Package ‘nlsem’

May 1, 2017

Version 0.8

Date 2017-05-01

Title Fitting Structural Equation Mixture Models

Depends R (>= 3.1.0), stats, graphics, orthopolynom

Imports nlme, lavaan, gaussquad, mvtnorm

Description Estimation of structural equation models with nonlinear effects and underlying nonnormal distributions.

License GPL (>= 2)

NeedsCompilation no

Author Nora Umbach [aut, cre],
Katharina Naumann [aut],
David Hoppe [aut],
Holger Brandt [aut],
Augustin Kelava [ctb],
Bernhard Schmitz [ctb]

Maintainer Nora Umbach <nora.umbach@web.de>

Repository CRAN

Date/Publication 2017-05-01 10:31:57 UTC

R topics documented:

`nlsem-package` *Fitting structural equation mixture models*

Description

Estimation of structural equation models with nonlinear effects and underlying nonnormal distributions.

Details

This is a package for estimating nonlinear structural equation mixture models using an expectation-maximization (EM) algorithm. Four different approaches are implemented. Firstly, the Latent Moderated Structural Equations (LMS) approach (Klein & Moosbrugger, 2000) and the Quasi-Maximum Likelihood (QML) approach (Klein & Muthen, 2007), which allow for two-way interaction and quadratic terms in the structural model. Due to the nonlinearity, the latent criterion variables cannot be assumed to be normally distributed. Therefore, the latent criterins's distribution is approximated with a mixture of normal distributions in LMS. Secondly, the Structural Equation finite Mixture Model (STEMM or SEMM) approach (Jedidi, Jagpal & DeSarbo, 1997), which uses mixtures to model latent classes. In this way it can deal with heterogeneity in the sample or nonlinearity and nonnormality of the latent variables and their indicators. And thirdly, a combination of these two approaches, the Nonlinear Structural Equation Mixture Model (NSEMM) approach (Kelava, Nagengast & Brandt, 2014). Here, interaction and quadratic terms as well as latent classes can be modeled.

The models can be specified with `specify_sem`. Depending on the specification of `interaction` and the number of latent classes (`num.classes`) the returned object will be of class `singleClass`, `semm`, or `nsemm`. Each of these can be estimated using `em` and models of type `singleClass` can additionally be fitted with the function `qml`.

Future Features

- NSEMM, LMS and QML for more than one latent endogenous variable.
- Parameter standardization.

References

Jedidi, K., Jagpal, H. S., & DeSarbo, W. S. (1997). STEMM: A General Finite Mixture Structural Equation Model, *Journal of Classification, 14*, 23–50. doi:http://dx.doi.org/10.1007/s003579900002

Kelava, A., Nagengast, B., & Brandt, H. (2014). A nonlinear structural equation mixture modeling approach for non-normally distributed latent predictor variables. *Structural Equation Modeling, 21*, 468-481. doi:http://dx.doi.org/10.1080/10705511.2014.915379

Klein, A. &, Moosbrugger, H. (2000). Maximum likelihood estimation of latent interaction effects with the LMS method. *Psychometrika, 65*, 457–474. doi:http://dx.doi.org/10.1007/bf02296338

Klein, A. &, Muthen, B. O. (2007). Quasi-Maximum Likelihood Estimation of Structural Equation Models With Multiple Interaction and Quadratic Effects. *Multivariate Behavior Research, 42*, 647–673. doi:http://dx.doi.org/10.1080/00273170701710205

`anova` *Anova Tables*

Description

Calculate likelihood ratio tests to compare two or more structural equation models.

Usage

```
## S3 method for class 'emEst'
anova(object, ..., test = c("Chisq", "none"))
```

Arguments

`object`	estimated structural equation model of class `emEst`.
`...`	additional objects of the same type.
`test`	a character string, (partially) matching one of 'Chisq' or 'none'. Should the p-values of the chi-square distributions be reported?

Value

Returns an object of class codeanova. These objects represent analysis-of-variance and analysis-of-deviance tables. It is not implemented for a single argument.

`as.data.frame` *Coerce to a Data Frame*

Description

Function to coerce an object created with `specify_sem` to a data frame.

Usage

```
## S3 method for class 'singleClass'
as.data.frame(x, ...)
## S3 method for class 'semm'
as.data.frame(x, ...)
## S3 method for class 'nsemm'
as.data.frame(x, ...)
```

Arguments

x	structural equation model of class singleClass, semm, or nsemm.
...	additional arguments.

