Package ‘edina’

March 25, 2020

Title  Bayesian Estimation of an Exploratory Deterministic Input, Noisy and Gate Model

Version  0.1.1

Description  Perform a Bayesian estimation of the exploratory deterministic input, noisy and gate (EDINA) cognitive diagnostic model described by Chen et al. (2018) <doi:10.1007/s11336-017-9579-4>.

URL  https://github.com/tmsalab/edina

BugReports  https://github.com/tmsalab/edina/issues

License  GPL (>= 2)

Depends  R (>= 3.5.0)

Imports  Rcpp, ggplot2, jjb, reshape2

LinkingTo  Rcpp, RcppArmadillo, rgen

Suggests  simcdm

LazyData  true

RoxygenNote  7.1.0

Encoding  UTF-8

Language  en-US

NeedsCompilation  yes

Author  James Joseph Balamuta [aut, cre, cph]
  (<https://orcid.org/0000-0003-2826-8458>),
  Steven Andrew Culpepper [aut, cph]
  (<https://orcid.org/0000-0003-4226-6176>),
  Jeffrey A. Douglas [aut]

Maintainer  James Joseph Balamuta <balamut2@illinois.edu>

Repository  CRAN

Date/Publication  2020-03-25 13:50:15 UTC
as_q_matrix

Description

as_q_matrix acts as an aliases.

Usage

as_q_matrix(x, ...)

# S3 method for class 'data.frame'
as_q_matrix(x, ...)

# S3 method for class 'matrix'
as_q_matrix(x, ...)

# Default S3 method:
as_q_matrix(x, ...)

Arguments

x Either a data.frame or matrix.
...
Not used
Value

A `q_matrix` object.

See Also

`q_matrix()`

Examples

# Q matrix values
x = matrix(c(1, 0, 0), nrow = 2)

# Construct class
q_mat = as_q_matrix(x)

---

**autoplot.auto_edina**  
*Graph the Auto EDINA Object*

Description

Presents either the fitting of model heuristics or the evolution of parameters on a graph

Usage

```r
## S3 method for class 'auto_edina'
autoplot(
  object,
  type = c("selection", "guessing", "slipping", "evolution"),
  ...
)
```

Arguments

- `object`: An `auto_edina` object.
- `type`: Kind of graph to display. Valid types: "selection" or "evolution".
- `...`: Not used.

Value

A `ggplot2` object.

See Also

`auto_edina()`, `best_model()`, `model_selection_graph()`, `parameter_evolution_graph()`
Examples

```r
if(requireNamespace("simcdm", quietly = TRUE)) {

  # Set a seed for reproducibility
  set.seed(1512)

  # Setup data simulation parameters
  N = 2 # Number of Examinees / Subjects
  J = 10 # Number of Items
  K = 2 # Number of Skills / Attributes

  # Note:
  # Sample size and attributes have been reduced to create a minimally
  # viable example that can be run during CRAN's automatic check.
  # Please make sure to have a larger sample size...

  # Assign slipping and guessing values for each item
  ss = gs = rep(.2, J)

  # Simulate an identifiable Q matrix
  Q = simcdm::sim_q_matrix(J, K)

  # Simulate subject attributes
  subject_alphas = simcdm::sim_subject_attributes(N, K)

  # Simulate items under the DINA model
  items_dina = simcdm::sim_dina_items(subject_alphas, Q, ss, gs)

  # Requires at least 15 seconds of execution time.
  # Three EDINA models will be fit with increasing number of attributes.
  model_set_edina = auto_edina(items_dina, k = 2:4)

  # Visualize results
  autoplot(model_set_edina, type = "selection")

  # Equivalent to:
  model_selection_graph(model_set_edina)

  # View model parameters
  autoplot(model_set_edina, type = "guessing")

  # Or directly call with:
  parameter_evolution_graph(model_set_edina, type = "guessing")
}
```

---

**auto_edina**

*Auto EDINA model selection routine*
**Description**

Automatically select an appropriate $K$ dimension for a $Q$ matrix under the Exploratory Deterministic Input, Noise And gate (EDINA) Model.

