Package ‘SEMsens’

Type Package
Title A Tool for Sensitivity Analysis in Structural Equation Modeling
Version 1.0.2
Description Perform sensitivity analysis in structural equation modeling using meta-heuristic optimization methods (e.g., ant colony optimization and others). The references for the proposed methods are:
We also thank Dr. Krzysztof Socha for sharing his research on ant colony optimization algorithm with continuous domains and associated R code, which provided the base for the development of this package.
Imports lavaan, stats
Depends R (>= 3.5.0)
License GPL-3
Encoding UTF-8
LazyData true
Suggests knitr, rmarkdown
VignetteBuilder rmarkdown, knitr
RoxygenNote 7.1.2
NeedsCompilation no
Author Walter Leite [aut, cre],
Zuchao Shen [aut],
Gail Fish [ctb],
YongSeok Lee [ctb],
Sanaz Nazari [ctb],
Jia Quan [ctb],
Eric Wright [ctb],
Huibin Zhang [ctb]
Maintainer Walter Leite <walter.leite@coe.ufl.edu>
Description


Details

The package covers sensitivity analysis using ant colony optimization and other meta-heuristic optimization methods (in development) to automatically search a phantom variable, if there is any, that meets the optimization function. The current package includes three main functions and they are gen.sens.pars function that generates sensitivity parameters (running in background for the sa.aco function), sa.aco function that performs sensitivity analysis, and sens.tables function that summarizes sensitivity analysis results.

Author(s)

Walter Leite, Zuchao Shen

Maintainer: Walter Leite walter.leite@coe.ufl.edu (University of Florida)
**Description**

This function can generate a set of path coefficients from a phantom variable to variables in a structural equation model based on given distributions of the rank of optimization target (with probability of using a distribution based on its rank).

**Usage**

```r
gen.sens.pars(
  dist.mean,
  dist.rank,
  n.of.ants,
  nl,
  q = 1e-04,
  k = 500,
  xi = 0.5
)
```

**Arguments**

- `dist.mean` List of means - coordinates
- `dist.rank` Rank of the archived values of objective function
- `n.of.ants` Number of ants used in each iteration after the initialization of k converged sensitivity analysis models, default value is 10.
- `nl` Neighborhood of the search area
- `q` Locality of the search (0,1), default is 0.0001.
- `k` Size of the solution archive, default is 100.
- `xi` Convergence pressure (0, Inf), suggested: (0,1), default is 0.5.

**Value**

Generated sensitivity parameter values (i.e., a matrix with n.of.ants rows and n.of.sens.pars columns)

**References**


We thank Dr. Krzysztof Socha for providing us the original code (http://iridia.ulb.ac.be/supp/IridiaSupp2008-001/) for this function.
Examples

```r
k <- 50 # size of archive
# Generate dist.mean and dist.rank
dist.mean <- cbind(rnorm(k), rnorm(k), rnorm(k), rnorm(k), rnorm(k))
y <- rowMeans(dist.mean)
dist.rank <- rank(-y, ties.method = "random")
# set up neighborhood
nl <- matrix(NA, k, k-1)
for (i in 1:k){
  nl[i,] <- (1:k)[1:k != i]
}
my.sens.pars <- gen.sens.pars(dist.mean, dist.rank, n.of.ants = 10,
  nl, q = 0.0001, k = 50, xi = 0.50)
my.sens.pars
```

---

**sa.aco**  
*Sensitivity Analysis for Structural Equation Modeling Using colony optimization (ACO)*

**Description**

This function can perform sensitivity analysis for structural equation modeling using ant colony optimization (ACO).

