# Package ‘VEMIRT’

April 9, 2024

**Type** Package

**Title** Variational Expectation Maximization for High-dimensional IRT Models

**Version** 1.3

**Date** 2024-04-09

**Maintainer** Jiaying Xiao <jxiao6@uw.edu>

**Description** VEMIRT is created to assist researchers in conducting exploratory and confirmatory multidimensional item response theory (MIRT) analysis and corresponding differential item functioning (DIF) analysis. The core computation engine of VEMIRT is a family of Gaussian Variational EM algorithms that are considerably more efficient than currently available algorithms in other software packages, especially when the number of latent factors exceeds four.

**License** GPL-3

**Imports** abind,
GPArotation,
MASS,
Matrix,
mirt,
polycor,
psych,
Rcpp,
RcppArmadillo,
testit,
tibble,
torch

**LinkingTo** Rcpp, RcppArmadillo, RcppEigen

**Encoding** UTF-8

**Depends** R (>= 2.10)

**LazyData** true

**RoxygenNote** 7.3.1

## R topics documented:

---

`VEMIRT-package` *VEMIRT: A package for high-dimensional IRT models*

---

### Description

VEMIRT is created to assist researchers to conduct exploratory and confirmatory multidimensional item response theory (MIRT) analysis and cooresponding item differential functioning (DIF) analysis. The core computation engine of VEMIRT is a family of Gaussian Variational EM algorithms that are considerably more efficient than currently available algorithms in other software packages, especially when the number of latent factors exceeds four.

### Identifying the number of factors

`pa_poly` identifies the number of factors via parallel analysis.

### Exploratory factor analysis

- `gvem_2PLEFA_rot` conducts M2PL Analysis with post-hoc rotation (Promax & CF-Quartimax)
- `gvem_2PLEFA_lasso` conducts M2PL Analysis with Lasso penalty
- `gvem_2PLEFA_adaptlasso` conducts M2PL Analysis with adaptive Lasso penalty
- `sgvem_3PLEFA_rot` conducts stochastic GVEM to futher imporve the computational effficiency for exploratory M3PL analysis
- `sgvem_3PLEFA_lasso` conducts M3PL Analysis with Lasso penalty
- `sgvem_3PLEFA_adaptlasso` conducts M3PL Analysis with adaptive Lasso penalty

### Confirmatory factor analysis

- gvem_2PLCFA conducts GVEM for confirmatory M2PL analysis
- sgvem_3PLCFA conducts stochastic GVEM for confirmatory M3PL analysis
- bs_2PLCFA conducts bootstrap sampling to correct bias and produce standard errors for confirmatory M2PL analysis
- importanceSampling conducts importance sampling to correct bias for M2PL analysis

### Differential item functioning analysis

- em_DIF conducts DIF analysis for M2PL models using EM algorithms
- gvemm_DIF conducts DIF analysis for M2PL models using GVEMM algorithms
- lrt_DIF conducts DIF analysis for M2PL models using the likelihood ratio test

### Author(s)

**Maintainer**: Jiaying Xiao <jxiao6@uw.edu> (ORCID)

Authors:

- Weicong Lyu <wlyu4@uw.edu> (ORCID)
- Ruoyi Zhu <zhux0445@uw.edu> (ORCID)
- Gongjun Xu <gongjun@umich.edu> (ORCID)
- Chun Wang <wang4066@uw.edu> (ORCID)

---

## bs_2PLCFA *Bootstrap Version of GVEM Confirmatory Analysis for M2PL*

---

### Description

A bootstrap version of GVEM (i.e., GVEM-BS) can be implemented to correct the bias on item parameters and compute standard errors under M2PL models

### Usage

```
bs_2PLCFA(gvem_result, boots = 5)
```

### Arguments

| | |
|---|---|
| gvem_result | a list that includes exploratory or confirmatory GVEM results for M2PL models. |
| boots | the number of bootstrap samples; default is 5 |

### Value

a list containing the following objects:

| | |
|---|---|
| boots_a | item discrimination parameters corrected by bootstrap sampling, a $J \times K$ matrix |
| boots_b | item difficulty parameters corrected by bootstrap sampling, a vector of length $J$ |
| sd_a | stardard errors of item discrimination parameters, a $J \times K$ matrix |
| sd_b | stardard errors of item difficulty parameters, a vector of length $J$ |

### See Also

gvem_2PLCFA,importanceSampling

### Examples

```
## Not run:
gvem_result <- gvem_2PLCFA(exampleData_2pl, exampleIndic_cfa2pl)
bs_2PLCFA(gvem_result, boots=10)
## End(Not run)
```

---

## coef.vemirt_DIF *Extract Parameter Estimates from DIF Analysis*

---

### Description

Extract Parameter Estimates from DIF Analysis

### Usage

```
coef(object, criterion = NULL)
```

### Arguments

| | |
|---|---|
| object | An object of class vemirt_DIF |
| criterion | Information criterion for model selection, one of 'AIC', 'BIC' and 'GIC', otherwise use the criterion specified when fitting the model(s) |

### See Also

em_DIF, gvemm_DIF, lrt_DIF, print.vemirt_DIF

---

## em_DIF *EM Algorithms for DIF Detection in 2PL Models*

---

### Description

EM Algorithms for DIF Detection in 2PL Models

### Usage

```
em_DIF(
  Y,
  D,
  X,
  method = "EMM",
  unif = F,
  Lambda0 = seq(0.2, 0.7, by = 0.1),
  criterion = "BIC",
  iter = 1000,
  eps = 0.001,
```

```
  c = 0.7,
  eta = 1
)
```

## Arguments

| | |
|---|---|
| `Y` | An $N \times J$ binary matrix of item responses |
| `D` | A $J \times K$ binary matrix of loading indicators |
| `X` | An $N$ dimensional vector of group indicators (integers from `1` to `G`) |
| `method` | Estimation algorithm, one of `'EM'`, `'EMM'` and `'Adapt'` |
| `unif` | Whether to detect uniform DIF only |
| `Lambda0` | A vector of `lambda0` values for $L_1$ penalty (`lambda` is `sqrt(N) * lambda0`) |
| `criterion` | Information criterion for model selection, one of `'BIC'` (recommended), `'AIC'` and `'GIC'` |
| `iter` | Maximum number of iterations |
| `eps` | Termination criterion on numerical accuracy |
| `c` | Constant for computing GIC |
| `eta` | Tuning constant for adaptive lasso (`'Adapt'` only) |

