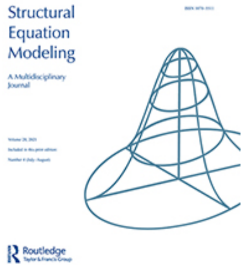

Routledge
Taylor & Francis Group

Structural Equation Modeling: A Multidisciplinary Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/hsem20

blcfa: An R Package for Bayesian Model Modification in Confirmatory Factor Analysis

Lijin Zhang, Junhao Pan, Laurette Dubé & Edward Haksing Ip

To cite this article: Lijin Zhang, Junhao Pan, Laurette Dubé & Edward Haksing Ip (2021) blcfa: An R Package for Bayesian Model Modification in Confirmatory Factor Analysis, Structural Equation Modeling: A Multidisciplinary Journal, 28:4, 649-658, DOI: 10.1080/10705511.2020.1867862

To link to this article: https://doi.org/10.1080/10705511.2020.1867862

Published online: 23 Mar 2021.

STRUCTURAL EQUATION MODELING: A MULTIDISCIPLINARY JOURNAL
2021, VOL. 28, NO. 4, 649–658
https://doi.org/10.1080/10705511.2020.1867862

blcfa: An R Package for Bayesian Model Modification in Confirmatory Factor Analysis

Lijin Zhang[a], Junhao Pan[a], Laurette Dubé[b], and Edward Haksing Ip[c]

[a]Sun Yat-sen University; [b]McGill University; [c]Wake Forest School of Medicine

ABSTRACT

In confirmatory factor analysis (CFA), post hoc model modification (PMM) indexes are often used to adjust for possible residual correlations between items. Although the approach is useful for improving model goodness-of-fit, it requires an iterative, one-item-pair-at-a-time procedure that can be tedious and prone to error. This paper provides a didactic discussion in the form of a tutorial of a more efficient and practical alternative and its implementation using an R-based package. The tutorial contains (1) the Bayesian covariance Lasso (least absolute shrinkage and selection operator) approach as an alternative to the PMM method, and (2) the R package **blcfa**, which implements the Bayesian covariance lasso and directly interfaces with M*plus*. It adopts a two-step approach by first estimating the entire residual covariance matrix, and then identifying the nonzero entries and seamlessly feeding them into M*plus*. Two examples were used to illustrate package implementation.

KEYWORDS
Bayesian method; blcfa package; confirmatory factor analysis; covariance Lasso

Confirmatory factor analysis (CFA) is one of the most widely used tools in psychology and social sciences for investigating theory-driven relationships between observed indicators and latent factors. Unlike an exploratory factor analysis (EFA), the number of latent factors and factor structure are assumed to be specified a priori in a CFA model. Specifically, in a traditional CFA, the cross-loadings and residual covariances between observed indicators (items) are fixed exactly at zero to reflect the parsimoniously non-overlapping factor structure, and conditional independence between item pairs is assumed given factor; see Figure 1(a) for an example of a two-factor model. The latter assumption implies that the covariance matrix is diagonal. However, these settings are often too strict to be satisfied in real applications and could thus lead to overly rejecting a correctly specified CFA model. In practice, the post hoc model modification (PMM) approach has been used to relax the strict assumption. An important goal of PMM is to free up some residual covariance parameters (e.g., Figure 1(b)) to achieve better goodness of fit. Operationally, the PMM first uses modification indexes (MIs; Sörbom, 1989) to indicate possible non-zero elements in off-diagonal entries in a residual covariance matrix. For example, in M*plus* (Muthén & Muthén, 1998–2017), a list of candidate item pairs and their corresponding MIs would be reported for modification consideration. Based on the diagnostic report, a user identifies an item pair to include in the model to free the corresponding (off-diagonal) residual covariance parameter. The user can then rerun and examine the updated model to determine if another item pair needs to be included, and if so, which pair should be included. The one-item-pair-at-a-time procedure is necessary because by modifying one residual covariance, the resulting model would generate a different set of MIs. For example, residual covariances for item pairs (1,6) and (2,3) are both significant, but after including (1,6) into the model (2,3) becomes not significant and (2,7) becomes significant. In other words, the PMM requires a user to sequentially tweak the residual covariance matrix one entry at a time in order to determine the next specific residual covariance parameters for inclusion in the modified model, refit the model, use the new set of MIs to determine the next residual covariance entry for inclusion, and so on. Such a sequential procedure, which has to be manually conducted, becomes tedious when the number of significant residual covariances is more than a few. The process is also subjected to the user's own decision of when to stop. Additionally, no guarantee exists that the modified covariance matrix is always positive definite and therefore a convergence problem can arise.

As an alternative to post hoc model modification (PMM), Pan et al. (2017) proposed the Bayesian covariance Lasso (least absolute shrinkage and selection operator) approach to the PMM. The Lasso-based approach avoids the ad-hoc sequential modification based on MIs and estimates a positive definite and sparse residual covariance structure of the observed indicators. That is, under the Bayesian covariance Lasso CFA (BLCFA) framework, the residual covariances that bounded away from zero will be *simultaneously* detected. The result can then be used in at least the following two ways. First, by efficiently identifying the collection of residual covariance parameters, a one-step modification is possible for improving the overall goodness of fit. Second, by using the simultaneous examination of residual covariances in a joint CFA model as a diagnostic tool, a user can determine if other forms of modification are necessary (e.g., deletion of certain domain-specific items based on cross-factor residual covariance or a modification of factor structure).

CONTACT Junhao Pan panjunh@mail.sysu.edu.cn Department of Psychology, Sun Yat-sen University, 132 Waihuan Dong Road, Higher Education Mega Center, Guangzhou 510006, China.

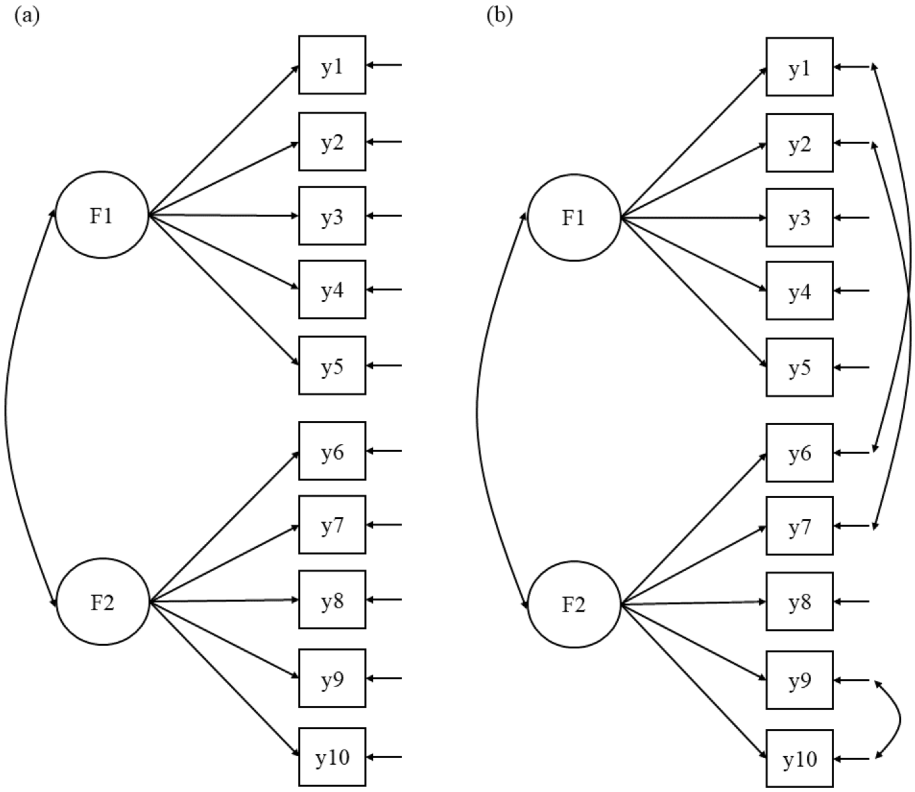

Figure 1. Model structure in software implementation example.