**Usage**

```r
sa.aco(
  data = NULL,
  sample.cov,
  sample.nobs,
  model,
  sens.model,
  opt.fun,
  d = NULL,
  paths = NULL,
  verbose = TRUE,
  max.value = Inf,
  max.iter = 1000,
  e = 1e-10,
  n.of.ants = 10,
  k = 100,
  q = 1e-04,
  sig.level = 0.05,
  rate.of.conv = 0.1,
  measurement = FALSE,
  xi = 0.5,
```
Arguments

- **data**: The data set used for analysis.
- **sample.cov**: Covariance matrix for SEM analysis when data are not available.
- **sample.nobs**: Number of observations for covariance matrix.
- **model**: The analytic model of interest.
- **sens.model**: Sensitivity analysis model template for structural equation modeling with a phantom variable. This is the model of interest with a phantom variable and sensitivity parameters added. See examples provided.
- **opt.fun**: Customized or preset optimization function. The argument can be customized as a function, e.g., `opt.fun = quote(new.par$pvalue[paths] - old.par$pvalue[paths])`, where `new.par` and `old.par` are the parameter estimates from the sensitivity analysis and analytic models, respectively. When `opt.fun` is 1, the optimization function is the average departure of new estimate from the old estimate divided by the old estimate: `y = mean(abs(new.par$est[paths] - old.par$est[paths]))/mean(abs(old.par$est[paths]));` When `opt.fun` is 2, the optimization function is the standard deviation of deviation divided by the old estimate: `y = stats::sd(new.par$est[paths] - old.par$est[paths])/mean(abs(old.par$est[paths]));` When `opt.fun` is 3, the optimization function is the average p value changed or `y = mean(abs(new.par$pvalue[paths] - old.par$pvalue[paths]));` When `opt.fun` is 4, the optimization function is the average distance from significance level or `y = mean(abs(new.par$pvalue[paths] - rep(sig.level,length(paths))));` When `opt.fun` is 5, we assess the change of RMSEA or `y = abs(unname(lavaan::fitmeasures(new.out)$"rmsea") - unname(lavaan::fitmeasures(old.out)$"rmsea"));` When `opt.fun` is 6, we optimize how close RMSEA is to 0.05 or `y = 1/abs(unname(lavaan::fitmeasures(new.out)$"rmsea")) - 0.05`.
- **d**: Domains for initial sampling, default is c(-1,1) for all sensitivity analysis parameters. It can be specified as a list of ranges. For example, `d = list(-0.8, 0.8, -0.9, 0.9)` for two sampling domains with the first from -0.8 to 0.8 and the second from -0.9 to 0.9.
- **paths**: Paths in the model to be evaluated in a sensitivity analysis. If not specified, all paths will be evaluated. It can be specified in a numeric format or in a model format. For example, if we evaluate the changes (in p value or parameter estimation) for paths in an analytic model, we may specify paths in a model format, e.g., `paths = 'm ~ x y ~ x + m'`. Or, alternatively, as specify `paths = c(1:3)` if these paths present in line 1 to 3 in the sensitivity analysis model results.
- **verbose**: Print out evaluation process if TRUE, default is TRUE.
- **max.value**: Maximal value of optimization when used as the stopping criterion. Default is infinite.
- **max.iter**: Maximal number of function evaluations when used as the stopping criterion.
- **e**: Maximum error value used when solution quality used as the stopping criterion, default is 1e-10.
n.of.ants Number of ants used in each iteration after the initialization of k converged sensitivity analysis models, default value is 10.

k Size of the solution archive, default is 100.

q Locality of the search (0,1), default is 0.0001.

sig.level Significance level, default value is 0.05.

rate.of.conv The convergence rate threshold for sensitivity analysis models, default is .10.

measurement Logical. If TRUE, the argument paths will include measurement paths in the lavaanify format. Default is FALSE.

xi Convergence pressure (0, Inf), suggested: (0,1), default is 0.5.

seed Random seed if specified, default is NULL.

... Additional arguments from the lavaan package.

Value

Sensitivity analysis results, including the number of evaluations (n.eval), number of iterations (n.iter), the maximum value of the objective function (max.y) and associated sensitivity parameters values (phantom.coef), analytic model (old.model), its results (old.model.par) and fit measures (old.model.fit), sensitivity analysis model (sens.model), its fit measures (sens.fit), outcome of the objective function (outcome), sensitivity parameters across all converged evaluations (sens.pars), sensitivity analysis model results (model.results), analytic model results (old.out), and the first converged sensitivity analysis model results (sens.out).