Value

Returns a data frame with first column label and one column for each latent class labeled class1, class2 and so on.

See Also

specify_sem, create_sem

Examples

```
# specify model
model <- specify_sem(num.x = 6, num.y = 3, num.xi = 2, num.eta = 1,
  xi = "x1-x3,x4-x6", eta = "y1-y3", interaction = "eta1~xi1:xi2")

# coerce to data frame
as.data.frame(model)
```

---

australia *Data from Australian subset of PISA 2009 data*

Description

The data stem from the large-scale assessment study PISA 2009 (Organisation for Economic Co-Operation and Development, 2010) where competencies of 15-year-old students in reading, mathematics, and science are assessed using nationally representative samples in 3-year cycles. In this example, data from the student background questionnaire from the Australian sample of PISA 2009 were used. Only data of students with complete responses (N = 1,069) were considered.

Usage

```
data(jordan)
```

Format

A data frame of nine variables and 1,069 observations:

x1 indicator for reading attitude, mean of items ST24Q04, ST24Q09, ST24Q01, and ST24Q03.

x2 indicator for reading attitude, mean of items ST24Q02, ST24Q05, ST24Q07, and ST24Q06.

x3 indicator for reading attitude, mean of items ST24Q08, ST24Q10, and ST24Q11.

x4 indicator for online activities, mean of items ST26Q02, ST26Q07, and ST26Q04.

x5 indicator for online activities, mean of items ST26Q03, and ST26Q06.

x6 indicator for online activities, mean of items ST26Q01, and ST26Q05.

y1 indicator for reading skill, mean of items R06, R102, and R219.

y2 indicator for reading skill, mean of items R220, R414, and R447.

y3 indicator for reading skill, mean of items R452 and R458.

Source

Organisation for Economic Co-Operation and Development (2010). *Pisa 2009 results: What students know and can do - Student performance in reading, mathematics and science* (Tech. Rep.). Paris, France. Obtained from: http://www.oecd.org/pisa/pisaproducts/pisa2009database-downloadabledata.htm

Examples

```
data(australia)
```

`count_free_parameters` *Count free parameters of structural equation model*

Description

Counts free parameters of a structural equation model of class `singleClass`, `semm`, or `nsemm`.

Usage

```
count_free_parameters(model)
```

Arguments

`model`	A model created with `specify_sem`.

Value

Returns the number of free parameters in the `model` (numeric).

Examples

```
model <- specify_sem(num.x = 4, num.y = 2, num.xi = 2, num.eta = 1,
  xi = "x1-x2,x3-x4", eta = "y1-y2", interaction = "eta1~xi1:xi2")
count_free_parameters(model)
```

create_sem *Create a structural equation model from a data frame*

Description

Create model matrices from a data frame with columns `label` (for parameter labels) and `class1` to `classX`.

Usage

```
create_sem(dat)
```

Arguments

`dat`	data frame with first column `label` and one column for each latent class labeled `class1`, `class2` and so on. See Details.

Details

Labels in column `label` need to be labeled in a certain way. Labels can be looked up by creating an object with `specify_sem` and then transforming it to a data frame with `as.data.frame`. See examples below.

Value

Gives back an object of class `singleClass`, `semm`, or `nsemm` which can be fitted using `em`.

See Also

`specify_sem`

Examples

```
# specify model
model <- specify_sem(num.x = 4, num.y = 1, num.xi = 2, num.eta = 1,
  xi = "x1-x2,x3-x4", eta = "y1", interaction = "eta1~xi1:xi2")
# create data frame
dat <- as.data.frame(model)
# recreate model
create_sem(dat)
```

em *Maximum likelihood estimation of structural equation mixture models*

Description

Fits a structural equation model with latent interaction effects using mixture approaches (LMS, SEMM, NSEMM).