Although the R codes for implementing a BLCFA were published in Pan et al. (2017), applied researchers must still take multiple steps in obtaining the necessary modified CFA. For example, the user has to examine the covariance matrix, identify the nonzero entries to retain, modify the matrix, and apply existing software to obtain the fitted model and related goodness-of-fit indexes. The problem becomes even more complicated in the case of a CFA model that needs further changes when diagnostic information is available (e.g., item deletion, factorial structure change, and so forth).

In this paper we describe the R package **blcfa**, which implements a two-step method by (1) applying the BLCFA to a data set in standard data format (e.g., as required by M*plus*) and identifies nonzero (off-diagonal) entries in the residual covariance matrix, and by (2) seamlessly integrating the BLCFA result with M*plus* software and deriving the necessary modified CFA result for the end user. In this two-step method, the user is also provided the option of directly examining the significant residual covariances detected after step (1) and makes changes to the generated M*plus* codes to select the desired covariance to modify. The package output includes model fit indexes and parameter estimates for the CFA with updated residual covariance parameters as typically reported in an M*plus* output file. Accordingly, the program will greatly improve CFA analysis efficiency for applied researchers with a working knowledge of M*plus*. For example, an end user can run **blcfa** and accept the result from the program. Alternatively, the end user can examine the required modified entries of the covariance matrix as diagnostic information and determine if item deletion is necessary. If revision is needed, the revised specification can also be implemented using **blcfa** and without switching between programs.

In this paper, we explain the basic principles of BLCFA and provide a step-by-step tutorial of how **blcfa** can be used to assess residual covariances and integrate nonzero entries with M*plus* to generate output. Both simulated data and real data are used for the illustration.

The Bayesian covariance Lasso CFA model

To explain BLCFA, we start with the standard CFA. Consider a collection of p-variate random response $\mathbf{y}_i = (y_{i1}, y_{i2}, \cdots, y_{ip})^T (i = 1, 2, \cdots, n)$:

$$\mathbf{y}_i = \boldsymbol{\mu} + \boldsymbol{\Lambda}\boldsymbol{\omega}_i + \boldsymbol{\varepsilon}_i, i = 1, 2, \ldots, n, \tag{1}$$

where $\boldsymbol{\mu}$ $(p \times 1)$ is vector of intercepts, and $\boldsymbol{\Lambda}$ $(p \times q)$ is a factor-loading matrix that reflects the relation of observed indicators in $\mathbf{y}_i$ with the latent factors in $\boldsymbol{\omega}_i$ $(q \times 1)$. Moreover, it is assumed that $\boldsymbol{\omega}_i$ follows $N[\mathbf{0}, \boldsymbol{\Phi}]$, and $\boldsymbol{\varepsilon}_{\mathrm{i}}$ $(p \times 1)$ is a random vector of measurement errors that follows $N[\mathbf{0}, \boldsymbol{\Psi}]$. Following the Bayesian modeling approach of Pan et al. (2017), the number of latent factors and the structure of $\boldsymbol{\Lambda}$ are specified a priori, the variance-covariance matrix $\boldsymbol{\Psi}$ is not necessarily diagonal, and the unknown parameters in the model follow specific prior distributions. The last two points are where the BLCFA differs from traditional CFA.

Relaxing the diagonality assumption in the covariance matrix can lead to a substantial increase in the number of model parameters. For example, in a 30-item data set, there are 435 unique off-diagonal parameters, which does not expect to be all non-zero. A commonly used approach to maintain model parsimoniousness is to use regularization. Within the frequentist framework, a commonly used regularization method is the Lasso, which adds an L_1-norm penalty to the likelihood in penalizing the total number of added parameters (Tibshirani, 1996). The Bayesian Lasso estimates accomplish the same goals through choosing appropriate forms of prior distributions. Specifically, the L_1-norm penalty is applied to the inverse of the covariance matrix with the graphical Lasso priors where independent exponential priors and the double exponential priors are assigned respectively for the diagonal and the off-diagonal elements of $\boldsymbol{\Psi}^{-1}$ (Khondker et al., 2013; Wang, 2012). Let $\Sigma = \boldsymbol{\Psi}^{-1} = (\sigma_{ij})_{p\times p}$ and λ be the shrinkage parameter, the double exponential density to have the form $\frac{\lambda}{2}\exp(-\lambda|\sigma_{ij}|)$, $i<j$, whereas the exponential density function has the form $\frac{\lambda}{2}\exp(-\frac{\lambda}{2}\sigma_{ii})$.

In frequentist Lasso estimates, the shrinkage parameter λ is often chosen by procedures such as cross-validation (Tibshirani, 1996). Under the Bayesian framework, the shrinkage parameter λ is empirically determined along with other parameters simultaneously (Park & Casella, 2008). An appropriate prior is assigned to λ; that is, $\lambda \sim Gamma(\alpha_{\lambda 0}, \beta_{\lambda 0})$ where the common choices for $\alpha_{\lambda 0}$ and $\beta_{\lambda 0}$ are $\alpha_{\lambda 0} = 1$ and $\beta_{\lambda 0}$ (e.g., value = 0.01) is small.

For the unknown parameters involved in $\boldsymbol{\mu}$, $\boldsymbol{\Lambda}$, $\boldsymbol{\Phi}$, the following conjugate prior distributions are considered: For $k = 1, 2, \cdots, p$,

$$\boldsymbol{\mu} \sim N(\boldsymbol{\mu}_0, \mathbf{H}_{\mu 0}), \ \boldsymbol{\Lambda}_k \sim N(\boldsymbol{\Lambda}_{0k}, \mathbf{H}_{0k}),$$

$$\boldsymbol{\Phi}^{-1} \sim \text{Wishart}(\mathbf{R}_0, \rho_0), \tag{2}$$

where $\boldsymbol{\Lambda}_k^T$ is the kth row of $\boldsymbol{\Lambda}$. $\boldsymbol{\mu}_0$, $\boldsymbol{\Lambda}_{0k}$, ρ_0, and positive definite matrices $\mathbf{H}_{\mu 0}$, $\mathbf{H}_{0k}$, and $\mathbf{R}_0$ are hyperparameters whose values are assumed to be given from prior information of previous studies or other sources.

The two-step method for model estimation

The two-step approach in the **blcfa** consists of the following procedure:

(1) detect the significant residual covariances that are different from zero by the Bayesian Covariance Lasso method;
(2) free the identified residual covariance parameters and seamlessly feed the output from (1) into M*plus* to obtain an appropriately modified CFA model for estimation.

In step (2), the residual covariances not deemed significantly different from zero are assigned a value of zero for inputting to M*plus*. The default significant criterion in detecting residual covariances is that zero is not included in the 95% highest posterior density (HPD) interval. Other criteria such as p-value (Muthén, 2010) with $\alpha = 0.05$ can also be specified. To illustrate the **blcfa** package usage in detail, we first provide an example using a small simulated data set so readers can follow and replicate. We then apply **blcfa** to a real data set to illustrate how the program functions in practice.

