Package ‘jafar’

September 30, 2025

Type Package

Title Bayesian Joint Additive Factor Regression for Multi-View Learning

Version 0.1.0

Description The package implements two supervised Bayesian factor models for multi-view data integration. The baseline Joint Factor Regression (jfr) model captures the combined variation across multiple data views using a single set of latent factors. A more refined Joint Additive FActor Regression (jafar) model explicitly decomposes variation into shared and view-specific components. Both models leverage extensions of the cumulative shrinkage process prior, learning adaptively the number of factors in a fully Bayesian way.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 3.5)

Imports Rcpp,
RcppArmadillo,
RcppParallel,
truncnorm,
Hmisc,
stats,
caret,
fields,
ggplot2,
cowplot,
latex2exp,
knitr,
svMisc,
MASS

LinkingTo Rcpp,
RcppArmadillo,
RcppParallel

SystemRequirements C++17

Contents

`gibbs_jafar` *Gibbs sampler for* `jafar`

Description

Fits a Bayesian Joint Additive FActor Regression (`jafar`) model using Gibbs sampling. Variation across multiple data-views is explained via shared and view-specific latent factors. The model can be fitted in both unsupervised and supervised settings. Default and optional outputs include posterior means of the induced covariances, posterior samples of residual variances, latent factors, and factor loadings. Supports parallel computation and tempered loading updates to limit rank estimation in extreme large-p-small-n settings.

Usage

```
gibbs_jafar(
  X_m,
  y = NULL,
  yBinary = F,
  K0 = NULL,
  K0_m = NULL,
  tMCMC = 20000,
  tBurnIn = 15000,
  tThin = 10,
  hyperparams = list(),
  get_latent_vars = TRUE,
  get_last_sample = FALSE,
  parallel = TRUE,
  tempered = FALSE,
  rescale_pred = FALSE
)
```

Arguments

- **X_m**: Multi-view input data, pre-processed via preprocess_X. List of length M; m-th element: matrix n x p_m[m]. Rows should correspond to samples, columns to features.
- **y**: Vector of responses (length n) pre-processed via preprocess_y. Set to NULL for unsupervised mode (default: NULL).
- **yBinary**: Logical, indicating if the response(s) are binary (default: FALSE).
- **K0**: Upper bound to numbers of shared latent factors (optional) If NULL, K0 is set to 3*log(max(p_m))
- **K0_m**: Upper bounds to numbers of view-specific latent factors (optional) Length should equal length(X_m). If NULL, K0[m] is set to 3*log(max_(p_m[m]))
- **tMCMC**: Total number of MCMC iterations (default: 20000).
- **tBurnIn**: Number of burn-in iterations (default: 15000).
- **tThin**: Thinning interval for saving samples (default: 10).
- **hyperparams**: List of hyperparameters for the D-CUSP prior distributions. Missing hyperparameters are replaced by default values encoded in set_hyperparameters.
- **get_latent_vars**: Return latent factors and loading matrices (logical, default: TRUE).
- **get_last_sample**: Return the last sample of the MCMC chain (logical, default: FALSE).
- **parallel**: Use parallel computation for the loadings update (logical, default: TRUE).
- **tempered**: Use tempered full-conditional for the loadings matrices (logical, default: FALSE).
- **rescale_pred**: Rescale loadings when computing response predictions (logical, default: FALSE).

Details

- Ensure that all matrices in X_m have the same number of rows (subjects).
- Missing data in X_m are allowed as NA and imputed in the MCMC.

Value

A list containing posterior samples, latent variables (if requested), and other relevant model outputs.

Note

All posterior samples are reported only after burn-in, except for K and K_Gm. The number of samples after thinning is tFull=tMCMC%/%tThin and tEff=(tMCMC-tBurnIn)%/%tThin for the full chain and post burn-in, respectively.
The output list includes:

- K: Number shared latent factors (vector of length tFull).
- K_Gm: Number view-specific latent factors (matrix tFull x M).
- K_Lm_eff: Numbers of shared factors active in each view (matrix tEff x M).
- K_Gm_eff: Numbers of specific factors active in each view (matrix tEff x M).
- active_Lm: Binary indicators of shared factors activity across views (binary array tEff x K x Ms).
- Cov_m_mean: Posterior mean of the covariance matrix for each dataset (list of length M; m-th element: matrix p_m[m] x p_m[m]).