Usage

```
em(model, data, start, qml = FALSE, verbose = FALSE, convergence = 1e-02,
   max.iter = 100, m = 16, optimizer = c("nlminb", "optim"),
   max.mstep = 1, max.singleClass = 1, neg.hessian = TRUE, ...)
```

Arguments

`model`	a specified structural equation model of class `singleClass`, `semm`, or `nsemm`.
`data`	the data the model should be fitted to. Data needs to be a matrix and variables need to be in the order x1, x2, ..., y1, y2, ... as specified in `specify_sem`. Data matrix needs no column names (will be ignored anyways).
`start`	starting values for parameters.
`qml`	logical. Indicating if QML estimation should be used instead of LMS for estimation of nonlinear effects. Defaults to FALSE. QML is much faster, though.
`verbose`	if output of EM algorithm should be shown during fitting.
`convergence`	convergence threshold.
`max.iter`	maximum number of iterations before EM algorithm stops.
`m`	number of nodes for Hermite-Gaussian quadrature. Defaults to 16. See Datails.
`optimizer`	which optimizer should be used in maximization step of EM algorithm: `nlminb` or `optim`.
`max.mstep`	maximum iteration steps the optimizer should use in its mstep during one EM iteration. Defaults to 1.
`max.singleClass`	maximum iteration steps for singleClass model inside of NSEMM model. Defaults to 1 (and should only be changed for valid reasons).
`neg.hessian`	should negative Hessian be calculated in last step of iteration.
`...`	additional arguments. See Details.

Details

`em` can be used to estimate parameters for structural equation mixture models with latent interaction effects with an EM algorithm. The maximization step of the EM algorithm can use two different optimizers: `optim` or `nlminb`. Default is `nlminb`.

Additional arguments can be passed to ... for these optimizers. See documentation for `optim` and `nlminb`.

The LMS approach (Klein & Moosbrugger, 2000) uses Hermite-Gauss quadrature for numerical approximation. The nodes used in this approximation need to be prespecified by the user. The more nodes are used the better the numerical approximation but also the slower the calculations.

Value

An object of class `emEst` that consists of the following components:

`model.class`	class of model that was fitted, can be `singleClass`, `semm`, or `nsemm`.
`coefficients`	estimated parameters.
`objective`	final loglikelihood obtained with EM algorithm.
`em_convergence`	yes or no. Did EM algorithm converge?
`Hessian`	Hessian matrix for final parameter estimation.
`loglikelihoods`	loglikelihoods obtained during each iteration of EM algorithm.
`info`	list of number of exogenous (`num.xi`) and endogenous (`num.eta`) variables and of indicators (`num.x` and `num.y`). Corresponds to specifications given to `specify_sem` when specifiying structural equation model.

References

Jedidi, K., Jagpal, H. S., & DeSarbo, W. S. (1997). STEMM: A General Finite Mixture Structural Equation Model, *Journal of Classification, 14*, 23–50. doi:http://dx.doi.org/10.1007/s003579900002

Kelava, A., Nagengast, B., & Brandt, H. (2014). A nonlinear structural equation mixture modeling approach for non-normally distributed latent predictor variables. *Structural Equation Modeling, 21*, 468-481. doi:http://dx.doi.org/10.1080/10705511.2014.915379

Klein, A. &, Moosbrugger, H. (2000). Maximum likelihood estimation of latent interaction effects with the LMS method. *Psychometrika, 65*, 457–474. doi:http://dx.doi.org/10.1007/bf02296338

See Also

`specify_sem`

Examples

```
###### Example for SEMM ######
# load data
data("PoliticalDemocracy", package = "lavaan")
dat <- as.matrix(PoliticalDemocracy[ ,c(9:11,1:8)])

# specify model of class SEMM
model <- specify_sem(num.x = 3, num.y = 8, num.xi = 1, num.eta = 2,
  xi = "x1-x3", eta = "y1-y4,y5-y8", rel.lat = "eta1~xi1,eta2~xi1,eta2~eta1",
  num.classes = 2, constraints = "direct1")

# fit model
set.seed(911)
start <- runif(count_free_parameters(model))
```

```
## Not run:
res <- em(model, dat, start, convergence = 0.1, max.iter = 200)
summary(res)
plot(res)

## End(Not run)

###### Example for LMS ######
model <- specify_sem(num.x = 11, num.y = 4, num.xi = 2, num.eta = 1,
  xi = "x1-x5,x6-x11", eta = "y1-y4", interaction = "eta1~xi1:xi2")

data("jordan")

set.seed(110)
start <- runif(count_free_parameters(model))
## Not run:
res <- em(model, jordan, start, convergence=1, verbose=TRUE)
summary(res)
plot(res)

## End(Not run)

###### Example using lavaan syntax ######
lav.model <- '
  eta =~ y1 + y2 + y3 + y4
  xi1 =~ x1 + x2 + x3 + x4 + x5
  xi2 =~ x6 + x7 + x8 + x9 + x10 + x11

  eta ~ xi1 + xi2 + xi1:xi2 + xi1:xi1'

model <- lav2nlsem(lav.model)

data("jordan")

set.seed(1118)
start <- runif(count_free_parameters(model))
## Not run:
res <- em(model, jordan, start, convergence=1, verbose=TRUE)

## End(Not run)
```

`fill_model` *Fills an empty structural equation model with parameters*

Description

Creates a model of the same class as `model` and puts `parameters` where `model` has `NA`'s.