Software implementation example

Before using the **blcfa** package, the M*plus* software needs to be available. If the user does not have access to M*plus* or the environment variable does not include the path of M*plus*, the user gets the following warning message:

```
1 Error: Failed to run the Mplus software,
check whether the Mplus has been installed or
the environment variable of your computer
includes the path of Mplus.
```

The user still obtains the M*plus* input file (.inp) and detailed results of the Bayesian Lasso CFA. However, the M*plus* output file that contains the second-step analysis results cannot be automatically generated. We recommend the latest version of R software as **blcfa** depends on other packages built with the latest version of R. Researchers that install the package from Github will be able to run the most up-to-date version of the **blcfa** codes:

```
1 install.packages(“devtools”)
2 library(devtools)
3 install_github(“zhanglj37/blcfa”)
```

The line numbers in the left margin of the code are not part of the R code. Specifically, the first line of code prompts the **devtools** package download (Wickham et al., 2019) from CRAN. The command *library(devtools)* loads the **devtools** package. The latest **blcfa** package can be downloaded from Github using the install_github() function from the **devtools** package. The **blcfa** package depends on other packages such as **MCMCpack, sna**, and so forth. These additional packages can be automatically downloaded after running the abovementioned three lines of codes.

The following simple example illustrates the essential **blcfa** functions. The data were generated based on the model described in Figure 1(b): a two-factor CFA model with five items per factor and three non-zero residual covariances ($\psi_{16}, \psi_{27}, \psi_{9,10}$). In this two-factor model, the true value of the first loading per factor was fixed to 1.0 and other loadings were set at 0.8. The factor variance and covariance were specified as 1.0 and 0.3, respectively. Residual correlations were set at 0.7 and residual variances (ψ_{kk}, where $k = 1, \cdots, 10$) were set at 0.36. The data file (simu_data.txt) is embedded in the **blcfa** package:

```
1 library(blcfa)
2 setwd(“C:/Users/Desktop/SimuExample/”)
3 filename = system.file(“extdata”, “simu_
  data.txt”, package = “blcfa”)
4 varnames<-c(paste(“y”, 1:10, sep = “ ”))
5 usevar <- varnames
6 myModel<-"
7 f1 =~ y1 + y2 + y3 + y4 + y5
8 f2 =~ y6 + y7 + y8 + y9 + y10
9 "
10 set.seed(1)
11 results <- blcfa(filename, varnames, usevar,
myModel, estimation = “both”, MCMAX = 5000, N.
burn = 2500, bloutput = TRUE, interval = TRUE)
```

Code for this example is also available through the command ?blcfa(). “In the second line”, the setwd() function is used to specify the working directory at which the user wants the **blcfa** package outputs (e.g., M*plus* files) stored. The embedded data file can be specified using the system.file() function. The *filename* variable records the absolute path of the data file, which is not in the current working directory. The *extdata* and *simu_data.txt* command refer to the location and file name of the raw data in the **blcfa** package, respectively. The names of variables should not be included in the data file. Each column in the data set represents the values of one specific variable and numbers on each line can be separated by one or more spaces or a tab. To facilitate a seamless integration with M*plus*, only ASCII data files (e.g., data file suffixed with *.dat, .txt*) should be used. There is no missing data in the data file *simu_data.txt*. Handling of missing data will be discussed in the "Real-data Illustration" section.

Lines 4 and 5 indicate the names of the variables in the data set and a subset of variables for analysis ($y_1 - y_{10}$). In lines 6–9, the model (Figure 1(a)) is defined by a string: $= \sim$ defines the relationship between latent factors and the corresponding observed indicators. Each variable and symbol should be separated by one or more spaces. The set.seed() function is used to specify the random seed used for random draws in the Bayesian lasso CFA. The set.seed command allows replicability of the result.

Line 11 shows the main **blcfa** function, which is called blcfa(). The parameter values in blcfa() can have different options as follows: *estimation* denotes the estimation method within M*plus*, which takes the location of the significant item pairs identified by the Bayesian Lasso CFA and re-estimates the residual covariance matrix entries for the identified item pairs. The choice of estimation method is specified in blcfa() and passed onto M*plus*. When *estimation* value = *both*, both the ML (maximum likelihood) CFA and the Bayesian CFA will be reported. The default value for *estimation* is *ml*. When non-normality is detected, (|Skewness| > 2 or |Kurtosis| > 7; West et al., 1995), the mean adjusted maximum likelihood (MLM) will be used instead of ML. The parameter value of *estimation* can also be set at *bayes* to conduct the Bayesian CFA alone in the second-step analysis. The Bayesian CFA used in M*plus* adopts a Gibbs sampling with a random-walk algorithm when residual covariance entries are estimated (Asparouhov & Muthén, 2010). Note that the choice of estimator in M*plus* has no influence on the residual covariance identification of item pairs in the **blcfa** first-step. In other words, the modified entry locations in the covariance matrix remain the same regardless of the choice of estimation method.

The variable *MCMAX* defines the total number of Markov Chain Monte Carlo (MCMC) samples for inference (the default value is 10,000). *N.burn* denotes the number of discarded MCMC samples (the default value is 5,000). By setting *bloutput* as TRUE (the default value is FALSE), users can obtain detailed results of the Bayesian covariance lasso CFA (e.g., posterior predictive *p*-value (PP*p*), estimates, and 95% HPD intervals of unknown model parameters). The parameter *interval* is used to specify the method for detecting significant residual covariances by a 95% HPD interval (the default is *interval* = TRUE) or *p*-value with $\alpha = 0.05$ (*interval* = FALSE).

Within the MCMC algorithm, two separate MCMC chains are generated and run simultaneously (see parallel computation in section Other Features). In this analysis, it took approximately 3 minutes to complete the Gibbs sampling within the following computing environment: PC with Intel Core i5-9400 F @2.90 GHz CPU and 16Gb RAM. The iteration progress of the MCMC chains is recorded in the *log.txt* file:

```
The program is running. See "log.txt" for
details.
```

Researchers can open the *log.txt* to check the process of the program:

```
Num of Iterations: 100
Num of Iterations: 200
Num of Iterations: 300
Num of Iterations: 400
. . .…
```

The metrics for assessing convergence, measured by the estimated potential scale reduction (EPSR) values (Gelman, 1996), are calculated by the **sna** package. The convergence criterion used in the package is EPSR value < 1.2 for each parameter. If the model fails to converge after the burn-in period, no result would be generated:

```
Error: Failed to satisfy the convergence
criterion. Check the epsr graph and increase
the values of N.burn and MCMAX.
```

The EPSR graph can be automatically generated by the package if the model fails to converge or the *bloutput* is set as TRUE. Users can inspect the EPSR graph and consider to increase the *N.burn* and *MCMAX* values. This run took approximately 1 minute to obtain the EPSR values. After achieving convergence (Figure 2), the package reports statistics from the posterior distribution. The MCMC sample means are used as point estimates. Significant residual covariances were detected by 95% HPD intervals. Three residual covariances were identified in the first-step analysis, and the data were deemed normally distributed (Skewness: −0.134~0.237, Kurtosis: −0.562~0.700).