- Marg_Var_m: Marginal variances of features (list of length M; m-th element: matrix tEff x p_m[m]).
- s2_inv_m: Inverse residual variances across views (list of length M; m-th element: matrix tEff x p_m[m]).
- mu_m: Features intercepts across views (list of length M; m-th element: matrix tEff x p_m[m]).
- hyper_param: List of hyperparameters used for the model, including user-specified values and defaults ones were missing.

If is_supervised = TRUE:

- K_T_eff: Numbers of shared factors active in the response (vector of length tEff).
- K_Tm_eff: Numbers of specific factors active in the response (matrix tEff x M).
- active_T: Binary indicators of shared factors activity in the response (binary matrix tEff x K).
- active_Tm: Binary indicators of specific factors activity in the response (list of length M; m-th element: matrix tEff x K_Gm[m]).
- s2_inv: Response inverse residual variances (vector of length tEff).
- mu_y: Response intercept (vector of length tEff).
- Theta: Response loadings on shared factors (matrix tEff x K).
- Theta_m: Response loadings on specific factors (list of length M; m-th element: matrix tEff x K_Gm[m]).
- y_MC: Latent probit utilities (matrix tEff x n). (only if yBinary = TRUE).

If get_latent_vars = TRUE:

- Lambda_m: Loadings matrices on shared factors (list of length M; m-th element: array tEff x p_m[m] x K).
- Gamma_m: Loadings matrices on view-specific factors (list of length M; m-th element: array tEff x p_m[m] x K_Gm[m]).
- eta: Shared latent factors (array tEff x n x K).
- phi_m: View-specific latent factors (list of length M; m-th element: array tEff x n x K_Gm[m]).

If the input matrices X_m contain missing values:

- Xm_MC: Posterior samples of imputed values for missing entries. A list of length M; the m-th element is itself a list (one per feature with missingness), each containing an tEff × n_miss matrix of imputed values across MCMC iterations.
- na_idx: List of length M; the m-th element gives the column indices of missing entries in X_m[[m]].
- na_row_idx: List of length M; the m-th element gives the corresponding row indices of missing entries in X_m[[m]].

If get_last_sample = TRUE:

- last_sample: List of posterior values of all parameters at the last MCMC iteration, including latent factors, loadings, residual variances, and hyperparameters.

References

Anceschi N., Ferrari F., Dunson D. B., & Mallick H. (2025). *Bayesian Joint Additive Factor Models for Multiview Learning*. https://arxiv.org/abs/2406.00778

Legramanti S., Durante D., & Dunson D. B. (2020). *Bayesian cumulative shrinkage for infinite factorizations*. Biometrika, 107(3), 745-752. https://doi.org/10.1093/biomet/asaa008

gibbs_jfr *Gibbs sampler for* jfr

Description

Fits a Bayesian Joint Factor Regression (jfr) model using Gibbs sampling. Variation across multiple data views is explained by a single set of global latent factors. The model can be fitted in both unsupervised and supervised settings. Default and optional outputs include posterior means of the induced covariances, posterior samples of residual variances, latent factors, and factor loadings. Supports parallel computation and tempered loading updates to limit rank estimation in extreme large-p-small-n settings.

Usage

```
gibbs_jfr(
  X_m,
  y = NULL,
  yBinary = F,
  K0 = NULL,
  tMCMC = 20000,
  tBurnIn = 15000,
  tThin = 10,
  hyperparams = list(),
  get_latent_vars = TRUE,
  get_last_sample = FALSE,
  parallel = TRUE,
  tempered = FALSE,
  rescale_pred = FALSE
)
```

Arguments

X_m	Multi-view input data, pre-processed via preprocess_X. List of length M; m-th element: matrix n x p_m[m]. Rows should correspond to samples, columns to features.
y	Vector of responses (length n) pre-processed via preprocess_y. Set to NULL for unsupervised mode (default: NULL).
yBinary	Logical, indicating if the response(s) are binary (default: FALSE).
K0	Upper bound to numbers of latent factors (optional) If NULL, K0 is set to 3*log(max(p_m))
tMCMC	Total number of MCMC iterations (default: 20000).
tBurnIn	Number of burn-in iterations (default: 15000).
tThin	Thinning interval for saving samples (default: 10).
hyperparams	List of hyperparameters for the I-CUSP prior distributions. Missing hyperparameters are replaced by default values encoded in set_hyperparameters.
get_latent_vars	Return latent factors and loading matrices (logical, default: TRUE).
get_last_sample	Return the last sample of the MCMC chain (logical, default: FALSE).

parallel	Use parallel computation for the loadings update (logical, default: TRUE).
tempered	Use tempered full-conditional for the loadings matrices (logical, default: FALSE).
rescale_pred	Rescale loadings when computing response predictions (logical, default: FALSE).