Usage

```
fill_model(model, parameters)
```

Arguments

`model`	a model created by `specify_sem` or `create_sem`.
`parameters`	numeric vector with length of number of free parameters in `model`. See `count_free_parameters`.

Value

Gives back an object of class `singleClass`, `semm`, or `nsemm`.

See Also

`specify_sem`, `create_sem`, `count_free_parameters`

Examples

```
# specify model
model <- specify_sem(num.x = 4, num.y = 1, num.xi = 2, num.eta = 1,
  xi = "x1-x2,x3-x4", eta = "y1", interaction = "eta1~xi1:xi2")
pars <- runif(count_free_parameters(model))
fill_model(model, parameters = pars)
```

`jordan` *Data from Jordan subset of PISA 2006 data*

Description

The data stem from the large-scale assessment study PISA 2006 (Organisation for Economic Co-Operation and Development, 2009) where competencies of 15-year-old students in reading, mathematics, and science are assessed using nationally representative samples in 3-year cycles. In this example, data from the student background questionnaire from the Jordan sample of PISA 2006 were used. Only data of students with complete responses to all 15 items (N = 6,038) were considered.

Usage

```
data(jordan)
```

Format

A data frame of fifteen variables and 6,038 observations:

x1 indicator for enjoyment of science, item ST16Q01: I generally have fun when I am learning <broad science> topics.

x2 indicator for enjoyment of science, item ST16Q02: I like reading about <broad science>.

x3 indicator for enjoyment of science, item ST16Q03: I am happy doing <broad science> problems.

x4 indicator for enjoyment of science, item ST16Q04: I enjoy acquiring new knowledge in <broad science>.

x5 indicator for enjoyment of science, item ST16Q05: I am interested in learning about <broad science>.

x6 indicator for academic self-concept in science, item ST37Q01: I can easily understand new ideas in <school science>.

x7 indicator for academic self-concept in science, item ST37Q02: Learning advanced <school science> topics would be easy for me.

x8 indicator for academic self-concept in science, item ST37Q03: I can usually give good answers to <test questions> on <school science> topics.

x9 indicator for academic self-concept in science, item ST37Q04: I learn <school science> topics quickly.

x10 indicator for academic self-concept in science, item ST37Q05: <School science> topics are easy for me.

x11 indicator for academic self-concept in science, item ST37Q06: When I am being taught <school science>, I can understand the concepts very well.

y1 indicator for career aspirations in science, item ST29Q01: I would like to work in a career involving <broad science>.

y2 indicator for career aspirations in science, item ST29Q02: I would like to study <broad science> after <secondary school>.

y3 indicator for career aspirations in science, item ST29Q03: I would like to spend my life doing advanced <broad science>.

y4 indicator for career aspirations in science, item ST29Q04: I would like to work on <broad science> projects as an adult.

Source

Organisation for Economic Co-Operation and Development (2009). *Pisa 2006: Science competencies for tomorrow's world* (Tech. Rep.). Paris, France. Obtained from: http://www.oecd.org/pisa/pisaproducts/database-pisa2006.htm

Examples

```
data(jordan)
```

`lav2nlsem` *Create a structural equation model from lavaan syntax*

Description

Create model matrices from a string specifying a structural equation model in lavaan syntax.

Usage

```
lav2nlsem(model, constraints=c("indirect", "direct1", "direct2"),
          class.spec="class")
```

Arguments

`model`	A description of the user-specified model. The model is described using the lavaan model syntax. See Details in `?model.syntax` in lavaan for more information.
`constraints`	which should be set for a model with more than one latent class. See Details in `?specify_sem`.
`class.spec`	String used to specify latent classes. Can be any string e.g. 'class', 'mixture', etc. Default is 'class'.