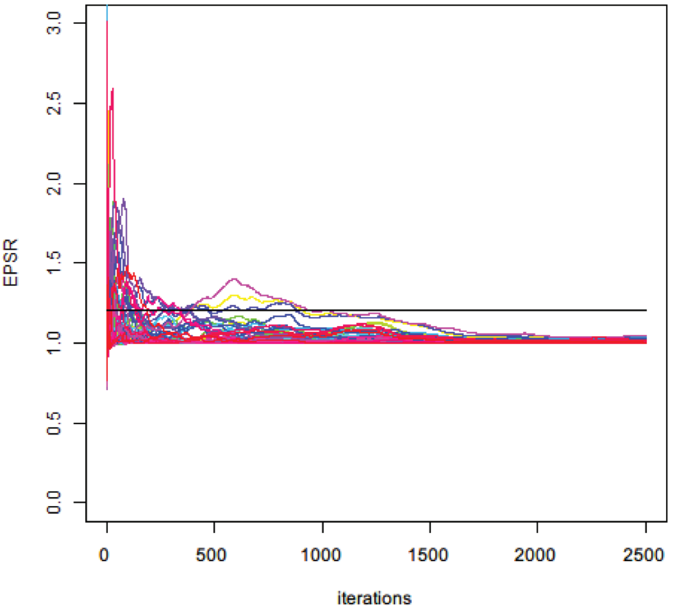

Figure 2. Estimated Potential Scale Reduction (EPSR) values in the software implementation example. The black horizontal line indicates the threshold of EPSR value 1.2).

The **MplusAutomation** package (Hallquist & Wiley, 2018) provides the runModels() function that can be used for running the M*plus* input file. As a result, **blcfa** generated the following M*plus* codes for ML estimation:

```
TITLE: Bayesian Lasso CFA
 DATA: FILE = D:/Software/R/R-4.0.1/
 library/blcfa/extdata/simu_data.txt;
VARIABLE:
 NAMES = y1 y2 y3 y4 y5 y6 y7 y8 y9 y10;
 USEV = y1 y2 y3 y4 y5 y6 y7 y8 y9 y10;
 ANALYSIS:
 ESTIMATOR = ML;
 MODEL:
 f1 by y1 y2 y3 y4 y5;
  f2 by y6 y7 y8 y9 y10;
  y6 with y1;
  y7 with y2;
  y10 with y9;

 OUTPUT: TECH1 STDYX;
```

In this example, M*plus* input and output files were generated in the working directory – "C:/Users/Desktop/SimuExample/". The *FILE* in the second line is the same as the *filename* variable as previously obtained through the system.file() function and records the absolute path of the data file, not the working directory. With exceptions for *ANALYSIS, OUTPUT*, and *PLOT*, the code for the Bayesian CFA model is similar to that for the ML estimator:

```
TITLE: Bayesian Lasso CFA
DATA: FILE = D:/Software/R/R-4.0.1/library/
blcfa/extdata/simu_data.txt;
VARIABLE:
 NAMES = y1 y2 y3 y4 y5 y6 y7 y8 y9 y10;
 USEV = y1 y2 y3 y4 y5 y6 y7 y8 y9 y10;
 ANALYSIS:
 ESTIMATOR = BAYES;
 ALGORITHM = GIBBS(RW);
 PROC = 2;
  BITERATIONS = (10000);
  MODEL:
  f1 by y1 y2 y3 y4 y5;
  f2 by y6 y7 y8 y9 y10;
  y6 with y1;
  y7 with y2;
  y10 with y9;

  OUTPUT: TECH1 TECH8 STDYX;
  PLOT: TYPE = PLOT2;
```

In the *ANALYSIS* part, ALGORITHM = GIBBS(RW) refers to the Gibbs sampling with the random-walk algorithm (Asparouhov & Muthén, 2010). PROC is specified at 2 to speed up computations as dual processors are available. BITERATIONS specifies the minimum number of iterations, which is set at 10,000. The maximum number of iterations is set at 50,000 by default and it can also be specified manually before the brackets. Implementation of the M*plus* code is automatic. If a user wants to inspect the results of the first-step analysis and the second-step analysis, the following codes can be applied within R:

```
## results of the first-step analysis
> results
$blcfa_est
$blcfa_est $ppp
[1] 0.5156

$blcfa_est $ly
est se p-value HPD_lower HPD_upper
f1 by y2 0.936 0.043 0 0.850 1.013
...

## results of the second-step analysis
> sum_second(results)
Reading model: blcfa_bayes.out
Reading model: blcfa_ml.out
$bayes_fit
...
Parameters CFI TLI BIC DIC pD RMSEA_Estimate
RMSEA_90CI_LB
1 34 1 0.999 10991.69 10846.96 33.226 0.001 0
         RMSEA_90CI_UB          RMSEA_pLT05
ObsRepChiSqDiff_95CI_LB
ObsRepChiSqDiff_95CI_UB
1 0.032 1 -28.816 29.6
PostPred_PValue Filename
1 0.465 blcfa_bayes.out

$bayes_par_est
 paramHeader param est posterior_sd pval
lower_2.5ci upper_2.5ci sig
1 F1.BY Y1 1.000 0.000 0 1.000 1.000 FALSE
2 F1.BY Y2 0.777 0.026 0 0.728 0.831 TRUE
...

$bayes_par_est_std
 paramHeader param est posterior_sd pval
lower_2.5ci upper_2.5ci sig
1 F1.BY Y1 0.863 0.013 0 0.835 0.888 TRUE
2 F1.BY Y2 0.811 0.017 0 0.776 0.842 TRUE
...

$ml_fit
...
ChiSqBaseline_PValue LL UnrestrictedLL CFI
TLI AIC BIC
1 0 -5389.93 -5373.401 0.999 0.999 10847.86
10991.16
...

$ml_par_est
paramHeader param est se est_se pval 1 F1.BY
Y1
1.000 0.000 999.000 999
...

$ml_par_est_std
paramHeader param est se est_se pval
1 F1.BY Y1 0.864 0.013 64.475 0
...
```

To save space, we use the " ... " sign to indicate the output not presented. The command *results* shows the detailed results of the first-step analysis, where PP*p* value (= 0.5156). The estimated loading (est) of y_2 on f_1 is 0.936, the standard error (se) estimate is 0.043, the p-value < 0.01, and the credible interval is $(0.850, 1.013)$.

The command *sum_second()* function summarizes the results of the second-step analysis, including the model fitting, and unstandardized and standardized estimates of the Bayesian CFA and the ML CFA. Model-fitting indices in the Bayesian CFA include the Bayesian adaptive comparative fit index (CFI = 1.000), the Tucker–Lewis index (TLI = 0.999), the Bayesian information criterion (BIC = 10991.69), the deviance information criterion (DIC = 10846.96), the root-mean-square error of approximation (RMSEA = 0.001), a 90% confidence interval of RMSEA [0, 0.032], p-value of RMSEA (1.000), a 95% confidence interval for the difference between the observed and the replicated Chi-Square Values [−28.816, 29.6], and the PP*p* value (PostPred_PValue = 0.465). Model-fitting indices in the ML CFA include the p-value of chi-square test of model fit for the baseline model (ChiSqBaseline_PValue = 0), the loglikelihood of model specified in the MODEL command (LL = −5389.93), loglikelihood of the model with unrestricted means, variances, and covariances of the observed variables (UnrestrictedLL = −5373.401), the CFI (0.999), the TLI (0.999), the Akaike information criterion (AIC = 10847.86), the BIC (10991.16), and so on. When *bloutput* was set as TRUE, detailed results of the first-step analysis were saved in the *results* folder rooted at the current working directory:

```
EPSR.png # Figure 2
ppp.csv # posterior predictive p-value
ly.csv # loadings
mu.csv # intercepts
phi.csv # the variance-covariance matrix of
factors
phi_cormatrix.csv # the correlation matrix
of factors
psx.csv # the variance-covariance matrix of
items
psx_sig.csv # significant residual covariances
```

Taking the *psx.csv* as example, this file records the estimates, standard errors, standardized estimates, p-values, and HPD intervals of residual variances and covariances (Table 1). The table also correctly identifies the three non-zero residual covariances in the first-step analysis. The ML CFA and Bayesian CFA estimates in the second-step analysis are similar (Table 2).