Details

- Ensure that all matrices in X_m have the same number of rows (subjects).
- Missing data in X_m are allowed as NA and imputed in the MCMC.

Value

A list containing posterior samples, latent variables (if requested), and other relevant model outputs.

Note

All posterior samples are reported only after burn-in, except for K. The number of samples after thinning is tFull=tMCMC%/%tThin and tEff=(tMCMC-tBurnIn)%/%tThin for the full chain and post burn-in, respectively.
The output list includes:

- K: Number shared latent factors (vector of length tFull).
- K_Lm_eff: Numbers of latent factors active in each view (matrix tEff x M).
- active_Lm: Binary indicators of latent factors activity across views (binary array tEff x K x M).
- Cov_m_mean: Posterior mean of the covariance matrix for each dataset (list of length M; m-th element: matrix p_m[m] x p_m[m]).
- Marg_Var_m: Marginal variances of features (list of length M; m-th element: matrix tEff x p_m[m]).
- s2_inv_m: Inverse residual variances across views (list of length M; m-th element: matrix tEff x p_m[m]).
- mu_m: Features intercepts across views (list of length M; m-th element: matrix tEff x p_m[m]).
- hyper_param: List of hyperparameters used for the model, including user-specified values and defaults ones were missing.

If is_supervised = TRUE:

- K_T_eff: Numbers of latent factors active in the response (vector of length tEff).
- active_T: Binary indicators of latent factors activity in the response (binary matrix tEff x K).
- s2_inv: Response inverse residual variances (vector of length tEff).
- mu_y: Response intercept (vector of length tEff).
- Theta: Response loadings on latent factors (matrix tEff x K).
- y_MC: Latent probit utilities (matrix tEff x n). (only if yBinary = TRUE).

If get_latent_vars = TRUE:

- Lambda_m: Loadings matrices on latent factors (list of length M; m-th element: array tEff x p_m[m] x K).
- eta: Latent factors (array tEff x n x K).

If the input matrices `X_m` contain missing values:

- `Xm_MC`: Posterior samples of imputed values for missing entries. A list of length `M`; the m-th element is itself a list (one per feature with missingness), each containing an `tEff` × `n_miss` matrix of imputed values across MCMC iterations.
- `na_idx`: List of length `M`; the m-th element gives the column indices of missing entries in `X_m[[m]]`.
- `na_row_idx`: List of length `M`; the m-th element gives the corresponding row indices of missing entries in `X_m[[m]]`.

If `get_last_sample = TRUE`:

- `last_sample`: List of posterior values of all parameters at the last MCMC iteration, including latent factors, loadings, residual variances, and hyperparameters.

References

Anceschi N., Ferrari F., Dunson D. B., & Mallick H. (2025). *Bayesian Joint Additive Factor Models for Multiview Learning*. https://arxiv.org/abs/2406.00778

Legramanti S., Durante D., & Dunson D. B. (2020). *Bayesian cumulative shrinkage for infinite factorizations*. Biometrika, 107(3), 745-752. https://doi.org/10.1093/biomet/asaa008

`multiviewMatchAlign` *Rotational alignment of latent factors and loading matrices*

Description

Post-processing routine to solve rotational ambiguity across MCMC samples of latent variables. The alignment is performed using multi-view `MatchAlign` on the shared component and regular `MatchAlign` on the specific ones.

Usage

```
multiviewMatchAlign(risMCMC)
```

Arguments

`risMCMC`	Posterior samples, as returned by `gibbs_jafar` or `gibbs_jfr`.

Value

A modified version of the input `risMCMC`, with latent factors, loading matrices, and response loadings (if supervised) rotated according to multi-view `MatchAlign`.