References


We thank Dr. Krzysztof Socha for providing us the ACO code for continuous domains (http://iridia.ulb.ac.be/supp/IridiaSupp2008-001/) that the current function is based on.

Examples

library(lavaan)
# Generate data, this is optional as lavaan also takes variance covariance matrix
sim.model <- 'x =~ x1 + 0.8*x2 + 1.2*x3
  y =~ y1 + 0.5*y2 + 1.5*y3
  m ~ 0.5*x
  y ~ 0.5*x + 0.8*m'
set.seed(10)
data <- simulateData(sim.model, sample.nobs = 1000L)
# standardize dataset
data = data.frame(apply(data,2,scale))

# Step 1: Set up the analytic model of interest
model <- 'x  =  x1  +  x2  +  x3
       y  =  y1  +  y2  +  y3
       m  =  x
       y  =  x  +  m'

# Step 2: Set up the sensitivity analysis model.
# The sensitivity parameters are phantom1, phantom2, and phantom3 in this example.
sens.model = 'x  =  x1  +  x2  +  x3
              y  =  y1  +  y2  +  y3
              m  =  x
              y  =  x  +  m
              x  ~  phantom1*phantom
              m  ~  phantom2*phantom
              y  ~  phantom3*phantom
              phantom  =  0  # added for mean of zero
              phantom  ~  1+phantom'  # added for unit variance

# Step 3: Set up the paths of interest to be evaluated in sensitivity analysis.
# Suppose we are interested in all direct and indirect paths.
paths <- 'm  ~  x
          y  ~  x  +  m'

# Step 4: Perform sensitivity analysis
my.sa <- sa.aco(data, model = model, sens.model = sens.model,
                 opt.fun = 3, k = 5, #p-value
                 paths = paths,
                 max.iter = 30)
#Note, please specify larger numbers for k (e.g., 100) and max.iter (e.g., 1000)

# Step 5: Summarize sensitivity analysis results.
# See sens.tables function for explanation of results.
tables <- sens.tables(my.sa)

---

sens.tables

Summary of sensitivity analysis results

Description

This function can summarize the sensitivity analysis results from sa.aco function.

Usage

sens.tables(expr = NULL, sig.level = 0.05, path = TRUE, sort = TRUE)
Arguments

- **expr**
  Returned object of `sa.aco` function.

- **sig.level**
  Significance level, default value is 0.05.

- **path**
  Logical, if TRUE, the function only present results for structural paths. If FALSE, the function will present results for all paths (including structural paths and measurement paths). Default value is TRUE.

- **sort**
  Logical, if TRUE, the function will present sorted results. If FALSE, the function will present unsorted results. Default value is TRUE.

Value

Lists of 5 summary tables. The first table (sens.summary) provides analytic model results (model path coefficient/model.est, p value/pvalue), mean, minimum, and maximum values of estimated path coefficients across all sensitivity analysis models (mean.est.sens, min.est.sens, and max.est.sens). The second table (phan.paths) provides the summary of sensitivity parameters, including the mean, minimum, and maximum values of each sensitivity parameters (mean.phan, min.phan, max.phan). The third table (phan.min) provides the sensitivity parameters that lead to the minimum path coefficient estimation in a sensitivity analysis model. The fourth table (phan.max) provides the sensitivity parameters that lead to the maximum path coefficient estimation in a sensitivity analysis model. The fifth table (p.paths) provides the sensitivity parameters, if any, that lead to the change of p value across the significance level.

References


Examples

```r
# see examples in the \code{\link{sa.aco}} function
```

Smith19.use data.

Description

Usage

smith19.use

Format

A data frame with 3444 observations and 39 variables

Details

The dataset was taken from the public-use data of the Early Childhood Longitudinal Study – Kindergarten Class of 1998-99 of the National Center for Educational Statistics (https://nces.ed.gov/ecls/kindergarten.asp/). This dataset should not be combined with other data for the purpose of identifying participants.
Index

* datasets
  smith19.use, 8

gen.sens.pars, 2, 3

sa.aco, 2, 4, 7, 8
SEMsens (SEMsens-package), 2
SEMsens-package, 2
sens.tables, 2, 7
smith19.use, 8