## Value

An object of class `vemirt_DIF`, which is a list containing three elements:

| | |
|---|---|
| `fit` | The best (with lowest information criterion) model, which is an element of `all` |
| `best` | The location of `fit` in `all` |
| `all` | A list of models whose length is equal to `Lambda0`: |
| `...$lambda0` | Corresponding element in `Lambda0` |
| `...$lambda` | `sqrt(N) * lambda0` |
| `...$iter` | Number(s) of iterations |
| `...$Sigma` | Group-level posterior covariance matrices |
| `...$Mu` | Group-level posterior mean vectors |
| `...$a` | Slopes for group 1 |
| `...$b` | Intercepts for group 1 |
| `...$gamma` | DIF parameters for the slopes |
| `...$beta` | DIF parameters for the intercepts |
| `...$ll` | Log-likelihood |
| `...$l0` | Number of nonzero parameters in `gamma` and `beta` |
| `...$AIC` | Akaike Information Criterion |
| `...$BIC` | Bayesian Information Criterion |
| `...$GIC` | Generalized Information Criterion |

## See Also

`gvemm_DIF`, `lrt_DIF`, `coef.vemirt_DIF`, `print.vemirt_DIF`

### Examples

```
## Not run:
with(exampleDIF, em_DIF(Y, D, X))
## End(Not run)
```

---

## exampleData_2pl *Response data set for M2PL*

---

### Description

The response data set is simulated based on a between-item M2PL model with 5 factors. The true factor correlations are set as 0.1.

### Usage

```
exampleData_2pl
```

### Format

A data frame with 2000 respondents and 75 items

---

## exampleData_3pl *Response data set for M3PL*

---

### Description

The response data set is simulated based on a within-item M3PL model with 3 factors. The true factor correlations are set as 0.1.

### Usage

```
exampleData_3pl
```

### Format

A data frame with 2000 respondents and 45 items `exampleDIF` *Simulated Data Set for DIF Analysis*

## Description

Simulated Data Set for DIF Analysis

## Usage

```
exampleDIF
```

## Format

A list of components of the data set:

| | |
|---|---|
| `Y` | Item responses |
| `D` | Loading indicators |
| `X` | Group indicators |
| `j` | Number of DIF items (the first `j` items have DIF) |
| `params` | A list of true parameters used for generating the item responses: |
| `...$a` | Slopes |
| `...$b` | Negated intercepts |
| `...$theta` | Latent traits |

`exampleIndic_cfa2pl` *Factor-loading indicator matrix for M2PL-CFA*

## Description

The factor-loading indicator matrix can be used as an input for confirmatory factor analysis.

## Usage

```
exampleIndic_cfa2pl
```

## Format

A data frame with 75 items and 5 factors

## exampleIndic_cfa3pl *Factor-loading indicator matrix for M3PL-CFA*

### Description

The factor-loading indicator matrix can be used as an input for confirmatory factor analysis.

### Usage

```
exampleIndic_cfa3pl
```

### Format

A data frame with 45 items and 3 factors

## exampleIndic_efa2pl_c1 *Factor-loading indicator matrix for M2PL-EFA with lasso/ adaptive penalty under constraint 1*

### Description

The factor-loading indicator matrix can be used as an input for exploratory factor analysis with lasso/ adaptive lasso penalty under constraint 1.

### Usage

```
exampleIndic_efa2pl_c1
```

### Format

A data frame with 75 items and 5 factors. Items 1, 16, 31, 46 and 61 can be combined as an identity matrix to satisfy constraint 1

## exampleIndic_efa2pl_c2 *Factor-loading indicator matrix for M2PL-EFA with lasso/ adaptive penalty under constraint 2*

### Description

The factor-loading indicator matrix can be used as an input for exploratory factor analysis with lasso/ adaptive lasso penalty for constraint 1.

### Usage

```
exampleIndic_efa2pl_c2
```

### Format

A data frame with 75 items and 5 factors. Items 1, 16, 31, 46 and 61 can be combined as a triangular matrix to satisfy constraint 2

---

`exampleIndic_efa3pl_c1`

*Factor-loading indicator matrix for M3PL-EFA with lasso/ adaptive penalty under constraint 1*

---

### Description

The factor-loading indicator matrix can be used as an input for exploratory factor analysis with lasso/ adaptive lasso penalty under constraint 1.

### Usage

```
exampleIndic_efa3pl_c1
```

### Format

A data frame with 45 items and 3 factors. Items 1, 16, and 19 can be combined as an identity matrix to satisfy constraint 1

---

`exampleIndic_efa3pl_c2`

*Factor-loading indicator matrix for M3PL-EFA with lasso/ adaptive penalty under constraint 2*

---

### Description

The factor-loading indicator matrix can be used as an input for exploratory factor analysis with lasso/ adaptive lasso penalty for constraint 1.

### Usage

```
exampleIndic_efa3pl_c2
```

### Format

A data frame with 45 items and 3 factors. Items 1, 16, and 19 can be combined as a triangular matrix to satisfy constraint 2 `exampleItem_2pl` *True item parameters for M2PL*

## Description

True item parameters for M2PL

## Usage

```
exampleItem_2pl
```

## Format

An object of class `data.frame` with 75 rows and 6 columns.

`exampleItem_3pl` *True item parameters for M3PL*

## Description

True item parameters for M3PL

## Usage

```
exampleItem_3pl
```

## Format

An object of class `data.frame` with 45 rows and 5 columns.