References

Anceschi N., Ferrari F., Dunson D. B., & Mallick H. (2025). *Bayesian Joint Additive Factor Models for Multiview Learning*. https://arxiv.org/abs/2406.00778

Poworoznek E., Anceschi N., Ferrari F., & Dunson D. B. (2025). *Efficiently Resolving Rotational Ambiguity in Bayesian Matrix Sampling with Matching*. Bayesian Analysis, 1–22. https://doi.org/10.1214/25-BA1544

`plot_coefficients` *Visualization of regression coefficients*

Description

Plot induced regression coefficients, directly relating the response `y` to the observed multi-view predictors. The corresponding representation is obtained by marginalizing out all latent factors.

Usage

```
plot_coefficients(yPred, out_path = "~/Desktop/", out_name = "coefficients")
```

Arguments

`yPred`	Response predictions, output of `predict_y` or `predict_y_raw`
`out_path`	Output path where the generated plot will be saved (default: "~/Desktop/")
`out_name`	Output file name (default: "coefficients")

`plot_correlations` *Visualization of induced correlation matrices*

Description

Plot the empirical and inferred within-view correlation matrices. The induced correlations on `X_m` are obtained by marginalizing out all latent factors.

Usage

```
plot_correlations(
  risMCMC,
  X_m = NULL,
  out_path = "~/Desktop/",
  out_name = "correlations"
)
```

Arguments

`risMCMC`	Posterior samples, output of `gibbs_jafar` or `gibbs_jfr`
`X_m`	Training set multi-view predictors (optional, default: `NULL`). If `NULL`, only inferred correlation matrices are visualized. If not `NULL`, the empirical correlation matrices are displayed besides the inferred ones
`out_path`	Output path where the generated plot will be saved (default: "~/Desktop/")
`out_name`	Output file name (default: "correlations")

`plot_loadings` *Visualization of loadings matrices*

Description

Plot posterior means of the loading matrices, including the response loadings in the supervised case. Rotational alignment must be performed in advance. To this end, make sure to provide in input the output of `multiviewMatchAlign`.

Usage

```
plot_loadings(
  risMCMC,
  out_path = "~/Desktop/",
  out_shared = "shared_loadings",
  out_specific = "specific_loadings"
)
```

Arguments

`risMCMC`	Postprocesed posterior samples, output of `multiviewMatchAlign`
`out_path`	Output path where the generated plot will be saved
`out_shared`	File name for the shared component plot (default: "n_factors_shared")
`out_specific`	File name for the specific components plot (default: "n_factor_specific")

`plot_n_factors` *Visualization of inferred ranks*

Description

Plot MCMC samples of the inferred number of factors.

Usage

```
plot_n_factors(
  risMCMC,
  out_path = "~/Desktop",
  out_shared = "n_factors_shared",
  out_specific = "n_factor_specific"
)
```

Arguments

`risMCMC`	Posterior samples, output of `gibbs_jafar` or `gibbs_jfr`
`out_path`	Output path where the generated plot will be saved
`out_shared`	File name for the shared component plot (default: "n_factors_shared")
`out_specific`	File name for the specific components plot (default: "n_factor_specific")

plot_predictions *Visualization of predicted responses*

Description

Plot response predictions against true values.

Usage

```
plot_predictions(
  yPred,
  yTrue,
  risMCMC,
  out_path = "~/Desktop/",
  out_name = "predictions"
)
```

Arguments

yPred	Response predictions, output of `predict_y` or `predict_y_raw`
yTrue	True values of the responses
risMCMC	Posterior samples, output of `gibbs_jafar` or `gibbs_jfr`
out_path	Output path where the generated plot will be saved (default: "~/Desktop/")
out_name	Output file name (default: "predictions")

predict.bsfp.oos *Out-of-sample prediction for* `bsfp`

Description

Modified version of the function `bsfp.predict` from the GitHub repo `bsfp` for out-of-sample predictions. Response predictions are generated by first sampling latent factors. The original version `bsfp.predict` includes the response in the conditioning set of such latent factors. `predict.bsfp.oos` allows proper out-of-sample prediction by excluding the response from such conditioning set.