**Usage**

```r
auto_edina(data, k = 2:4, burnin = 10000, chain_length = 20000)
```

**Arguments**

- `data` Binary responses to assessments in matrix form with dimensions $N \times J$.
- `k` Number of Attribute Levels as a positive integer.
- `burnin` Number of Observations to discard on the chain.
- `chain_length` Length of the MCMC chain

**Value**

An `auto_edina` object that contains:
- `edina_models`: A list containing all estimated edina model objects.
- `criterions`: Information criterions calculated for each model
- `k_checked`: Varying $k$ dimensions checked.
- `j`: Number of Items

**See Also**

`autoplot.auto_edina()`, `best_model()`, `model_selection_graph()`, `parameter_evolution_graph()`

**Examples**

```r
if(requireNamespace("simcdm", quietly = TRUE)) {

  # Set a seed for reproducibility
  set.seed(1512)

  # Setup data simulation parameters
  N = 15  # Number of Examinees / Subjects
  J = 10  # Number of Items
  K = 2   # Number of Skills / Attributes

  # Note:
  # Sample size and attributes have been reduced to create a minimally
  # viable example that can be run during CRAN's automatic check.
  # Please make sure to have a larger sample size...

  # Assign slipping and guessing values for each item
  ss = gs = rep(.2, J)

  # Simulate an identifiable Q matrix
```
Q = simcdm::sim_q_matrix(J, K)

# Simulate subject attributes
subject_alphas = simcdm::sim_subject_attributes(N, K)

# Simulate items under the DINA model
items_dina = simcdm::sim_dina_items(subject_alphas, Q, ss, gs)

# Requires at least 15 seconds of execution time.
# Three EDINA models will be fit with increasing number of attributes.
model_set_edina = auto_edina(items_dina, k = 2:4)

# Display results
model_set_edina

# Retrieve criterion table
table = summary(model_set_edina)

# Extract "best model"
best_model(model_set_edina)

---

**best_model**

*Extract the Best Model*

**Description**

Extracts the best model from the auto_*() search procedure.

**Usage**

```r
best_model(x, ...)
```

### S3 method for class 'auto_edina'

```r
best_model(x, ic = c("ppp", "bic", "dic"), ...)
```

**Arguments**

- **x**
  - An auto_edina object

- **...**
  - Not used.

- **ic**
  - Information criterion name. Default "ppp".

**Value**

An edina model object corresponding to the smallest value of requested information criterion.
BIC.edina

See Also

DIC.edina(), BIC.edina(), PPP.edina()

---

**BIC.edina**

*Bayesian Information Criterion (BIC)*

**Description**

Calculate BIC for EDINA models.

**Usage**

```r
## S3 method for class 'edina'
BIC(object, ...)
```

**Arguments**

- `object` An edina object
- `...` Not used.

**Value**

The BIC value of the given model.

**BIC Computation Procedure**

\[
BIC = -2 \log p(y|\hat{\theta}) + (k + 2) \cdot j + 2^k \log(n)
\]

**See Also**

PPP.edina(), DIC.edina()
DIC

Deviance Information Criterion (DIC)

Description

Calculate DIC for EDINA models.

Usage

DIC(object, ...)

## S3 method for class 'edina'
DIC(object, ...)

Arguments

- object: An edina object
- ...: Not used.

Value

The DIC value of the given model.