Table 1. Partial results of the residual variances and covariances in the first-step analysis.

	est	se	cor	*p*	HPD_lower	HPD_upper
y1 with y1	0.298	0.060	1.000	0.000	0.179	0.414
y2 with y1	0.035	0.050	0.105	0.277	−0.047	0.131
y2 with y2	0.377	0.062	1.000	0.000	0.268	0.491
y3 with y1	0.010	0.047	0.028	0.465	−0.065	0.104
y3 with y2	0.037	0.050	0.095	0.218	−0.046	0.137
y3 with y3	0.399	0.081	1.000	0.000	0.263	0.552
y4 with y1	0.001	0.049	0.002	0.593	−0.092	0.087
y4 with y2	0.040	0.049	0.106	0.189	−0.050	0.129
y4 with y3	−0.025	0.042	−0.063	0.269	−0.103	0.055
y4 with y4	0.380	0.079	1.000	0.000	0.194	0.510
y5 with y1	0.043	0.051	0.133	0.191	−0.045	0.146
y5 with y2	0.018	0.046	0.050	0.376	−0.071	0.097
y5 with y3	−0.002	0.047	−0.005	0.468	−0.089	0.091
y5 with y4	0.006	0.055	0.016	0.517	−0.104	0.100
y5 with y5	0.354	0.068	1.000	0.000	0.211	0.485
y6 with y1	0.184	0.037	0.607	0.000	0.112	0.250
y6 with y2	0.001	0.036	0.004	0.494	−0.063	0.072
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
y7 with y2	0.246	0.042	0.655	0.000	0.174	0.334
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
y10 with y9	0.140	0.076	0.562	0.002	0.004	0.258
y10 with y10	0.244	0.080	1.000	0.000	0.103	0.373

To save space, we selected the 20 lines in *psx.csv*. est = estimate; se = standard error; cor = correlation (standardized estimate); HPD_lower = lower bound of 95% highest posterior density interval; HPD_upper = upper bound of 95% highest posterior density interval.

Table 2. Standardized estimates (Est) in the second-step analysis of the software implementation example.

	ML			Bayes		
Parameter	Est	SE	*p*	Est	SD	95% CI
λ_{11}	0.864	0.013	0.000	0.863	0.013	(0.835,0.888)
λ_{21}	0.810	0.017	0.000	0.811	0.017	(0.776,0.842)
λ_{31}	0.768	0.021	0.000	0.768	0.020	(0.725,0.806)
λ_{41}	0.782	0.020	0.000	0.783	0.020	(0.742,0.818)
λ_{51}	0.817	0.017	0.000	0.817	0.017	(0.780,0.848)
λ_{62}	0.848	0.015	0.000	0.845	0.014	(0.817,0.871)
λ_{72}	0.809	0.017	0.000	0.809	0.017	(0.775,0.841)
λ_{82}	0.808	0.018	0.000	0.808	0.018	(0.769,0.841)
λ_{92}	0.822	0.018	0.000	0.823	0.017	(0.788,0.854)
$\lambda_{10,2}$	0.825	0.017	0.000	0.826	0.016	(0.792,0.856)
ϕ_{12}	0.258	0.046	0.000	0.260	0.045	(0.166,0.342)
ψ_{16}	0.750	0.035	0.000	0.747	0.036	(0.672,0.814)
ψ_{27}	0.679	0.032	0.000	0.679	0.031	(0.615,0.735)
$\psi_{9,10}$	0.675	0.029	0.000	0.674	0.028	(0.614,0.721)

ML = maximum likelihood estimator; Bayes = Bayesian estimation; Est = estimate; SE = standard error; SD = standard deviation of the Markov Chain Monte Carlo samples; 95% CI = 95% confidence interval.

Real-data illustration

We used a previously published structural equation modeling example in Byrne (2012) for our real-data illustration. Byrne and Watkins (2003) analyzed this data set for demonstrating the measurement invariance of a self-concept scale across two groups with different cultural background. Byrne (2012) further used this data to illustrate how to conduct a multigroup CFA model using M*plus*. The data were collected from a survey on four nonacademic self-concept subscales of the Self-Description Questionnaire-I (SDQ-I, Marsh, 1992). There are four latent factors of nonacademic self-concept with eight items per factor (Physical Appearance, PAP, e.g., "I am good looking"; Physical Abilities, PAB, e.g., "I can run fast"; Peer Relations, PER, e.g., "I have lots of friends"; Parent Relations, PAR, e.g., "My parents understand me"). For each item, participants were asked to rate on a 5-point Likert-type scale where 1 = *False*, 2 = *Mostly False*, 3 = *Sometimes False Sometimes True*, 4 = *Mostly True* and 5 = *True*.

The data set was collected from a sample of Nigerian adolescents (N = 463) and included missing scores. Ninety-three participants (20.09%) did not respond to at least one item. The proportion of missing values ranges from 3.13% to 40.63%. Following Byrne and Watkins (2003), cases with more than 8% missing values (N = 30) were deleted. The remaining missing

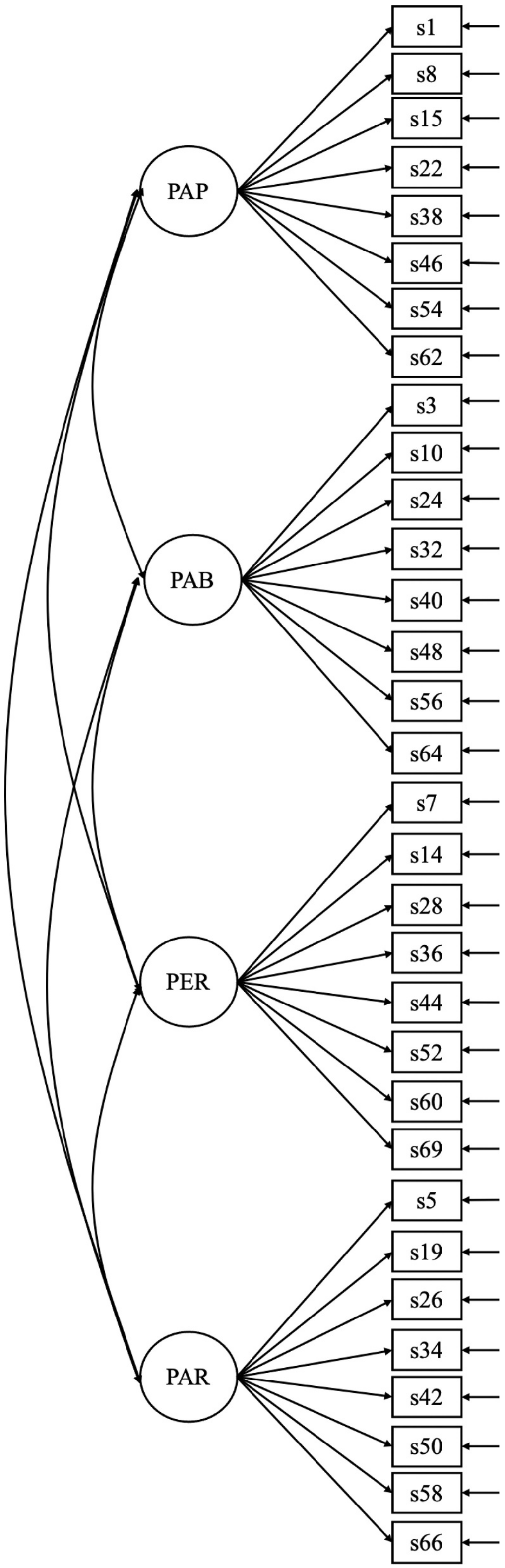

Figure 3. Model structure of the four nonacademic self-concept dimensions of the self-description questionnaire-I (PAP, physical appearance; PAB, physical abilities; PER, peer relations; PAR, parent relations).

scores were assumed as missing-at-random and imputed using a block Gibbs sampler within **blcfa**.

