Package 'LAWBL'

July 23, 2020

Type Package

Title Latent (Variable) Analysis with Bayesian Learning

Version 1.1.0

Author Jinsong Chen [aut, cre, cph]

Maintainer Jinsong Chen <jinsong.chen@live.com>

Description An analytical framework for latent variables with different Bayesian learning methods, currently based on the partially confirmatory factor analysis (PCFA) model by Chen, Guo, Zhang, & Pan (2020) <DOI:10.1037/met0000293>.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.6.0)

Imports stats, MASS, coda

RoxygenNote 7.1.1

URL https://github.com/Jinsong-Chen/LAWBL,
https://jinsong-chen.github.io/LAWBL

BugReports https://github.com/Jinsong-Chen/LAWBL/issues

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2020-07-23 16:22:07 UTC

R topics documented:

`LAWBL-package` *LAWBL: Latent (Variable) Analysis with Bayesian Learning*

Description

The long-term goal of this package is to provide a analytical framework for modeling latent variables with different Bayesian learning methods.

Details

Currently, this package includes the Partially Confirmatory Factor Analysis (PCFA), a partially confirmatory approach covering a wide range of the exploratory-confirmatory continuum in factor analytic models (Chen, Guo, Zhang, & Pan, 2020). There are two major model variants with different constraints for identification. One assumes local independence (LI) with a more exploratory tendency, which can be also called the E-step. The other allows local dependence (LD) with a more confirmatory tendency, which can be also called the C-step. Parameters are obtained by sampling from the posterior distributions with the Markov chain Monte Carlo (MCMC) techniques. Different Bayesian Lasso methods are used to regularize the loading pattern and local dependence.

Although only continuous data are supported currently, inclusion of mixed-type data is on schedule. More Bayesian learning approaches will be also included in future releases of this package.

Note

This package is under development. You are very welcome to send me any comments or suggestions for improvements, and to share with me any problems you may encounter with the use of this package.

Author(s)

Jinsong Chen, `<jinsong.chen@live.com>`

References

Chen, J., Guo, Z., Zhang, L., & Pan, J. (2020). A partially confirmatory approach to scale development with the Bayesian Lasso. *Psychological Methods*. Advance online publication. DOI:10.1037/met0000293

pcfa *Partially Confirmatory Factor Analysis*

Description

PCFA is a partially confirmatory approach covering a wide range of the exploratory-confirmatory continuum in factor analytic models (Chen, Guo, Zhang, & Pan, 2020). There are two major model variants with different constraints for identification. One assumes local independence (LI) with a more exploratory tendency, which can be also called the E-step. The other allows local dependence (LD) with a more confirmatory tendency, which can be also called the C-step. Parameters are obtained by sampling from the posterior distributions with the Markov chain Monte Carlo (MCMC) techniques. Different Bayesian Lasso methods are used to regularize the loading pattern and LD. The estimation results can be summarized with `summary.lawbl` and the factorial eigenvalue can be plotted with `plot_eigen`.

Usage

```
pcfa(
  dat,
  Q,
  LD = TRUE,
  PPMC = FALSE,
  burn = 5000,
  iter = 5000,
  update = 1000,
  missing = NA,
  rseed = 12345,
  digits = 4,
  alas = FALSE
)
```

Arguments

`dat`	A $N \times J$ `matrix` or `data.frame` consisting of the responses of N individuals to J items. Only continuous data are supported currently.
`Q`	A $J \times K$ design matrix for the loading pattern with K factors and J items. Elements are 1, -1, and 0 for specified, unspecified, and zero-fixed loadings, respectively. For models with LI or the E-step, one can specify a few (e.g., 2) loadings per factor. For models with LD or the C-step, the sufficient condition of one specified loading per item is suggested, although there can be a few items without any specified loading (Chen, Guo, Zhang, & Pan, 2020). See `Examples`.
`LD`	logical; `TRUE` for allowing LD (model with LD or C-step).
`PPMC`	logical; `TRUE` for conducting posterior predictive model checking.
`burn`	Number of burn-in iterations before posterior sampling.
`iter`	Number of formal iterations for posterior sampling.

`update` Number of iterations to update the sampling information.

`missing` Value for missing data (default is `NA`) (under development).

`rseed` An integer for the random seed.

`digits` Number of significant digits to print when printing numeric values.

`alas` logical; for adaptive Lasso or not. The default is `FALSE`, which seems slightly stabler.

Value

`pcfa` returns an object of class `pcfa`. It contains a lot of information about the posteriors that can be summarized using `summary.lawbl`. The factorial eigenvalue can be plotted with `plot_eigen`.