DIC Computation Procedure

\[ DIC = -2 \left( \log p \left( y|\hat{\theta} \right) - 2 \left( \log p \left( y|\hat{\theta} \right) - \frac{1}{N} \sum_{n=1}^{N} \log p \left( y|\theta_n \right) \right) \right) \]

See Also

PPP.edina(), BIC.edina()
Arguments

- `data`: Binary responses to assessments in matrix form with dimensions $N \times J$.
- `k`: Number of Attribute Levels as a positive integer.
- `burnin`: Number of Observations to discard on the chain.
- `chain_length`: Length of the MCMC chain

Value

An `edina` object that contains:

- `coefficients`: Estimated coefficients of the model fit
- `loglike_summed`: Summed log-likelihood
- `loglike_pmean`: Mean of log-likelihood
- `pi_classes`: Latent classes
- `avg_q`: Estimated Averaged Q Matrix
- `est_q`: Estimated Dichotomous Q Matrix
- `or_tested`: Odds Ratio used in the model selection.
- `sample_or`: Odds Ratio for the sample.
- `n`: Number of Observations
- `j`: Number of Items
- `k`: Number of Traits
- `burnin`: Amount of iterations to discard
- `chain_length`: Amount of iterations to retain.
- `timing`: Duration of the run
- `dataset_name`: Name of the data set used in estimation.

See Also

`auto_edina()`, `summary.edina()`, `print.edina()`

Examples

if(requireNamespace("simcdm", quietly = TRUE)) {

  # Set a seed for reproducibility
  set.seed(1512)

  # Setup data simulation parameters
  N = 1  # Number of Examinees / Subjects
  J = 10  # Number of Items
  K = 2  # Number of Skills / Attributes

  # Note:
  # Sample size and attributes have been reduced to create a minimally
  # viable example that can be run during CRAN's automatic check.
# Please make sure to have a larger sample size...

# Assign slipping and guessing values for each item
ss = gs = rep(.2, J)

# Simulate an identifiable Q matrix
Q = simcdm::sim_q_matrix(J, K)

# Simulate subject attributes
subject_alphas = simcdm::sim_subject_attributes(N, K)

# Simulate items under the DINA model
items_dina = simcdm::sim_dina_items(subject_alphas, Q, ss, gs)

# Compute the edina model
edina_model = edina(items_dina, k = K)

# Display results
edina_model

# Provide a summary overview
summary(edina_model)

## S3 method for class 'q_matrix'
extract_q_matrix(x, ...)

## S3 method for class 'edina'
extract_q_matrix(x, binary = TRUE, ...)

## Default S3 method:
extract_q_matrix(x, ...)

Arguments

x An edina or q_matrix object
...

Additional parameters

Description

Given a modeling object, extract the Q Matrix

Usage

extract_q_matrix(x, ...)

# Please make sure to have a larger sample size...

# Assign slipping and guessing values for each item
ss = gs = rep(.2, J)

# Simulate an identifiable Q matrix
Q = simcdm::sim_q_matrix(J, K)

# Simulate subject attributes
subject_alphas = simcdm::sim_subject_attributes(N, K)

# Simulate items under the DINA model
items_dina = simcdm::sim_dina_items(subject_alphas, Q, ss, gs)

# Compute the edina model
edina_model = edina(items_dina, k = K)

# Display results
edina_model

# Provide a summary overview
summary(edina_model)

extract_q_matrix Extract Q Matrix

Description

Given a modeling object, extract the Q Matrix

Usage

extract_q_matrix(x, ...)

## S3 method for class 'q_matrix'
extract_q_matrix(x, ...)

## S3 method for class 'edina'
extract_q_matrix(x, binary = TRUE, ...)

## Default S3 method:
extract_q_matrix(x, ...)

Arguments

x An edina or q_matrix object
...

Additional parameters
binary

Boolean to indicate whether the $Q$ matrix is shown in dichotomous form or in an estimated form.

Value

A matrix that is either dichotomous or estimated depending on the value of the binary parameter.

See Also

q_matrix(), as_q_matrix(), edina(), auto_edina()

Examples

# Q matrix values
x = matrix(c(1, 0, 0, 1), nrow = 2)

# Show Q matrix structure
Q = q_matrix(x)

# Retrieve Q matrix
extract_q_matrix(Q)

model_selection_graph

View Model Selection Statistics Across Models

Description

Displays information about the value of each model information criterion for a given model across the dimensions the Q matrix is estimated.