Details

nlsem can only fit a certain group of models and it is only feasible to specify models in the lavaan syntax that can be fitted with nlsem; that means models with latent variables and latent interactions only.

Parameter restrictions in lavaan style can be used to some extent; meaning parameters can be fixed to a certain value with `1*x1`. Equality restrictions are handled via the `constraints` argument and will be ignored in the lavaan syntax.

Value

Gives back an object of class `singleClass`, `semm`, or `nsemm` which can be fitted using `em`.

References

Rosseel, Y. (2012). lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software, 48(2)*, 1 - 36. doi:http://dx.doi.org/10.18637/jss.v048.i02

See Also

`specify_sem`, `create_sem`

Examples

```
# create model with three latent classes
lav.model <- '
          class: 1
          eta =~ y1 + y2 + y3 + y4
          xi1 =~ x1 + x2 + x3 + x4 + x5
          xi2 =~ x6 + x7 + x8 + x9 + x10 + x11

          eta ~ xi1 + xi2 + xi1:xi1

          class: 2
          eta =~ y1 + y2 + y3 + y4
```

```
          xi1 =~ x1 + x2 + x3 + x4 + x5
          xi2 =~ x6 + x7 + x8 + x9 + x10 + x11

          eta ~ xi1 + xi2 + xi1:xi2 + xi1:xi1

          class: 3
          eta =~ y1 + y2 + y3 + y4
          xi1 =~ x1 + x2 + x3 + x4 + x5
          xi2 =~ x6 + x7 + x8 + x9 + x10 + x11

          eta ~ xi1 + xi2 + xi1:xi2'

model <- lav2nlsem(lav.model, constraints = "direct1", class.spec = "class")
```

`qml` *Quasi-maximum likelihood estimation of a nonlinear structural equation model*

Description

Fits a structural equation model with latent interaction effects using Quasi-maximum likelihood estimation.

Usage

```
qml(model, data, start, max.iter = 150, optimizer = c("nlminb",
    "optim"), neg.hessian = TRUE, ...)
```

Arguments

`model`	a specified structural equation model of class `singleClass`.
`data`	the data the model should be fitted to. Data needs to be a matrix and variables need to be in the order x1, x2, ..., y1, y2, ... as specified in `specify_sem`. Data matrix needs no column names (will be ignored anyways).
`start`	starting values for parameters.
`max.iter`	maximum number of iterations for optimizer.
`optimizer`	which optimizer should be used for maximization of parameters: `nlminb` or `optim`.
`neg.hessian`	should negative Hessian be calculated.
`...`	additional arguments. See Details.

Details

Additional arguments can be passed to ... for these optimizers. See documentation for `optim` and `nlminb`.

Quasi-maximum likelihood (QML) estimation is in principle a faster version for LMS, but might be less accurate for normal data. For practical purposes differences are negligible, though. For nonnormal data QML outperforms LMS.

Value

An object of class `qmlEst` that consists of the following components:

- `model.class` — class of model that was fitted. Will always be `singleClass`.
- `coefficients` — estimated parameters.
- `objective` — final loglikelihood obtained with EM algorithm.
- `convergence` — convergence code for optimizer. See documentation for `optim` and `nlminb`.
- `Hessian` — negative Hessian matrix for final parameter estimation.
- `info` — list of number of exogenous (`num.xi`) and endogenous (`num.eta`) variables and of indicators (`num.x` and `num.y`). Corresponds to specifications given to `specify_sem` when specifiying structural equation model.

References

Klein, A. &, Muthen, B. O. (2007). Quasi-Maximum Likelihood Estimation of Structural Equation Models With Multiple Interaction and Quadratic Effects. *Multivariate Behavior Research, 42*, 647–673. doi:http://dx.doi.org/10.1080/00273170701710205

See Also

`specify_sem`

Examples

```
# specify model of class singleClass
sc <- specify_sem(num.x=4, num.y=2, num.xi=2, num.eta=1, xi="x1-x2,x3-x4",
                  eta="y1-y2", interaction="eta1~xi1:xi2")

# simulate data
pars.orig <- c(0.6, 0.7,                # Lx
               0.8,                     # Ly
               0.2, 0.4,                # G
               0.25, 0.25, 0.25, 0.25,  # Td
               0.25, 0.25,              # Te
               0.2,                     # Psi
               0.49, 0.235, 0.64,       # Phi
               0, 0,                    # nu.x
               0,                       # nu.x
               1,                       # alpha
               1, 1,                    # tau
               0.7                      # Omega
              )

dat <- simulate(sc, parameters=pars.orig, seed=81)

# fit model
set.seed(1609)
start <- runif(count_free_parameters(sc))
## Not run:
qml1 <- qml(sc, dat, start)
```

```
summary(qml1)

## End(Not run)
```

---

simulate *Simulate data from a structural equation model*

Description

Simulate data from a structural equation mixture model.