Following Byrne (2012), we treated response data as continuous and used a four-factor CFA model (Figure 3). The Bayesian lasso CFA was applied to the data set with following code:

```
1 library(blcfa)
2 setwd(“C:/Users/Desktop/RealExample/”)
3 filename = “SDQ.dat”
4 varnames <- c(‘s1’, ‘s3’, ‘s5’, ‘s7’, ‘s8’,
‘s10’,
5 ‘s14’, ‘s15’, ‘s19’, ‘s22’, ‘s24’, ‘s26’,
6 ‘s28’, ‘s32’, ‘s34’, ‘s36’, ‘s38’, ‘s40’,
7 ‘s42’, ‘s44’, ‘s46’, ‘s48’, ‘s50’, ‘s52’,
8 ‘s54’, ‘s56’, ‘s58’, ‘s60’, ‘s62’, ‘s64’, ‘s66’,
‘s69’)
9 usevar <- varnames
10 myModel<-”
11 f1 = ~s1 + s8 + s15 + s22 + s38 + s46 + s54 + s62
12 f2 = ~s3 + s10 + s24 + s32 + s40 + s48 + s56 + s64
13 f3 = ~s7 + s14 + s28 + s36 + s44 + s52 + s60 + s69
14 f4 = ~s5 + s19 + s26 + s34 + s42 + s50 + s58 + s66
15 ”
16 set.seed(1234)
17 blcfa(filename, varnames, usevar, myModel,
estimation=“both”, ms = 999, MCMAX = 16000, N.
burn = 8000, bloutput = TRUE)
```

Specifically, the third line of code denotes the name of the data file. The *filename*, which does not contain an absolute path, is relative to the current working directory defined in the second line. In lines 11–14, the four-factor model is defined according to the relationship between latent factors and the corresponding observed indicators in Figure 3. In line 17, the *ms* parameter (here set to the value 999) is used to identify missing or invalid value. When NA is used to represent the missing value, there is no need to define the ms variable. The **blcfa** package imputes missing values using Gibbs sampler and generates a new data set with imputed values for M*plus* analysis.

In this example, we selected *estimation = “both”* so both the ML (frequentist) and Bayesian estimation results were generated. A preliminary analysis showed that the data did not satisfy the multivariate normal distribution assumption. As a result, the algorithm switched from ML over to MLM. The four-factor model with identified residual covariances was eventually re-analyzed using the MLM and the Bayesian estimation. Additionally, the input data set to M*plus* were also updated by standardizing the raw data and imputing missing values (Line 2: data_imputed.txt). The input file of the final model with MLM estimation is as follows.

```
1 TITLE: Bayesian Lasso CFA
2 DATA: FILE = data_imputed.txt;
3 VARIABLE:
4 NAMES = s1 s3 s5 s7 s8 s10 s14 s15 s19 s22
5 s24 s26 s28 s32 s34 s36 s38 s40 s42 s44
6 s46 s48 s50 s52 s54 s56 s58 s60 s62 s64
7 s66 s69;
8 USEV = s1 s3 s5 s7 s8 s10 s14 s15 s19 s22
9 s24 s26 s28 s32 s34 s36 s38 s40 s42 s44
10 s46 s48 s50 s52 s54 s56 s58 s60 s62 s64
11 s66 s69;
12 ANALYSIS:
13 ESTIMATOR = MLM;
14 MODEL:
15 f1 by s1 s8 s15 s22 s38 s46 s54 s62;
16 f2 by s3 s10 s24 s32 s40 s48 s56 s64;
17 f3 by s7 s14 s28 s36 s44 s52 s60 s69;
```

```
f4 by s5 s19 s26 s34 s42 s50 s58 s66;

s8 with s1;
s22 with s15;
s26 with s19;
s28 with s19;
s38 with s3;
s42 with s28;
s46 with s14;
s46 with s26;
s46 with s32;
s48 with s38;
s50 with s38;
s50 with s42;
s50 with s48;
s52 with s10;
s54 with s44;
s56 with s48;
s62 with s34;
s66 with s28;
s66 with s36;
s66 with s42;
s69 with s38;
s69 with s50;

OUTPUT: TECH1 STDYX;
```

The analysis shows that in the residual variance-covariance matrix, 22 or 4.4% significant item-pairs were simultaneously detected out of a total 496 (=C(32,2)) item pairs by the BLCFA method, and all of them were found significant in the second-step analysis using the Bayesian estimation (Table 3). To save space, we highlight one within-factor and one between-factor residual covariances for illustration. The BLCFA method identified significant positive correlation between two items within the PAR factor: the 19th item (*I like my parents*) and the 26th item (*My parents like me*), which seemed to indicate reciprocality in affection between parent and child that leads to positive residual covariance. Result also showed that the 44th item (*Other Kids want me to be their friends*) of PER factor was positively correlated with the 54th item (*I'm better looking than most of my friends*) of the PAP factor. It is possible that the significant residual covariance between the responses to these two items was related to a common driver such as narcissism.

Table 3. The significant residual covariances and their estimates in the second-step analysis.

	MLM			Bayes		
Residual Covariance	Est	SE	*p*	Est	SD	95% CI
s8 with s1	0.133	0.042	0.001	0.139	0.039	(0.066,0.221)
s22 with s15	0.085	0.050	0.090	0.088	0.039	(0.014,0.166)
s26 with s19	0.161	0.045	0.000	0.166	0.034	(0.105,0.235)
s28 with s19	0.064	0.032	0.044	0.065	0.028	(0.012,0.120)
s38 with s3	0.072	0.034	0.037	0.073	0.032	(0.011,0.135)
s42 with s28	0.125	0.046	0.006	0.128	0.037	(0.059,0.201)
s46 with s14	0.152	0.047	0.001	0.155	0.039	(0.078,0.234)
s46 with s26	0.093	0.037	0.011	0.096	0.036	(0.028,0.169)
s46 with s32	0.118	0.036	0.001	0.120	0.038	(0.046,0.196)
s48 with s38	0.106	0.036	0.003	0.109	0.041	(0.030,0.192)
s50 with s38	0.138	0.043	0.001	0.141	0.039	(0.069,0.219)
s50 with s42	0.089	0.035	0.011	0.091	0.035	(0.023,0.163)
s50 with s48	0.108	0.042	0.011	0.110	0.036	(0.040,0.183)
s52 with s10	0.113	0.046	0.014	0.115	0.036	(0.046,0.189)
s54 with s44	0.130	0.034	0.000	0.131	0.035	(0.063,0.203)
s56 with s48	0.070	0.032	0.031	0.071	0.034	(0.005,0.139)
s62 with s34	0.161	0.040	0.000	0.164	0.045	(0.079,0.257)
s66 with s28	0.114	0.040	0.005	0.116	0.042	(0.038,0.201)
s66 with s36	0.070	0.037	0.059	0.069	0.034	(0.006,0.137)
s66 with s42	0.127	0.053	0.016	0.128	0.036	(0.061,0.202)
s69 with s38	0.091	0.041	0.028	0.092	0.039	(0.017,0.171)
s69 with s50	0.145	0.050	0.004	0.148	0.039	(0.074,0.226)

MLM = mean-adjusted maximum likelihood estimator; Bayes = Bayesian estimation; Est = estimate; SE = standard error; SD = standard deviation of the Markov Chain Monte Carlo samples; 95% CI = 95% confidence interval.