`gvemm_DIF` *GVEMM Algorithms for DIF Detection in 2PL Models*

## Description

GVEMM Algorithms for DIF Detection in 2PL Models

### Usage

```
gvemm_DIF(
  Y,
  D,
  X,
  method = "IWGVEMM",
  Lambda0 = seq(0.2, 0.7, by = 0.1),
  criterion = "GIC",
  iter = 1000,
  eps = 0.001,
  c = 0.7,
  S = 10,
  M = 10,
  lr = 0.1
)
```

### Arguments

| | |
|---|---|
| Y | An $N \times J$ binary matrix of item responses |
| D | A $J \times K$ binary matrix of loading indicators |
| X | An $N$ dimensional vector of group indicators (integers from 1 to G) |
| method | Estimation algorithm, one of 'GVEMM' and 'IWGVEMM' |
| Lambda0 | A vector of lambda0 values for $L_1$ penalty (lambda is sqrt(N) * lambda0) |
| criterion | Information criterion for model selection, one of 'GIC' (recommended), 'BIC' and 'AIC' |
| iter | Maximum number of iterations |
| eps | Termination criterion on numerical accuracy |
| c | Constant for computing GIC |
| S | Sample size for approximating the expected lower bound ('IWGVEMM' only) |
| M | Sample size for approximating a tighter lower bound ('IWGVEMM' only) |
| lr | Learning rate for the Adam optimizer ('IWGVEMM' only) |

### Value

An object of class vemirt_DIF, which is a list containing three elements:

| | |
|---|---|
| fit | The best (with lowest information criterion) model, which is an element of all |
| best | The location of fit in all |
| all | A list of models whose length is equal to Lambda0: |
| ...$lambda0 | Corresponding element in Lambda0 |
| ...$lambda | sqrt(N) * lambda0 |
| ...$iter | Number(s) of iterations |
| ...$SIGMA | Person-level posterior covariance matrices |
| ...$MU | Person-level posterior mean vectors |
| ...$Sigma | Group-level posterior covariance matrices |
| ...$Mu | Group-level posterior mean vectors |

| | |
|---|---|
| `...$a` | Slopes for group 1 |
| `...$b` | Intercepts for group 1 |
| `...$gamma` | DIF parameters for the slopes |
| `...$beta` | DIF parameters for the intercepts |
| `...$ll` | Log-likelihood |
| `...$l0` | Number of nonzero parameters in `gamma` and `beta` |
| `...$AIC` | Akaike Information Criterion |
| `...$BIC` | Bayesian Information Criterion |
| `...$GIC` | Generalized Information Criterion |

## See Also

`em_DIF`, `lrt_DIF`, `coef.vemirt_DIF`, `print.vemirt_DIF`

## Examples

```
## Not run:
with(exampleDIF, gvemm_DIF(Y, D, X))
## End(Not run)
```

---

`gvem_2PLCFA` *Confirmatory M2PL Analysis*

---

## Description

Confirmatory M2PL Analysis

## Usage

```
gvem_2PLCFA(u, indic, max.iter = 5000, SE.est = FALSE)
```

## Arguments

| | |
|---|---|
| `u` | a $N \times J$ `matrix` or a `data.frame` that consists of binary responses of $N$ individuals to $J$ items. The missing values are coded as `NA` |
| `indic` | a $J \times K$ `matrix` or a `data.frame` that describes the factor loading structure of $J$ items to $K$ factors. It consists of binary values where 0 refers to the item is irrelevant with this factor, 1 otherwise |
| `max.iter` | the maximum number of iterations for the EM cycle; default is 5000 |
| `SE.est` | whether to estimate SE for item parameters using the updated supplemented expectation maximization (USEM); default is FALSE |

### Value

a list containing the following objects:

| | |
|---|---|
| `ra` | item discrimination parameters, a $J \times K$ `matrix` |
| `rb` | item difficulty parameters, vector of length $J$ |
| `reta` | variational parameters $\eta(\xi)$, a $N \times J$ matrix |
| `reps` | variational parameters $\xi$, a $N \times J$ matrix |
| `rsigma` | population variance-covariance matrix, a $K \times K$ matrix |
| `mu_i` | mean parameter for each person, a $K \times N$ matrix |
| `sig_i` | covariance matrix for each person, a $K \times K \times N$ array |
| `n` | the number of iterations for the EM cycle |
| `Q_mat` | factor loading structure, a $J \times K$ matrix |
| `GIC` | model fit index |
| `AIC` | model fit index |
| `BIC` | model fit index |
| `SE` | Standard errors of item parameters, a $J \times (K+1)$ matrix where the last column includes SE estimates for item difficulty parameters |

### See Also

`sgvem_3PLCFA`,`importanceSampling`,`bs_2PLCFA`

### Examples

```
## Not run:
gvem_2PLCFA(exampleData_2pl, exampleIndic_cfa2pl)
## End(Not run)
```

---

## `gvem_2PLEFA_adaptlasso` *Exploratory M2PL Analysis with Adaptive Lasso Penalty*

---

### Description

Exploratory M2PL Analysis with Adaptive Lasso Penalty

### Usage

```
gvem_2PLEFA_adaptlasso(
  u,
  indic,
  max.iter = 5000,
  constrain = "C1",
  non_pen = NULL,
  gamma = 2
)
```

## Arguments

| | |
|---|---|
| u | a $N \times J$ `matrix` or a `data.frame` that consists of binary responses of $N$ individuals to $J$ items. The missing values are coded as `NA` |
| indic | a $J \times K$ `matrix` or a `data.frame` that describes the factor loading structure of $J$ items to $K$ factors. It consists of binary values where 0 refers to the item is irrelevant with this factor, 1 otherwise. For exploratory factor analysis with adaptive lasso penalty, `indic` should be imposed certain constraints on the a $K \times K$ sub-matrix to ensure identifiability. The remaining parts do not assume any pre-specified zero structure but instead, the appropriate lasso penalty would recover the true zero structure. Also see `constrain` |
| max.iter | the maximum number of iterations for the EM cycle; default is 5000 |
| constrain | the constraint setting: `"C1"` or `"C2"`. To ensure identifiablity, `"C1"` sets a $K \times K$ sub-matrix of `indic` to be an identity matrix.This constraint anchor $K$ factors by designating $K$ items that load solely on each factor respectively. Note that the $K \times K$ matrix does not have to appear at the top of the `indic` matrix. `"C2"` sets the $K \times K$ sub-matrix to be a lower triangular matrix with the diagonal being ones. That is, there are test items associated with each factor for sure and they may be associated with other factors as well. Nonzero entries (in the lower triangular part) except for the diagonal entries of the sub-matrix are penalized during the estimation procedure. For instance, assume $K = 3$, then the `"C2"` constraint will imply the following submatrix: $C2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$. As shown, item 1 is allowed to only load on the first factor, item 2 will for sure load on the second factor but it may also load on the first factor (hence a penalty is added on the $(2, 1)$ element of `"C1"`, i.e., $C2_{2,1}$ ). Item 3 will for sure load on the third factor but it may also load on the first two factors. However, note that for all remaining items their loading vector will all be $(1, 1, 1)$ hence indistinguishable from the third anchor item. Therefore, we need to alert the algorithm that this third anchor item will for sure load on the third factor, and and whether or not it loads on the first two factors depends on the regularization results. Therefore, we need to specify `"non_pen="` to identify the $K$th anchor item. Although, `"C2"` is much weaker than `"C1"`, it still ensures empirical identifiability. Default is `"C1"`. During estimation, under both the `"C1"` and `"C1"` constraints, the population means and variances are constrained to be 0 and 1, respectively. |
| non_pen | the index of an item which is associated with each factor to satisfy `"C2"`. For `C1`, the input can be `NULL` |
| gamma | a numerical value of adaptive lasso parameter. Zou (2006) recommended three values, 0.5, 1, and 2. The default value is 2. |