References

Chen, J., Guo, Z., Zhang, L., & Pan, J. (2020). A partially confirmatory approach to scale development with the Bayesian Lasso. *Psychological Methods*. Advance online publication. DOI:10.1037/met0000293

Examples

```
####################################
#  Example 1: Estimation with LI   #
####################################

dat <- sim18cfa1$dat
J <- ncol(dat)
K <- 3
Q<-matrix(-1,J,K);
Q[1:2,1]<-Q[9:10,2]<-Q[13:14,3]<-1
Q
m0 <- pcfa(dat = dat, Q = Q, LD = FALSE,burn = 1000, iter = 1000)
summary(m0) # summarize basic information
summary(m0, what = 'lambda') #summarize significant loadings
summary(m0, what = 'qlambda') #summarize significant loadings in pattern/Q-matrix format
summary(m0, what = 'offpsx') #summarize significant LD terms

####################################
#  Example 2: Estimation with LD   #
####################################

Q<-matrix(-1,J,K);
Q[1:6,1]<-Q[7:12,2]<-Q[13:18,3]<-1
Q
m1 <- pcfa(dat = dat, Q = Q,burn = 1000, iter = 1000)
summary(m1) # summarize basic information
summary(m1, what = 'qlambda') #summarize significant loadings in pattern/Q-matrix format
summary(m1, what = 'offpsx') #summarize significant LD terms
```

plot_eigen *Factorial eigenvalue plot*

Description

Provide plots based on the factorial eigenvalues of a `pcfa` objects.

Usage

```
plot_eigen(obj, what = "trace")
```

Arguments

`obj`	A `lawbl` object
`what`	A list of options for what to plot. • `trace`: The trace of each factor's eigenvalue. • `density`: The trace of each factor's eigenvalue. • `APSR`: The pseudo Gelman-Rubin diagnostics of each factor's eigenvalue.

Examples

```

dat <- sim18cfa0$dat
J <- ncol(dat)
K <- 3
Q<-matrix(-1,J,K);
Q[1:2,1]<-Q[9:10,2]<-Q[13:14,3]<-1


m0 <- pcfa(dat = dat, Q = Q, LD = FALSE,burn = 1000, iter = 1000)
plot_eigen(m0) # trace
plot_eigen(m0, what='density')
plot_eigen(m0, what='APSR')
```

sim18cfa0 *Simulated CFA data with LI*

Description

CFA data simulated based on 18 items, 3 factors and local independence; factorial correlation $\Phi = .3$.

Usage

```
sim18cfa0
```

Format

A list with components:

`dat` A dataset with simulated responses of 1000 individuals to 18 items

`qlam` Loading pattern and values used to simulated the data

`sim18cfa1` *Simulated CFA data with LD*

Description

CFA data simulated based on 18 items, 3 factors and local dependence; factorial correlation $\Phi = .3$.

Usage

```
sim18cfa1
```

Format

A list with components:

`dat` A dataset with simulated responses of 1000 individuals to 18 items

`qlam` Loading pattern and values used to simulated the data

`LD` Local dependence between items (LD effect = .3)

`summary.lawbl` *Summary method for* `pcfa` *objectects*

Description

Provide basic information for an `PCFA` object, and summarize various posteriors.

Usage

```
## S3 method for class 'lawbl'
summary(
  object,
  what = "basic",
  med = FALSE,
  SL = 0.05,
  detail = FALSE,
  digits = 4,
  ...
)
```

Arguments

`object` A `lawbl` object

`what` A list of options for what to summarize.

- `basic`: Basic information about the model and posteriors.
- `lambda`: Loading estimates.
- `qlambda`: Loading estimates in pattern/Q-matrix format.
- `eigen`: Factorial eigen value.
- `dpsx`: Diagonal elements in the residual covariance matrix `PSX`.
- `offpsx`: Off-diagonal elements in `PSX`; local dependence terms.
- `phi`: Factorial correlations.
- `shrink`: (Ave) shrinkage for each facor's loadings and LD (if `LD` in `pcfa` = T).
- `all`: All above information.

`med` logical; if the posterior median (`TRUE`) or mean (`FALSE`) is used as the estimate.

`SL` Significance level for interval estimate. The default is .05.

`detail` logical; if only significant (`FALSE`) or all (`TRUE`) estimates are presented.

`digits` Number of significant digits to print when printing numeric values.

`...` additional arguments

Value

A list or matrix containing the summarized information based on the option `what`.

Examples

```
dat <- sim18cfa0$dat
J <- ncol(dat)
K <- 3
Q<-matrix(-1,J,K);
Q[1:2,1]<-Q[9:10,2]<-Q[13:14,3]<-1

m0 <- pcfa(dat = dat, Q = Q, LD = FALSE,burn = 1000, iter = 1000)
summary(m0) # summarize basic information
summary(m0, what = 'lambda') #summarize significant loadings
summary(m0, what = 'qlambda') #summarize significant loadings in pattern/Q-matrix format
summary(m0, what = 'offpsx') #summarize significant LD terms
summary(m0, what = 'all') #summarize all information
```

Index