Table 4. Standardized estimates (Est) in the second-step analysis of the real-data analysis.

	MLM			Bayes		
Parameter	Est	SE	*p*	Est	SD	95% CI
λ_{11}	0.585	0.045	0.000	0.580	0.047	(0.458,0.654)
λ_{81}	0.652	0.043	0.000	0.650	0.036	(0.575,0.714)
$\lambda_{15,1}$	0.555	0.046	0.000	0.554	0.040	(0.470,0.627)
$\lambda_{22,1}$	0.554	0.048	0.000	0.554	0.040	(0.472,0.627)
$\lambda_{38,1}$	0.362	0.049	0.000	0.359	0.046	(0.265,0.447)
$\lambda_{46,1}$	0.636	0.047	0.000	0.634	0.035	(0.561,0.698)
$\lambda_{54,1}$	0.331	0.044	0.000	0.331	0.049	(0.233,0.424)
$\lambda_{62,1}$	0.639	0.047	0.000	0.639	0.035	(0.566,0.703)
λ_{32}	0.689	0.033	0.000	0.678	0.031	(0.616,0.740)
$\lambda_{10,2}$	0.426	0.044	0.000	0.426	0.044	(0.337,0.508)
$\lambda_{24,2}$	0.631	0.035	0.000	0.631	0.034	(0.560,0.694)
$\lambda_{32,2}$	0.423	0.046	0.000	0.422	0.044	(0.332,0.505)
$\lambda_{40,2}$	0.810	0.022	0.000	0.811	0.024	(0.759,0.855)
$\lambda_{48,2}$	0.591	0.035	0.000	0.590	0.037	(0.513,0.659)
$\lambda_{56,2}$	0.671	0.040	0.000	0.671	0.032	(0.604,0.730)
$\lambda_{64,2}$	0.468	0.045	0.000	0.468	0.042	(0.382,0.547)
λ_{73}	0.414	0.045	0.000	0.406	0.047	(0.316,0.490)
$\lambda_{14,3}$	0.500	0.046	0.000	0.500	0.044	(0.411,0.581)
$\lambda_{28,3}$	0.533	0.045	0.000	0.531	0.042	(0.446,0.609)
$\lambda_{36,3}$	0.512	0.046	0.000	0.513	0.042	(0.426,0.592)
$\lambda_{44,3}$	0.564	0.045	0.000	0.564	0.040	(0.483,0.640)
$\lambda_{52,3}$	0.303	0.049	0.000	0.303	0.050	(0.203,0.398)
$\lambda_{60,3}$	0.553	0.043	0.000	0.554	0.041	(0.468,0.629)
$\lambda_{69,3}$	0.657	0.041	0.000	0.656	0.036	(0.582,0.722)
λ_{54}	0.607	0.044	0.000	0.601	0.038	(0.534,0.678)
$\lambda_{19,4}$	0.664	0.034	0.000	0.664	0.033	(0.594,0.724)
$\lambda_{26,4}$	0.707	0.032	0.000	0.706	0.030	(0.642,0.760)
$\lambda_{34,4}$	0.532	0.047	0.000	0.534	0.040	(0.450,0.608)
$\lambda_{42,4}$	0.521	0.044	0.000	0.520	0.040	(0.436,0.595)
$\lambda_{50,4}$	0.564	0.047	0.000	0.564	0.038	(0.485,0.634)
$\lambda_{58,4}$	0.666	0.037	0.000	0.669	0.032	(0.601,0.728)
$\lambda_{66,4}$	0.473	0.043	0.000	0.472	0.043	(0.386,0.552)
ϕ_{12}	0.362	0.054	0.000	0.364	0.053	(0.255,0.463)
ϕ_{13}	0.726	0.046	0.000	0.724	0.041	(0.641,0.797)
ϕ_{23}	0.381	0.056	0.000	0.383	0.050	(0.281,0.476)
ϕ_{14}	0.837	0.053	0.000	0.834	0.030	(0.767,0.888)
ϕ_{24}	0.440	0.047	0.000	0.442	0.047	(0.346,0.530)
ϕ_{34}	0.737	0.039	0.000	0.734	0.037	(0.659,0.803)

ML = maximum likelihood estimator; Bayes = Bayesian estimation; Est = estimate; SE = standard error; SD = standard deviation of the Markov Chain Monte Carlo samples; 95% CI = 95% confidence interval.

The results derived from the frequentist (MLM) and the Bayesian approaches were consistent. Estimates of loadings, factor variances and covariances for the two methods were almost identical. For example, the standardized factor loadings were between 0.303 and 0.810 using the MLM estimator and between 0.303 and 0.811 using the Bayesian estimation (Table 4). The correlations between the four factors were found to be significantly positive for both methods.

Other features

This package contains other features designed to improve the functionality and user experience of the SEM tool, which we highlight here. Readers may find these features useful for their specific applications.

(1) Parallel computation. The total number of available CPU cores is automatically detected using the detectCores() function. If the CPU has more than one core, two cores will be selected for the two MCMC chains to run separately and simultaneously using the foreach() function from the **doParallel** package (Ooi et al., 2019). Otherwise, the two chains can only run successively such that the run time of the Gibbs sampling doubles. Because the Bayesian Lasso estimation is computationally intensive, parallel computation allows users to accomplish tasks that include a large number of modified residual covariances within a reasonable timeframe.[1]

(2) Missing values. Under the missing-at-random (MAR) assumption for missing values, this package imputes missing values and generates an imputed data set. The block Gibbs sampler proposed by Pan et al. (2017) offers a convenient way to handle the MAR missing response. The conditional distributions of a missing variable use data from other variables to inform the imputation. As an example, in the real-data application, the missing values of the 46th and 60th items in the response of the 84th participant were imputed as 0.000 and −0.009 after data standardization.

(3) Non-normality detection. When the *estimation* is set at *ml* or *both*, the **blcfa** tests whether the item responses are normally distributed. A M*plus* input file (normal.inp) is automatically generated to detect non-normality. As described in the first example, the MLM method will be applied if the data fail to satisfy the multivariate normal assumption. Note that if the environment variable does not include the M*plus* software path, the normal test will not be conducted and the ML estimator will be applied.