## Value

a list containing the following objects:

| | |
|---|---|
| ra | item discrimination parameters, a $J \times K$ `matrix` |
| rb | item difficulty parameters, vector of length $J$ |
| reta | variational parameters $\eta(\xi)$, a $N \times J$ matrix |
| reps | variational parameters $\xi$, a $N \times J$ matrix |
| rsigma | population variance-covariance matrix, a $K \times K$ matrix |
| mu_i | mean parameter for each person, a $K \times N$ matrix |

| | |
|---|---|
| `sig_i` | covariance matrix for each person, a $K \times K \times N$ array |
| `n` | the number of iterations for the EM cycle |
| `Q_mat` | factor loading structure, a $J \times K$ matrix |
| `GIC` | model fit index |
| `AIC` | model fit index |
| `BIC` | model fit index |
| `lbd` | numerical value of lasso penalty parameter $\lambda$ |

## References


Cho, A. E., Xiao, J., Wang, C., & Xu, G. (2022). Regularized Variational Estimation for Exploratory Item Factor Analysis. *Psychometrika*. https://doi.org/10.1007/s11336-022-09874-6

Zou, H. (2006). The adaptive LASSO and its oracle properties. *Journal of the American Statistical Association, 7*, 1011418–1429.


## See Also

`gvem_2PLEFA_rot`, `gvem_2PLEFA_lasso`, `exampleIndic_efa2pl_c1`, `exampleIndic_efa2pl_c2`

## Examples

```
## Not run:
gvem_2PLEFA_adaptlasso(exampleData_2pl, exampleIndic_efa2pl_c1,constrain="C1",non_pen=NULL,gamma=2)
gvem_2PLEFA_adaptlasso(exampleData_2pl, exampleIndic_efa2pl_c2,constrain="C2",non_pen=61,gamma=2)
## End(Not run)
```

---

# `gvem_2PLEFA_lasso` *Exploratory M2PL Analysis with Lasso Penalty*

---

## Description

Exploratory M2PL Analysis with Lasso Penalty

## Usage

```
gvem_2PLEFA_lasso(u, indic, max.iter = 5000, constrain = "C1", non_pen = NULL)
```

## Arguments

| | |
|---|---|
| `u` | a $N \times J$ `matrix` or a `data.frame` that consists of binary responses of $N$ individuals to $J$ items. The missing values are coded as `NA` |
| `indic` | a $J \times K$ `matrix` or a `data.frame` that describes the factor loading structure of $J$ items to $K$ factors. It consists of binary values where 0 refers to the item is irrelevant with this factor, 1 otherwise. For exploratory factor analysis with lasso penalty, `indic` should be imposed certain constraints on the a $K \times K$ sub-matrix to ensure identifiability. The remaining parts do not assume any pre-specified zero structure but instead, the appropriate lasso penalty would recover the true zero structure. Also see `constrain` |
| `max.iter` | the maximum number of iterations for the EM cycle; default is 5000 |

| | |
|---|---|
| constrain | the constraint setting: "C1" or "C2". To ensure identifiablity, "C1" sets a $K \times K$ sub-matrix of indic to be an identity matrix.This constraint anchor $K$ factors by designating $K$ items that load solely on each factor respectively. Note that the $K \times K$ matrix does not have to appear at the top of the indic matrix. "C2" sets the $K \times K$ sub-matrix to be a lower triangular matrix with the diagonal being ones. That is, there are test items associated with each factor for sure and they may be associated with other factors as well. Nonzero entries (in the lower triangular part) except for the diagonal entries of the sub-matrix are penalized during the estimation procedure. For instance, assume $K = 3$, then the "C2" constraint will imply the following submatrix: $C2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$. As shown, item 1 is allowed to only load on the first factor, item 2 will for sure load on the second factor but it may also load on the first factor (hence a penalty is added on the $(2, 1)$ element of "C1", i.e., $C2_{2,1}$ ). Item 3 will for sure load on the third factor but it may also load on the first two factors. However, note that for all remaining items their loading vector will all be $(1, 1, 1)$ hence indistinguishable from the third anchor item. Therefore, we need to alert the algorithm that this third anchor item will for sure load on the third factor, and and whether or not it loads on the first two factors depends on the regularization results. Therefore, we need to specify "non_pen=" to identify the $K$th anchor item. Although, "C2" is much weaker than "C1", it still ensures empirical identifiability. Default is "C1". During estimation, under both the "C1" and "C1" constraints, the population means and variances are constrained to be 0 and 1, respectively. |
| non_pen | the index of an item which is associated with each factor to satisfy "C2". For C1, the input can be NULL |

## Value

a list containing the following objects:

| | |
|---|---|
| ra | item discrimination parameters, a $J \times K$ matrix |
| rb | item difficulty parameters, vector of length $J$ |
| reta | variational parameters $\eta(\xi)$, a $N \times J$ matrix |
| reps | variational parameters $\xi$, a $N \times J$ matrix |
| rsigma | population variance-covariance matrix, a $K \times K$ matrix |
| mu_i | mean parameter for each person, a $K \times N$ matrix |
| sig_i | covariance matrix for each person, a $K \times K \times N$ array |
| n | the number of iterations for the EM cycle |
| Q_mat | factor loading structure, a $J \times K$ matrix |
| GIC | model fit index |
| AIC | model fit index |
| BIC | model fit index |
| lbd | numerical value of lasso penalty parameter $\lambda$ |