Usage

```
## S3 method for class 'bsfp.oos'
predict(
  bsfp.fit,
  test_data,
  response_type = "continuous",
  model_params = NULL,
  nsample,
  progress = TRUE,
  starting_values = NULL
)
```

Arguments

- `bsfp.fit` — Results from fitting `bsfp` on training data.
- `test_data` — Matrix-list dataset of held-out test data.
- `response_type` — Continuous or binary response. Must be one of 'continuous' (deafult) or 'binary'.
- `model_params` — May be `NULL` if `model_params=NULL` in `bsfp` fit. Otherwise, specify as `(error_vars, joint_vars, indiv_vars, beta_vars, response_vars)`.
- `nsample` — Integer specifing number of Gibbs sampling iterations
- `progress` — Boolean determining if progress of the sampler be displayed
- `starting_values` — List of starting values for $\mathbf{V}, \mathbf{U}_s, \mathbf{W}_s, \mathbf{V}_s$ for $s = 1, \ldots, q$. If `NULL`, initialize from prior.

Details

Generate new scores for held-out test data based on a training fit of `bsfp`. Uses the estimated ranks and joint and individual loadings. Cannot be used if missing values are present in test data.

Value

Returns a list with the following parameters:

- `test_data` — Test data provided by user
- `EY.draw` — List of posterior samples for the E(Y|X), i.e. $\beta_0 + \mathbf{V}\boldsymbol{\beta}_{joint} + \sum_{s=1}^{q} \mathbf{V}_s\boldsymbol{\beta}_s$ for each Gibbs sampling iteration.
- `V.draw` — List of posterior samples for joint scores, $\mathbf{V}$
- `U.train` — List of posterior samples for joint loadings for each source, $\mathbf{U}_s$ for $s = 1, \ldots, q$ given by the training `bsfp` fit
- `W.train` — List of posterior samples for individual loadings for each source, $\mathbf{W}_s$ for $s = 1, \ldots, q$ given by the training `bsfp` fit
- `Vs.draw` — List of posterior samples for individual scores for each source, $\mathbf{V}_s$ for $s = 1, \ldots, q$
- `ranks` — Vector with the estimated joint and individual ranks. `ranks[1]` is the estimated joint rank. `ranks[2:(q+1)]` correspond to the individual ranks for each source.
- `tau2.train` — List of posterior samples for the response variance if the response was continuous given by training `bsfp` fit
- `beta.train` — List of posterior samples for the regression coefficients used in the predictive model given by training `bsfp` fit
- `Xm.draw` — List of posterior samples for missing predictors imputations

predict_y — *Response predictions for* jafar *and* jfr

Description

Compute induced regression coefficients and predicted responses either in-sample or out-of-sample.

Usage

```
predict_y(Xpred, risMCMC, rescale_pred = FALSE)
```

Arguments

Xpred	A list of M features matrices, the m-th one of dimension nPred x p_m[m] or possibly missing (i.e. X_m[[m]]=NULL).
risMCMC	Output of gibbs_jafar or gibbs_jfr containing posterior samples.
rescale_pred	Rescale loadings when computing response predictions (logical, default: FALSE).

Value

A list containing posterior samples of the predicted responses (matrix tEff x nPred), and of the induced regression coefficients for each view (list of length M; m-th element: tEff x p_m[m]).

predict_y_raw — *Response predictions for* jafar *and* jfr

Description

Compute predicted responses, either in-sample or out-of-sample.

Usage

```
predict_y_raw(Xpred, risMCMC, rescale_pred = FALSE)
```

Arguments

Xpred	A list of M features matrices, the m-th one of dimension nPred x p_m[m] or possibly missing (i.e. X_m[[m]]=NULL).
risMCMC	Output of gibbs_jafar or gibbs_jfr containing posterior samples.
rescale_pred	Rescale loadings when computing response predictions (logical, default: FALSE).

Value

A list containing posterior samples of the predicted responses (matrix tEff x nPred).

preprocess_X *Multi-view data preprocessing*

Description

Center and rescale, and apply feature-wise cdf transform (optional). Multi-view data for out-of-sample observations can optionally be provided as input. If so, the corresponding features are rescaled coherently with the training set.