Usage

```
## S3 method for class 'singleClass'
simulate(object, nsim = 1, seed = NULL, n = 400, m = 16, parameters, ...)
## S3 method for class 'semm'
simulate(object, nsim = 1, seed = NULL, n = 400, parameters, ...)
## S3 method for class 'nsemm'
simulate(object, nsim = 1, seed = NULL, n = 400, m = 16, parameters, ...)
```

Arguments

object	structural equation model of class singleClass, semm, or nsemm.
parameters	'true' parameters which should be used to simulate data.
nsim	number of response vectors to simulate. Defaults to 1.
seed	set seed. Default is NULL.
n	data for how many observations should be simulated.
m	number of nodes for Hermite-Gaussian quadrature. Only needed for singleClass and nsemm.
...	additional arguments.

Value

Returns a matrix with n rows and as many columns as indicators are entered into the model.

Examples

```
# specify model
model <- specify_sem(num.x = 6, num.y = 3, num.xi = 2, num.eta = 1,
  xi = "x1-x3,x4-x6", eta = "y1-y3", interaction = "eta1~xi1:xi2")

# original parameters
pars.orig <- c(.6, .5, .4, .5, .4, .6, .5, .2, .6, .7, .3, .2, .5,
               .7, .3, .4, .6, .2, .3, .4, .6, .2, .2, .2, .2, .3,
               .3, 1, 0, 0, .8)

# simulate data from model
dat <- simulate(model, parameters = pars.orig)
```

specify_sem *Specify a structural equation model*

Description

Specify a structural equation model with constraints.

Usage

```
specify_sem(num.x, num.y, num.xi, num.eta, xi, eta,
            constraints = c("indirect", "direct1", "direct2"),
            num.classes = 1, rel.lat = "default", interaction = "none")
```

Arguments

num.x	number of observed variables for xi.
num.y	number of observed variables for eta.
num.xi	number of latent exogenous variabeles.
num.eta	number of latent endognous variables.
xi	which observed variables are indicators for which exogenous variable. See Details.
eta	which observed variables are indicators for which endogenous variable. See Details.
constraints	which should be set for a model with more than one latent class. See Details.
num.classes	number of latent classes.
interaction	define which interaction terms should be included. Default is 'none'. See Details for how to enter interaction terms.
rel.lat	define relations between latent variables. Influences Beta and Gamma matrices. For 'defaults' and how to define see Details.

Details

The notation for the `matrices` given back by `specify_sem` follows typical notation used in structural equation modeling. The notation, of course, may vary dependingly. Therefore, here are examples for typical structural equation models with the notation used by `specify_sem` (in matrix notation):

Structural model for LMS, QML (nonlinear SEM), and NSEMM (nonlinear SEM with latent classes):

$$\boldsymbol{\eta} = \boldsymbol{\alpha} + \boldsymbol{\Gamma\xi} + \boldsymbol{\xi}'\boldsymbol{\Omega\xi} + \boldsymbol{\zeta}$$

Structural model for SEMM (linear SEM with latent classes):

$$\boldsymbol{B\eta} = \boldsymbol{\alpha} + \boldsymbol{\Gamma\xi} + \boldsymbol{\zeta}$$

Measurement model:

$$\mathbf{x} = \boldsymbol{\nu}_x + \boldsymbol{\lambda}_x \boldsymbol{\xi} + \boldsymbol{\delta}$$

$$\mathbf{y} = \boldsymbol{\nu}_y + \boldsymbol{\lambda}_y \boldsymbol{\eta} + \boldsymbol{\varepsilon}$$

Which indicators belong to which latent variable is defined by `xi` and `eta`. Must be specified in the following way: `xi='x1-x2,x3-x4'` which means that variables `x1, x2` are indicators for `xi1` and `x3, x4` are indicators for `xi2`. And accordingly for the endogenous variables `eta`.