## References

Cho, A. E., Xiao, J., Wang, C., & Xu, G. (2022). Regularized Variational Estimation for Exploratory Item Factor Analysis. *Psychometrika*. https://doi.org/10.1007/s11336-022-09874-6

## See Also

gvem_2PLEFA_rot, gvem_2PLEFA_adaptlasso, exampleIndic_efa2pl_c1, exampleIndic_efa2pl_c2

## Examples

```
## Not run:
gvem_2PLEFA_lasso(exampleData_2pl, exampleIndic_efa2pl_c1,constrain="C1")
gvem_2PLEFA_lasso(exampleData_2pl, exampleIndic_efa2pl_c2,constrain="C2",non_pen=61)
## End(Not run)
```

---

# gvem_2PLEFA_rot   *Exploratory M2PL Analysis with Post-hoc Rotation*

---

## Description

Exploratory M2PL Analysis with Post-hoc Rotation

## Usage

```
gvem_2PLEFA_rot(u, domain, max.iter = 5000, rot = "Promax")
```

## Arguments

| | |
|---|---|
| u | a $N \times J$ matrix or a data.frame that consists of binary responses of $N$ individuals to $J$ items. The missing values are coded as NA |
| domain | the number of factors |
| max.iter | the maximum number of iterations for the EM cycle; default is 5000 |
| rot | the post-hoc rotation method: Promax or CF-Quartimax; default is "Promax", but may also be "cfQ" for conducting the CF-Quartimax rotation |

## Value

a list containing the following objects:

| | |
|---|---|
| ra | item discrimination parameters, a $J \times K$ matrix |
| rb | item difficulty parameters, vector of length $J$ |
| reta | variational parameters $\eta(\xi)$, a $N \times J$ matrix |
| reps | variational parameters $\xi$, a $N \times J$ matrix |
| rsigma | population variance-covariance matrix, a $K \times K$ matrix |
| mu_i | mean parameter for each person, a $K \times N$ matrix |
| sig_i | covariance matrix for each person, a $K \times K \times N$ array |
| n | the number of iterations for the EM cycle |
| rk | factor loadings, a $J \times K$ matrix |
| Q_mat | factor loading structure, a $J \times K$ matrix |
| GIC | model fit index |
| AIC | model fit index |
| BIC | model fit index |
| ur_a | item discrimination parameters before conducting the rotation, a $J \times K$ matrix |

## See Also

gvem_2PLEFA_lasso, gvem_2PLEFA_adaptlasso

## Examples

```
## Not run:
gvem_2PLEFA_rot(exampleData_2pl, domain=5,max.iter=3000)
gvem_2PLEFA_rot(exampleData_2pl, domain=5,rot="cfQ")
## End(Not run)
```

---

importanceSampling *Importance Weighted Version of GVEM Analysis for M2PL Models*

---

## Description

An importance weighted version of GVEM (i.e., IW-GVEM) can be implemented to correct the bias on item parameters under M2PL models

## Usage

```
importanceSampling(u, gvem_result, S = 10, M = 10, max.iter = 10)
```

## Arguments

| | |
|---|---|
| u | a $N \times J$ matrix or a data.frame that consists of binary responses of $N$ individuals to $J$ items. The missing values are coded as NA |
| gvem_result | a list that includes exploratory or confirmatory GVEM results for M2PL models. |
| S | the number of times to draw samples;default is 10 |
| M | the number of samples drawn from the variational distributions;default is 10 |
| max.iter | the maximum number of iterations for the EM cycle; default is 10 |

## Value

a list containing the following objects:

| | |
|---|---|
| ra | item discrimination parameters estimated by GVEM, a $J \times K$ matrix |
| rb | item difficulty parameters estimated by GVEM, vector of length $J$ |
| reta | variational parameters $\eta(\xi)$, a $N \times J$ matrix |
| reps | variational parameters $\xi$, a $N \times J$ matrix |
| rsigma | population variance-covariance matrix estimated by GVEM, a $K \times K$ matrix |
| mu_i | mean parameter for each person, a $K \times N$ matrix |
| sig_i | covariance matrix for each person, a $K \times K \times N$ array |
| n | the number of iterations for the EM cycle |
| rk | factor loadings, a $J \times K$ matrix, for exploratory analysis only |
| Q_mat | factor loading structure, a $J \times K$ matrix |
| GIC | model fit index |
| AIC | model fit index |

| | |
|---|---|
| BIC | model fit index |
| SE | Standard errors of item parameters, a $J \times (K+1)$ matrix where the last column includes SE estimates for item difficulty parameters, for confirmatory analysis only |
| ur_a | item discrimination parameters before conducting the rotation, a $J \times K$ matrix, for exploratory analysis only |
| new_a | item discrimination parameters estimated by IW-GVEM, a $J \times K$ matrix |
| new_b | item difficulty parameters estimated by IW-GVEM, vector of length $J$ |
| new_Sigma_theta | population variance-covariance matrix estimated by IV-GVEM, a $K \times K$ matrix |
| best_lr | The learning rate used for importance sampling |
| best_lr | The lower bound value for importance sampling |

## See Also

gvem_2PLCFA, gvem_2PLEFA_rot,bs_2PLCFA

## Examples

```
## Not run:
CFA_result <- gvem_2PLCFA(exampleData_2pl, exampleIndic_cfa2pl)
importanceSampling(exampleData_2pl,CFA_result)
## End(Not run)
```

---

# lrt_DIF *Likelihood Ratio Test for DIF Detection in 2PL Models*

---

## Description

Likelihood Ratio Test for DIF Detection in 2PL Models

## Usage

```
lrt_DIF(Y, D, X, unif = F)
```

## Arguments

| | |
|---|---|
| Y | An $N \times J$ binary matrix of item responses |
| D | A $J \times K$ binary matrix of loading indicators |
| X | An $N$ dimensional vector of group indicators (integers from 1 to G) |
| unif | Whether to detect uniform DIF only |