(4) In practice, the non-normality test and model estimation are often separately conducted. We provide an example for using the mplus_ml() function in this package to detect non-normality of data and generate the M*plus* file for the CFA in a single step without doing the Bayesian Lasso CFA analysis. If the user is confident of the diagonal structure of the residual variance-covariance matrix, the user can follow the following code:

```
library(blcfa)
setwd(“C:/Users/Desktop/SimuExample/”)
filename = system.file(“extdata”, “simu_data.txt”, package = “blcfa”)
varnames<-c(paste(“y”, 1:10, sep = ““))
usevar <- varnames
myModel<-”
f1 = ~y1 + y2 + y3 + y4 + y5
f2 = ~y6 + y7 + y8 + y9 + y10
“
 mplus_ml(filename, varnames, usevar, myModel)
```

(5) Convergence check. The **blcfa** automatically checks the model convergence with the results of two MCMC chains. Convergence checks are computationally intensive and time-consuming. The package offers users the flexibility to generate a single MCMC chain and obtain estimates without conducting any convergence check:

```
blcfa(filename, varnames, usevar, myModel, estimation = “ml”, MCMAX = 6000, N.burn = 3000, bloutput = TRUE, conver_check = FALSE)
```

The *conver_check* that is *TRUE* by default can be set at *FALSE* to switch off the convergence check. This function is attractive when conducting a simulation study, for example. Another use of this function is when a user wants to check if the model converges with a pre-specified value of *N.burn*. The number of *MCMAX* can be set at a value slightly higher than *N. burn* to reduce run time:

```
blcfa(filename, varnames, usevar, myModel, estimation = “ml”, MCMAX = 3100, N.burn = 3000, bloutput = TRUE)
```

After reaching the appropriate *N.burn* value that is deemed sufficient for model convergence, convergence check can be switched off and the *MCMAX* value increased to obtain accurate estimates.

Discussion

This article offers a didactic discussion of a two-step CFA for simultaneously detecting significant residual covariances and attaining a modified CFA with additional covariance parameter, which is implemented in the R package **blcfa**. Besides explaining the underlying principle of the procedure, we offer a tutorial for the **blcfa** package that implements the two-step approach.

The benefits of **blcfa** can be summarized as follows. Model modification via freeing some of the residual covariance parameters can lead to improved goodness of fit in the CFA without major changes to the factor structure. Indeed, it was shown that proper modifications of residual covariances can reduce bias in structural models (Pan et al., 2017). The package **blcfa** allows users to simultaneously detect all significant residual correlations in one step and fit a CFA without going through the tedious procedure required of the PMM sequential approach. The approach implemented in **blcfa** therefore is also less vulnerable to subjective choice in selecting which MIs to include. By assigning zero to non-significant residual covariances, **blcfa** provides a sparse representation of the covariance structure, which is more interpretable than a covariance matrices with many close-to-zero entries. An important practical benefit of **blcfa** is the integration of the Bayesian Lasso estimation results into M*plus*. Thus, a user can now conduct a CFA without having to switch programs. Finally, a user may also want to take advantage of results from the Bayesian Lasso analysis for model diagnostics. For example, a high value of residual

[1] Parallel computation is not available using Rstudio v1.3.959 or lower on MacOS with R v4.0.0 or higher. This bug was fixed by Rstudio v1.3.1056 released on July 15, 2020. Therefore, Rstudio downloaded after July 15, 2020 or R are suggested when using the current version of the **blcfa** package on MacOS.

covariance in specific cross-loaded item pair could indicate a problem in the specified factor structure.

In the future development of **blcfa**, we plan to incorporate an adaptive version of the lasso method (Zou, 2006), which may improve power in the detection of residual covariance when sample size or/and effect size are small. Additionally, in addition to using the HPD interval method, alternative methods of detecting significance in residual covariance include *p*-values and the threshold method (Feng et al., 2015). Work is currently underway comparing performances of the methods in terms of power and the Type I error rate. Based on the finding, additional features for method of detecting significance in residual covariance will be incorporated into **blcfa**.

Funding

This work was supported by the National Natural Science Foundation of China [NSFC 31871128] and the MOE (Ministry of Education) Project of Humanities and Social Science of China [18YJA190013], and the NSF (National Science Foundation) Grant [SES-1424875] and NIH (National Institute of Health) Grant [1UL1TR001420-01] from the United States.

ORCID

Lijin Zhang http://orcid.org/0000-0002-4222-8850
Junhao Pan http://orcid.org/0000-0001-7156-6077
Edward Haksing Ip http://orcid.org/0000-0002-4811-4205

References

Asparouhov, T., & Muthén, B. O. (2010). *Bayesian analysis using Mplus: Technical implementation* (Technical appendix). Los Angeles, CA: Muthén & Muthén. https://www.statmodel.com/download/Bayes3.pdf

Byrne, B. M. (2012). *Structural equation modeling with Mplus: Basic concepts, applications, and programming*. Routledge.

Byrne, B. M., & Watkins, D. (2003). The issue of measurement invariance revisited. *Journal of Cross-cultural Psychology*, *34*, 155–175. https://doi.org/10.1177/0022022102250225

Feng, X., Wu, H., & Song, X. (2015). Bayesian adaptive Lasso for ordinal regression with latent variables. *Sociological Methods & Research*, *46*, 1–28. https://doi.org/10.1177/0049124115610349

Gelman, A. (1996). Inference and monitoring convergence. In W. R. Gilks, S. Richardson, & D. J. Spiegelharter (Eds.), *Markov Chain Monte Carlo in Practice* (pp. 131–144). Chapman & Hall.

Hallquist, M. N., & Wiley, J. F. (2018). MplusAutomation: An R package for facilitating large-scale latent variable analyses in Mplus. *Structural Equation Modeling*, *25*, 621–638. https://doi.org/10.1080/10705511.2017.1402334

Khondker, Z. S., Zhu, H., Chu, H., Lin, W., & Ibrahim, J. G. (2013). The Bayesian covariance lasso. *Statistics and Its Interface*, *6*, 243–259. https://doi.org/10.4310/SII.2013.v6.n2.a8

Marsh, H. W. (1992). *Self Description Questionnaire (SDQ) I: A theoretical and empirical basis for the measurement of multiple dimensions of preadolescent self-concept: A test manual and research monograph. Macarthur*. Faculty of Education, University of Western Sydney.

Muthén, B. O. (2010). *Bayesian analysis in Mplus: A brief introduction*. Los Angeles, CA: Muthén & Muthén. https://www.statmodel.com/download/IntroBayesVersion%203.pdf.

Muthén, L. K., & Muthén, B. O. (1998-2017). *Mplus userś guide*. Eighth edn. Muthén & Muthén.

Ooi, H., Weston, S., & Tenenbaum, D. (2019). *doParallel: Foreach Parallel Adaptor for the 'parallel' Package [Computer software manual]. (R package version 1.0.15)*. https://CRAN.R-project.org/package=doParallel

Pan, J. H., Ip, E. H., & Dubé, L. (2017). An alternative to post hoc model modification in confirmatory factor analysis: The Bayesian lasso. *Psychological Methods*, *22*, 687–704. https://doi.org/10.1037/met0000112

Park, T., & Casella, G. (2008). The Bayesian Lasso. *Journal of the American Statistical Association*, *103*, 681–686. https://doi.org/10.1198/016214508000000337

Sörbom, D. (1989). Model modification. *Psychometrika*, *54*, 371–384. https://doi.org/10.1007/BF02294623

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, *58*, 267–288. https://doi.org/10.1111/j.1467-9868.2011.00771.x

Wang, H. (2012). Bayesian graphical lasso models and efficient posterior computation. *Bayesian Analysis*, *7*, 867–886. https://doi.org/10.1214/12-BA729

West, S. G., Finch, J. F., & Curran, P. J. (1995). Structural equation models with non-normal variables. In R. H. Hoyle (Ed.), *Structural Equation Modeling* (pp. 56–75). Sage Publications, Inc.

Wickham, H., Hester, J., & Winston., C., RStudio., & R Core team. (2019). *devtools: Tools to make developing R packages easier [Computer software manual]. (R package version 2.2.1)*. https://CRAN.R-project.org/package=devtools

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, *101*, 1418–1429. https://doi.org/10.1198/016214506000000735