Interactions between latent exogenous variables are defined by `interaction='eta1~xi1:xi2,eta1~xi1:xi1'`. It is important to note, that interactions must always start with `xi1` and build from there. A definition like `interaction='eta1~xi1:xi2,eta1~xi2:xi3'` is not feasible and must be changed to `interaction='eta1~xi1:xi2,eta1~xi1:xi3'` (by simple switching `xi1` and `xi2` in one's definitions). `interaction` fills the $\mathbf{\Omega}$ matrix (see above) and must always be a triangular matrix where the lower triangle is filled with 0's (see Klein & Moosbrugger, 2000, for details).

`rel.lat` defines which latent variables influence each other. It must be defined like `rel.lat='eta1~xi1+xi2,eta2~eta1'`. Free parameters will be set accordingly in $\mathbf{B}$ and $\mathbf{\Gamma}$ matrices. When nothing is defined, $\mathbf{\Gamma}$ defaults to all `NA`s (which means all ξ's influence all η's) and $\mathbf{B}$ is an identity matrix.

Structural equation models with latent classes like SEMM and NSEMM can be used in two different approaches usually called direct and indirect. When constraints are set to `indirect` then parameters for the latent classes are constraint to be equal except for the parameters for the mixture distributions (τ's and Φ). In a direct approach, parameters for the latent classes are estimated independently. For `direct1` all parameters will be estimated independently for each latent class. For `direct2` it is assumed that the measurement model is equal for both groups and only the parameters for the mixtures and the structural model are estimated separately.

Value

An object of class `singleClass`, `semm`, or `nsemm` which can be used to estimate parameters using `em` that consists of the following components:

`matrices`	list of matrices specifying the structural equation model.
`info`	list of informations about structural equation model.

References

Jedidi, K., Jagpal, H. S., & DeSarbo, W. S. (1997). STEMM: A General Finite Mixture Structural Equation Model, *Journal of Classification, 14*, 23–50. doi:http://dx.doi.org/10.1007/s003579900002

Kelava, A., Nagengast, B., & Brandt, H. (2014). A nonlinear structural equation mixture modeling approach for non-normally distributed latent predictor variables. *Structural Equation Modeling, 21*, 468-481. doi:http://dx.doi.org/10.1080/10705511.2014.915379

Klein, A. &, Moosbrugger, H. (2000). Maximum likelihood estimation of latent interaction effects with the LMS method. *Psychometrika, 65*, 457–474. doi:http://dx.doi.org/10.1007/bf02296338

Klein, A. &, Muthen, B. O. (2007). Quasi-Maximum Likelihood Estimation of Structural Equation Models With Multiple Interaction and Quadratic Effects. *Multivariate Behavior Research, 42*, 647–673. doi:http://dx.doi.org/10.1080/00273170701710205

See Also

create_sem

Examples

```
# with default constraints
model <- specify_sem(num.x = 6, num.y = 3, num.xi = 2, num.eta = 1,
  xi = "x1-x3,x4-x6", eta = "y1-y3")

# create data frame
specs <- as.data.frame(model)
# and add custom constraints
constr <- c(1, NA, NA, 0, 0, 0, 0, 0, 0, 1, NA, NA, 1, NA, NA, NA, NA, 1, NA,
  0, 0, 0, 0, 0, 0, NA, 0, 0, 0, 0, 0, 0, NA, 0, 0, 0, 0, 0, 0, NA, 0, 0, 0,
  0, 0, 0, NA, 0, 0, 0, 0, 0, 0, NA, NA, 0, 0, 0, NA, 0, 0, 0, NA, NA, NA,
  NA, 0, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0, 0, 0, 0, NA, 0)
specs$class1 <- constr
# create model from data frame
model.custom <- create_sem(specs)
```

---

summary *Summarize output from EM algorithm for structural equation models*

Description

Summarize data from object obtained from em.

Usage

```
## S3 method for class 'emEst'
summary(object, print.likelihoods = FALSE, ...)
```

Arguments

object	estimated structural equation model of class emEst obtained from em.
print.likelihoods	if loglikelihoods for each iteration step of EM algorithm should be shown in summary output.
...	additional arguments.

Value

Returns a list that consists of the following components:

estimates	table of estimated parameters with standard errors and t and p values.
iterations	iterations needed by EM algorithm till convergence.
finallogLik	final loglikelihood obtained by EM algorithm.
loglikelihoods	table of loglikelohoods for each iteration of EM algorithm with difference and relative change.

Index