## Value

A list:

| | |
|---|---|
| Sigma | Group-level posterior covariance matrices |
| Mu | Group-level posterior mean vectors |
| a | Slopes for group 1 |
| b | Intercepts for group 1 |
| gamma | DIF parameters for the slopes |
| beta | DIF parameters for the intercepts |

## See Also

em_DIF, gvemm_DIF, coef.vemirt_DIF, print.vemirt_DIF

## Examples

```
## Not run:
with(exampleDIF, lrt_DIF(Y, D, X))
## End(Not run)
```

---

# pa_poly — *Parallel analysis using polychoric correlation*

---

## Description

Identify the number of factors

## Usage

```
pa_poly(data, n.iter = 10, figure = TRUE)
```

## Arguments

| | |
|---|---|
| data | a $N \times J$ matrix or a data.frame that consists of the responses of $N$ individuals to $J$ items without any missing values. The responses are binary or polytomous. |
| n.iter | Number of simulated analyses to perform |
| figure | By default, pa_poly draws an eigenvalue plot. If FALSE, it suppresses the graphic output |

## Value

pa_poly returns a data.frame with the eigenvalues for the real data and the simulated data.

## Examples

```
## Not run:
pa_poly(exampleData_2pl, n.iter=20)
## End(Not run)
```

## print.vemirt_DIF *Print DIF Items*

### Description

Print DIF Items

### Usage

```
print(x, criterion = NULL)
```

### Arguments

| | |
|---|---|
| x | An object of class vemirt_DIF |
| criterion | Information criterion for model selection, one of 'AIC', 'BIC' and 'GIC', otherwise use the criterion specified when fitting the model(s) |

### See Also

em_DIF, gvemm_DIF, lrt_DIF, coef.vemirt_DIF

## sgvem_3PLCFA *Stochastic GVEM for Confirmatory M3PL Analysis*

### Description

Stochastic GVEM for Confirmatory M3PL Analysis

### Usage

```
sgvem_3PLCFA(
  u,
  indic,
  samp = 50,
  forgetrate = 0.51,
  mu_b,
  sigma2_b,
  Alpha,
  Beta,
  max.iter = 5000
)
```

### Arguments

| | |
|---|---|
| u | a $N \times J$ matrix or a data.frame that consists of binary responses of $N$ individuals to $J$ items. The missing values are coded as NA |
| indic | a $J \times K$ matrix or a data.frame that describes the factor loading structure of $J$ items to $K$ factors. It consists of binary values where 0 refers to the item is irrelevant with this factor, 1 otherwise |

| | |
|---|---|
| samp | a subsample for each iteration; default is 50 |
| forgetrate | the forget rate for the stochastic algorithm. The value should be within the range from 0.5 to 1. Default is 0.51 |
| mu_b | the mean parameter for the prior distribution of item difficulty parameters |
| sigma2_b | the variance parameter for the prior distribution of item difficulty parameters |
| Alpha | the $\alpha$ parameter for the prior distribution of guessing parameters |
| Beta | the $\beta$ parameter for the prior distribution of guessing parameters |
| max.iter | the maximum number of iterations for the EM cycle; default is 5000 |

## Value

a list containing the following objects:

| | |
|---|---|
| ra | item discrimination parameters, a $J \times K$ matrix |
| rb | item difficulty parameters, vector of length $J$ |
| rc | item guessing parameters, vector of length $J$ |
| rs | variational parameters $s$, a $N \times J$ matrix |
| reta | variational parameters $\eta(\xi)$, a $N \times J$ matrix |
| reps | variational parameters $\xi$, a $N \times J$ matrix |
| rsigma | population variance-covariance matrix, a $K \times K$ matrix |
| mu_i | mean parameter for each person, a $K \times N$ matrix |
| sig_i | covariance matrix for each person, a $K \times K \times N$ array |
| n | the number of iterations for the EM cycle |
| Q_mat | factor loading structure, a $J \times K$ matrix |
| GIC | model fit index |
| AIC | model fit index |
| BIC | model fit index |

## References

Cho, A. E., Wang, C., Zhang, X., & Xu, G. (2021). Gaussian variational estimation for multidimensional item response theory. *British Journal of Mathematical and Statistical Psychology, 74*, 52-85.

Cho, A. E., Xiao, J., Wang, C., & Xu, G. (2022). Regularized Variational Estimation for Exploratory Item Factor Analysis. *Psychometrika*. https://doi.org/10.1007/s11336-022-09874-6

## See Also

gvem_2PLCFA

## Examples

```
## Not run:
sgvem_3PLCFA(exampleData_3pl, exampleIndic_cfa3pl,samp=50,forgetrate=0.51,
mu_b=0,sigma2_b=4,Alpha=10,Beta=40)
## End(Not run)
```

`sgvem_3PLEFA_adaptlasso` *Stochastic GVEM with Adaptive Lasso Penalty for Exploratory M3PL Analysis*

## Description

Stochastic GVEM with Adaptive Lasso Penalty for Exploratory M3PL Analysis

## Usage

```
sgvem_3PLEFA_adaptlasso(
  u,
  indic,
  samp = 50,
  forgetrate = 0.51,
  mu_b,
  sigma2_b,
  Alpha,
  Beta,
  max.iter = 5000,
  constrain = "C1",
  non_pen = NULL,
  gamma = 2
)
```

## Arguments

| | |
|---|---|
| `u` | a $N \times J$ `matrix` or a `data.frame` that consists of binary responses of $N$ individuals to $J$ items. The missing values are coded as `NA` |
| `indic` | a $J \times K$ `matrix` or a `data.frame` that describes the factor loading structure of $J$ items to $K$ factors. It consists of binary values where 0 refers to the item is irrelevant with this factor, 1 otherwise. For exploratory factor analysis with lasso penalty, `indic` should be imposed certain constraints on the a $K \times K$ sub-matrix to ensure identifiability. The remaining parts do not assume any pre-specified zero structure but instead, the appropriate lasso penalty would recover the true zero structure. Also see `constrain` |
| `samp` | a subsample for each iteration; default is 50 |
| `forgetrate` | the forget rate for the stochastic algorithm. The value should be within the range from 0.5 to 1. Default is 0.51 |
| `mu_b` | the mean parameter for the prior distribution of item difficulty parameters |
| `sigma2_b` | the variance parameter for the prior distribution of item difficulty parameters |
| `Alpha` | the $\alpha$ parameter for the prior distribution of guessing parameters |
| `Beta` | the $\beta$ parameter for the prior distribution of guessing parameters |
| `max.iter` | the maximum number of iterations for the EM cycle; default is 5000 |
| `constrain` | the constraint setting: `"C1"` or `"C2"`. To ensure identifiablity, `"C1"` sets a $K \times K$ sub-matrix of `indic` to be an identity matrix.This constraint anchor $K$ factors by designating $K$ items that load solely on each factor respectively. Note that |