Usage

model_selection_graph(x, ...)

Arguments

x

An auto_edina or auto_errum object.

... Not used

Value

A ggplot2 object

See Also

autoplot.auto_edina()
**parameter_evolution_graph**

*View Slipping and Guessing Parameter Changes Across Models*

**Description**

Displays the slipping and guessing parameter changes for each model across the dimensions the Q matrix is estimated.

**Usage**

```r
parameter_evolution_graph(x, ...) 
```

**Arguments**

- **x**: An auto_edina or auto_errum object.
- **...**: Not used

**Value**

A ggplot2 object

**See Also**

`autoplot.auto_edina()`

---

**PPP**

*Posterior Predictive Probabilities (PPPs)*

**Description**

Computes posterior predictive probabilities (PPPs) based on the odds ratios for each pair of items.

**Usage**

```r
PPP(object, ...) 
```

**Arguments**

- **object**: An edina object
- **...**: Not used.
- **alpha**: Defining region to indicate the level of extremeness the data must before the model is problematic.
Details

PPPs that smaller than 0.05 or greater than 0.95 tend to be extreme and evidence of misfit. As a result, this is more of a heuristic metric.

Value

The PPP value given the specified alpha value.

PPP Computation Procedure

1. simulate observed responses $\mathbf{Y}^{(r)}$ using model parameters from iteration $r$ of the MCMC sampler
2. computing the odds ratio for each pair of items at iteration $r$ as

$$OR^{(r)} = \frac{n^{(r)}_{11} n^{(r)}_{00}}{n^{(r)}_{10} n^{(r)}_{01}}$$

, where $n^{(r)}_{11}$ is the frequency of ones on both variables at iteration $r$, $n^{(r)}_{10}$ is the frequency of ones on the first item and zeros on the second at iteration $r$, etc.; and
3. computing PPPs for each item pair as the proportion of generated $OR^{(r)}$'s that exceeded elements of the observed odds ratios.

---

**Description**

Custom print method for displaying the results of the Auto EDINA method.

**Usage**

```r
## S3 method for class 'auto_edina'
print(x, ...)
```

**Arguments**

- `x` An auto_edina object
- `...` Additional values passed onto the print.data.frame method.

**Value**

None.

The function provides a side-effect of displaying the overview of computed results across all models estimated.
### print.edina

**Printing out the EDINA Object**

**Description**

Custom print method for computing the EDINA.

**Usage**

```r
## S3 method for class 'edina'
print(x, binary = FALSE, ...)
```

**Arguments**

- `x`: An edina object
- `binary`: Boolean to indicate whether the Q matrix is shown in dichotomous form or in an estimated form.
- `...`: Additional methods passed onto the print.matrix method.

**Value**

None.

The function provides a side-effect of displaying the overview of the model estimated.

### print.q_matrix

**Printing out a Q Matrix Object**

**Description**

Custom print method for the Q Matrix Object.

**Usage**

```r
## S3 method for class 'q_matrix'
print(x, ...)
```

**Arguments**

- `x`: An q_matrix object
- `...`: Additional methods passed onto the print.matrix method.

**Value**

An invisible matrix without the q_matrix class displayed as a part of the output displayed.
See Also

q_matrix(), as_q_matrix()

Examples

# Q matrix values
x = matrix(c(1, 0, 0, 1), nrow = 2)

# Show Q matrix structure
q_matrix(x)
print.summary_edina  Printing out the Summary EDINA Object

Description

Custom print method for displaying the EDINA model summary information.

Usage

```r
## S3 method for class 'summary_edina'
print(x, binary = FALSE, ...)
```

Arguments

- `x`: A summary_edina object
- `binary`: Boolean to indicate whether the \( Q \) matrix is shown in dichotomous form or in an estimated form.
- `...`: Passed onto the print.data.frame method.

