dataset is the answer to Inductive Reasoning Developmental Test of 1,803 Brazilians with age varying from 5 to 85 years.

Usage

```
data(TDRI)
```

Format

A binary matrix with 1,803 rows and 56 columns.

Details

It presents data from 1,803 Brazilians (52.5% female) with age varying from 5 to 85 years (M = 15.75; SD = 12.21) that answered to the Inductive Reasoning Developmental Test – IRDT, with 56 items designed to assess developmentally sequenced and hierarchically organized inductive reasoning.

Source

https://figshare.com/articles/dataset/TDRI_dataset_csv/3142321

`twopl` *2PL Rasch model.*

Description

twopl is used to fit 2PL Rasch model. Unlike 1PL model, 2PL model assumes the item effect can vary according to respondent, allowing additional parameter multiplied with respondent effect.

Usage

```
twopl(
  data,
  niter = 15000,
  nburn = 2500,
  nthin = 5,
  nprint = 500,
  jump_beta = 0.4,
  jump_theta = 1,
  jump_alpha = 1,
  pr_mean_beta = 0,
  pr_sd_beta = 1,
  pr_mean_theta = 0,
  pr_mean_alpha = 0.5,
  pr_sd_alpha = 1,
  pr_a_theta = 0.001,
  pr_b_theta = 0.001
)
```

Arguments

`data`	Matrix; binary item response matrix to be analyzed. Each row is assumed to be respondent and its column values are assumed to be response to the corresponding item.
`niter`	Numeric; number of iterations to run MCMC sampling. default value is 15000.
`nburn`	Numeric; number of initial, pre-thinning, MCMC iterations to discard. default value is 2500.
`nthin`	Numeric;number of thinning, MCMC iterations to discard. default value is 5.
`nprint`	Numeric; MCMC samples is displayed during execution of MCMC chain for each `nprint`. default value is 500.
`jump_beta`	Numeric; jumping rule of the proposal density for beta. default value is 0.4.
`jump_theta`	Numeric; jumping rule of the proposal density for theta. default value is 1.0.
`jump_alpha`	Numeric; jumping rule of the proposal density for alpha default value is 1.0.

pr_mean_beta	Numeric; mean of normal prior for beta. default value is 0.
pr_sd_beta	Numeric; standard deviation of normal prior for beta. default value is 1.0.
pr_mean_theta	Numeric; mean of normal prior for theta. default value is 0.
pr_mean_alpha	Numeric; mean of normal prior for alpha. default value is 0.5.
pr_sd_alpha	Numeric; mean of normal prior for beta. default value is 1.0.
pr_a_theta	Numeric; shape parameter of inverse gamma prior for variance of theta. default value is 0.001.
pr_b_theta	Numeric; scale parameter of inverse gamma prior for variance of theta. default value is 0.001.

Details

twopl models the probability of correct response by respondent j to item i with item effect β_i, respondent effect θ_j. For 2pl model, the the item effect is assumed to have additional discrimination parameter α_i multiplied by θ_j:

$$logit(P(Y_{j,i} = 1|\theta_j, \beta_i, \alpha_i)) = \theta_j * \alpha_i + \beta_i$$

Value

twopl returns an object of list containing the following components:

beta_estimate	posterior estimation of beta.
theta_estimate	posterior estimation of theta.
sigma_theta_estimate	posterior estimation of standard deviation of theta.
alpha_estimate	posterior estimation of alpha.
beta	posterior samples of beta.
theta	posterior samples of theta.
theta_sd	posterior samples of standard deviation of theta.
alpha	posterior samples of alpha.
accept_beta	accept ratio of beta.
accept_theta	accept ratio of theta.
accept_alpha	accept ratio of alpha.

Examples

```
## Not run:
# generate example item response matrix
data     <- matrix(rbinom(500, size = 1, prob = 0.5),ncol=10,nrow=50)

result <- twopl(data)

## End(Not run)
```

Index