Usage

```
preprocess_X(X_m, X_m_test = NULL, copula = FALSE)
```

Arguments

X_m	Multi-view data for the training set. List of length M; m-th element: matrix n x p_m[m].
X_m_test	Multi-view data for the test set. List of length M; m-th element: matrix nTest x p_m[m]. (optional, default: NULL)
copula	Apply cdf transformation (logical, default: FALSE)

Value

List of pre-processed features and rescaling factors

Note

The output list contains:

- preprocess_X_m: List of the rescaling transformations applied to the response, as returned by preProcess from the caret package. This object can be used to back-transform the data to the original scale.
- X_m: Rescaled multi-view features for the training set.
- X_m_test: Rescaled multi-view features for the test set (if provided).

preprocess_y *Response preprocessing*

Description

Center and rescale. Response for out-of-sample observations can optionally be provided as input. If so, the corresponding values are rescaled coherently with the training set.

Usage

```
preprocess_y(yTrain, yTest = NULL)
```

Arguments

yTrain	Train set responses
yTest	Test set responses (optional, default, NULL)

Value

List of pre-processed responses and rescaling transformation

Note

The output list contains:

- preprocess_y: The rescaling transformation applied to the response, as returned by preProcess from the caret package. This object can be used to back-transform the data to the original scale.
- yTrain: Rescaled responses for the training set.
- yTest: Rescaled responses for the test set (if provided).

---

reorder_features *Optional pre-process of the multi-view data*

Description

Reorder features via hierarchical clustering for better visualization. Multi-view data for out-of-sample observations can optionally be provided as input. If so, the corresponding features are re-ordered coherently with the training set.

Usage

```
reorder_features(X_m, X_m_test = NULL, K0_HC = 15)
```

Arguments

X_m	Multi-view data for the training set. List of length M; m-th element: matrix n x p_m[m].
X_m_test	Multi-view data for the test set. List of length M; m-th element: matrix nTest x p_m[m].
K0_HC	Reference number of clusters for hierarchical clustering (default: 15)

Value

List of pre-processed features and rescaling transformations

Note

The output list contains:

- idx_sort_HC_m: List of features reordering obtained by running hierarchical clustering on empirical correlations. This object can be used to restore the data to its original order.
- X_m: Reordered multi-view features for the training set.
- X_m_test: Reordered multi-view features for the test set (if provided).

set_hyperparameters *Set the hyperparameters for* jafar *and* jfr

Description

Helper function to set hyperparameters for gibbs_jfr and gibbs_jafar. Supports both unsupervised and supervised settings.

Usage

```
set_hyperparameters(hyperparams_list, M, is_supervised = FALSE)
```

Arguments

hyperparams_list	Named list of model hyperparameters.
M	Integer, number of data-views.
is_supervised	Running supervised model (logical, default: FALSE).

Details

Missing hyperparameters are assigned default values.

Value

A named list of hyperparameters with defaults filled in where missing. Scalar values are replicated M times where necessary.

Note

Default hyperparameters include:

- seed: random seed for reproducibility (default: 123).
- t0, t1, t0_adapt: adaptation parameters for MCMC (default: t0=-1, t1=-5e-4, t0_adapt=200).
- a_m, b_m: shape and rate of inverse-gamma prior for idiosyncratic noise in each view. Scalars of vectors of length M (default: a_m[m]=3, b_m[m]=1).
- prec0m: precision of normal prior on intercepts. Scalar of vector of length M (default: prec0m[m]=2).
- var_spike: variance of normal spike in cusps. Scalar of vector of length M (default: var_spike[m]=0.005).
- a_chi, b_chi: hyperparameters for slab inverse-gamma prior in cusps. Scalars of vectors of length M (default: a_chi[m]=0.5, b_chi[m]=0.1).
- alpha_L, alpha_G: Dirichlet process concentration parameters giving the expected number of factors, shared and local. Scalars of vectors of length M (default: alpha_L[m]=5, alpha_G[m]=5).

If is_supervised = TRUE, additional hyperparameters for the response model are

- a_sig, b_sig: shape and rate of inverse-gamma prior for idiosyncratic noise (default: a_sig=3, b_sig=1).
- prec0: precision of normal prior on intercept (default: prec0=2).
- var_spike_y: variance of normal spike (default: var_spike_y=0.005).

- `a_theta`, `b_theta`: hyperparameters for slab inverse-gamma prior in the slab (default: `a_theta=0.5`, `b_theta=0.1`).
- `a_xi`, `b_xi`: shape parameters for beta prior on mixture weight in response loadings (default: `a_xi=3`, `b_xi=2`).

Index