Value

None.

The function provides a side-effect of displaying the overview of the model estimated.

q_graph  Graph \( Q \) Matrix

Description

Provides a heatmap approach to showing the estimated binary or averaged values of the \( Q \) Matrix.

Usage

```r
q_graph(x, ...)
```

```r
## S3 method for class 'auto_edina'
q_graph(x, binary = TRUE, ic = c("ppp", "bic", "dic"), ...)
```

```r
## S3 method for class 'edina'
q_graph(x, binary = TRUE, ...)
```

```r
## S3 method for class 'matrix'
q_graph(x, ...)
```

```r
## S3 method for class 'q_matrix'
q_graph(x, ...)
```
Arguments

x 
Either an edina, auto_edina, or q_matrix object.

... Additional parameters not used

binary 
Boolean to indicate if a classified Q (dichotomous by decision rule) or an estimate Q (non-dichotomous) or should be shown. Default: TRUE.

ic 
Information criterion name. Default "ppp".

Value

A ggplot2 object with a heatmap overview of the estimated Q matrix.

Examples

```r
q = q_matrix(matrix(c(1, 0, 1, 1, 0, 1), ncol = 3))
q_graph(q)
```

---

### `q_matrix`

**Create a Q Matrix Object**

**Description**

Provides a way to create an object as a "q_matrix".

**Usage**

```r
q_matrix(x)
```

**Arguments**

x 
Either a data.frame or matrix.

**Value**

A q_matrix object.

**See Also**

`as_q_matrix()`
Examples

```r
# Q matrix values
x = matrix(c(1, 0, 0, 1), nrow = 2)

# Q matrix wrapper
q_mat = q_matrix(x)

# Data Frame encoding of Q
q_df = data.frame(
  k1 = c(1, 0),
  k2 = c(0, 1)
)

# Create a Q matrix
q_mat = q_matrix(q_df)
```

---

**summary.auto_edina**

*Summarize auto_edina model data*

**Description**

Custom method for displaying the results of the `auto_edina`.

**Usage**

```r
## S3 method for class 'auto_edina'
summary(object, ...)
```

**Arguments**

- `object` An `auto_edina` object
- `...` Not used.

**Value**

The original `auto_edina` object with an added class of `summary.auto_edina`. 
Summary

Summarize the EDINA Object

Description

Provide a more detailed view inside of edina model object.

Usage

```r
## S3 method for class 'edina'
summary(object, alpha = 0.05, ...)
```

Arguments

- `object`: An edina object
- `alpha`: Defining region to indicate the level of extremeness the data must before the model is problematic.
- `...`: Not used.

Value

A summary object that includes everything in the original `edina()` object and:

- `model_fit`: Matrix of model fit summary statistics.
- `alpha`: Alpha-value used to compute `PPP()`s.
Index

as_q_matrix, 2
as_q_matrix(), 11, 15, 17
auto_edina, 4
auto_edina(), 3, 9, 11
autoplot.auto_edina, 3
autoplot.auto_edina(), 5, 11, 12
best_model, 6
best_model(), 3, 5
BIC.edina, 7
BIC.edina(), 7, 8
DIC, 8
DIC.edina(), 7
edina, 8
edina(), 11, 19
extract_q_matrix, 10
model_selection_graph, 11
model_selection_graph(), 3, 5
parameter_evolution_graph, 12
parameter_evolution_graph(), 3, 5
PPP, 12
PPP(), 19
PPP.edina(), 7, 8
print.auto_edina, 13
print.edina, 14
print.edina(), 9
print.q_matrix, 14
print.summary_auto_edina, 15
print.summary_edina, 16
q_graph, 16
q_matrix, 17
q_matrix(), 3, 11, 15
summary.auto_edina, 18
summary.edina, 19
summary.edina(), 9