the $K \times K$ matrix does not have to appear at the top of the indic matrix. "C2" sets the $K \times K$ sub-matrix to be a lower triangular matrix with the diagonal being ones. That is, there are test items associated with each factor for sure and they may be associated with other factors as well. Nonzero entries (in the lower triangular part) except for the diagonal entries of the sub-matrix are penalized during the estimation procedure. For instance, assume $K = 3$, then the "C2" constraint will imply the following submatrix: $C2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$. As shown, item 1 is allowed to only load on the first factor, item 2 will for sure load on the second factor but it may also load on the first factor (hence a penalty is added on the $(2, 1)$ element of "C1", i.e., $C2_{2,1}$ ). Item 3 will for sure load on the third factor but it may also load on the first two factors. However, note that for all remaining items their loading vector will all be $(1, 1, 1)$ hence indistinguishable from the third anchor item. Therefore, we need to alert the algorithm that this third anchor item will for sure load on the third factor, and and whether or not it loads on the first two factors depends on the regularization results. Therefore, we need to specify "non_pen=" to identify the $K$th anchor item. Although, "C2" is much weaker than "C1", it still ensures empirical identifiability. Default is "C1". During estimation, under both the "C1" and "C1" constraints, the population means and variances are constrained to be 0 and 1, respectively.

| | |
|---|---|
| non_pen | the index of an item which is associated with each factor to satisfy "C2". For C1, the input can be NULL |
| gamma | a numerical value of adaptive lasso parameter. Zou (2006) recommended three values, 0.5, 1, and 2. The default value is 2. |

## Value

a list containing the following objects:

| | |
|---|---|
| ra | item discrimination parameters, a $J \times K$ matrix |
| rb | item difficulty parameters, vector of length $J$ |
| rc | item guessing parameters, vector of length $J$ |
| rs | variational parameters $s$, a $N \times J$ matrix |
| reta | variational parameters $\eta(\xi)$, a $N \times J$ matrix |
| reps | variational parameters $\xi$, a $N \times J$ matrix |
| rsigma | population variance-covariance matrix, a $K \times K$ matrix |
| mu_i | mean parameter for each person, a $K \times N$ matrix |
| sig_i | covariance matrix for each person, a $K \times K \times N$ array |
| n | the number of iterations for the EM cycle |
| Q_mat | factor loading structure, a $J \times K$ matrix |
| GIC | model fit index |
| AIC | model fit index |
| BIC | model fit index |
| lbd | numerical value of lasso penalty parameter $\lambda$ |

## References

Cho, A. E., Xiao, J., Wang, C., & Xu, G. (2022). Regularized Variational Estimation for Exploratory Item Factor Analysis. *Psychometrika*. https://doi.org/10.1007/s11336-022-09874-6

Zou, H. (2006). The adaptive LASSO and its oracle properties. *Journal of the American Statistical Association, 7*, 1011418–1429.

## See Also

sgvem_3PLEFA_rot, sgvem_3PLEFA_lasso, exampleIndic_efa3pl_c1, exampleIndic_efa3pl_c2

## Examples

```
## Not run:
sgvem_3PLEFA_adaptlasso(exampleData_3pl, exampleIndic_efa3pl_c1,samp=50,
forgetrate=0.51,mu_b=0,sigma2_b=4,Alpha=10,Beta=40,max.iter=5000,
constrain="C1",non_pen=NULL,gamma=2)
sgvem_3PLEFA_adaptlasso(exampleData_3pl, exampleIndic_efa3pl_c2,samp=50,
forgetrate=0.51,mu_b=0,sigma2_b=4,Alpha=10,Beta=40,max.iter=5000,
constrain="C2",non_pen=19,gamma=2)
## End(Not run)
```

---

# sgvem_3PLEFA_lasso *Stochastic GVEM with Lasso Penalty for Exploratory M3PL Analysis*

---

## Description

Stochastic GVEM with Lasso Penalty for Exploratory M3PL Analysis

## Usage

```
sgvem_3PLEFA_lasso(
  u,
  indic,
  samp = 50,
  forgetrate = 0.51,
  mu_b,
  sigma2_b,
  Alpha,
  Beta,
  max.iter = 5000,
  constrain = "C1",
  non_pen = NULL
)
```

## Arguments

| | |
|---|---|
| u | a $N \times J$ `matrix` or a `data.frame` that consists of binary responses of $N$ individuals to $J$ items. The missing values are coded as `NA` |

| | |
|---|---|
| indic | a $J \times K$ matrix or a data.frame that describes the factor loading structure of $J$ items to $K$ factors. It consists of binary values where 0 refers to the item is irrelevant with this factor, 1 otherwise. For exploratory factor analysis with lasso penalty, indic should be imposed certain constraints on the a $K \times K$ sub-matrix to ensure identifiability. The remaining parts do not assume any pre-specified zero structure but instead, the appropriate lasso penalty would recover the true zero structure. Also see constrain |
| samp | a subsample for each iteration; default is 50 |
| forgetrate | the forget rate for the stochastic algorithm. The value should be within the range from 0.5 to 1. Default is 0.51 |
| mu_b | the mean parameter for the prior distribution of item difficulty parameters |
| sigma2_b | the variance parameter for the prior distribution of item difficulty parameters |
| Alpha | the $\alpha$ parameter for the prior distribution of guessing parameters |
| Beta | the $\beta$ parameter for the prior distribution of guessing parameters |
| max.iter | the maximum number of iterations for the EM cycle; default is 5000 |
| constrain | the constraint setting: "C1" or "C2". To ensure identifiablity, "C1" sets a $K \times K$ sub-matrix of indic to be an identity matrix.This constraint anchor $K$ factors by designating $K$ items that load solely on each factor respectively. Note that the $K \times K$ matrix does not have to appear at the top of the indic matrix. "C2" sets the $K \times K$ sub-matrix to be a lower triangular matrix with the diagonal being ones. That is, there are test items associated with each factor for sure and they may be associated with other factors as well. Nonzero entries (in the lower triangular part) except for the diagonal entries of the sub-matrix are penalized during the estimation procedure. For instance, assume $K = 3$, then the "C2" constraint will imply the following submatrix: $C2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$. As shown, item 1 is allowed to only load on the first factor, item 2 will for sure load on the second factor but it may also load on the first factor (hence a penalty is added on the $(2,1)$ element of "C1", i.e., $C2_{2,1}$ ). Item 3 will for sure load on the third factor but it may also load on the first two factors. However, note that for all remaining items their loading vector will all be $(1,1,1)$ hence indistinguishable from the third anchor item. Therefore, we need to alert the algorithm that this third anchor item will for sure load on the third factor, and and whether or not it loads on the first two factors depends on the regularization results. Therefore, we need to specify "non_pen=" to identify the $K$th anchor item. Although, "C2" is much weaker than "C1", it still ensures empirical identifiability. Default is "C1". During estimation, under both the "C1" and "C1" constraints, the population means and variances are constrained to be 0 and 1, respectively. |
| non_pen | the index of an item which is associated with each factor to satisfy "C2". For C1, the input can be NULL |

## Value

a list containing the following objects:

| | |
|---|---|
| ra | item discrimination parameters, a $J \times K$ matrix |
| rb | item difficulty parameters, vector of length $J$ |
| rc | item guessing parameters, vector of length $J$ |
| rs | variational parameters $s$, a $N \times J$ matrix |

| | |
|---|---|
| reta | variational parameters $\eta(\xi)$, a $N \times J$ matrix |
| reps | variational parameters $\xi$, a $N \times J$ matrix |
| rsigma | population variance-covariance matrix, a $K \times K$ matrix |
| mu_i | mean parameter for each person, a $K \times N$ matrix |
| sig_i | covariance matrix for each person, a $K \times K \times N$ array |
| n | the number of iterations for the EM cycle |
| Q_mat | factor loading structure, a $J \times K$ matrix |
| GIC | model fit index |
| AIC | model fit index |
| BIC | model fit index |
| lbd | numerical value of lasso penalty parameter $\lambda$ |

## References

Cho, A. E., Xiao, J., Wang, C., & Xu, G. (2022). Regularized Variational Estimation for Exploratory Item Factor Analysis. *Psychometrika*. https://doi.org/10.1007/s11336-022-09874-6

## See Also

sgvem_3PLEFA_rot, sgvem_3PLEFA_adaptlasso, exampleIndic_efa3pl_c1, exampleIndic_efa3pl_c2

## Examples

```
## Not run:
sgvem_3PLEFA_lasso(exampleData_3pl, exampleIndic_efa3pl_c1,samp=50,
forgetrate=0.51,mu_b=0,sigma2_b=4,Alpha=10,Beta=40,max.iter=5000,
constrain="C1",non_pen=NULL)
sgvem_3PLEFA_lasso(exampleData_3pl, exampleIndic_efa3pl_c2,samp=50,
forgetrate=0.51,mu_b=0,sigma2_b=4,Alpha=10,Beta=40,max.iter=5000,
constrain="C2",non_pen=19)
## End(Not run)
```

---

sgvem_3PLEFA_rot *Stochastic GVEM for Exploratory M3PL Analysis*

---

## Description

Stochastic GVEM for Exploratory M3PL Analysis

## Usage

```
sgvem_3PLEFA_rot(
  u,
  domain,
  samp = 50,
  forgetrate = 0.51,
  mu_b,
  sigma2_b,
  Alpha,
```

```
  Beta,
  max.iter = 5000,
  rot = "Promax"
)
```

## Arguments

| | |
|---|---|
| u | a $N \times J$ matrix or a data.frame that consists of binary responses of $N$ individuals to $J$ items. The missing values are coded as NA |
| domain | the number of factors |
| samp | a subsample for each iteration; default is 50 |
| forgetrate | the forget rate for the stochastic algorithm. The value should be within the range from 0.5 to 1. Default is 0.51 |
| mu_b | the mean parameter for the prior distribution of item difficulty parameters |
| sigma2_b | the variance parameter for the prior distribution of item difficulty parameters |
| Alpha | the $\alpha$ parameter for the prior distribution of guessing parameters |
| Beta | the $\beta$ parameter for the prior distribution of guessing parameters |
| max.iter | the maximum number of iterations for the EM cycle; default is 5000 |
| rot | the post-hoc rotation method: Promax or CF-Quartimax; default is "Promax", but may also be "cfQ" for conducting the CF-Quartimax rotation |

## Value

a list containing the following objects:

| | |
|---|---|
| ra | item discrimination parameters, a $J \times K$ matrix |
| rb | item difficulty parameters, vector of length $J$ |
| rc | item guessing parameters, vector of length $J$ |
| rs | variational parameters $s$, a $N \times J$ matrix |
| reta | variational parameters $\eta(\xi)$, a $N \times J$ matrix |
| reps | variational parameters $\xi$, a $N \times J$ matrix |
| rsigma | population variance-covariance matrix, a $K \times K$ matrix |
| mu_i | mean parameter for each person, a $K \times N$ matrix |
| sig_i | covariance matrix for each person, a $K \times K \times N$ array |
| n | the number of iterations for the EM cycle |
| Q_mat | factor loading structure, a $J \times K$ matrix |
| rk | factor loadings, a $J \times K$ matrix |
| GIC | model fit index |
| AIC | model fit index |
| BIC | model fit index |
| ur_a | item discrimination parameters before conducting the rotation, a $J \times K$ matrix |

## See Also

sgvem_3PLEFA_lasso, sgvem_3PLEFA_adaptlasso

## Examples

```
## Not run:
sgvem_3PLEFA_rot(exampleData_3pl, 3,samp=50,forgetrate=0.51,
mu_b=0,sigma2_b=4,Alpha=10,Beta=40,max.iter=5000,rot="Promax")
## End(Not run)
```

# Index