`lsttheory`: An R Package for Fast Computation of State Trait Models

Axel Mayer
Ghent University

February 23, 2014

Abstract

This R (?, ?) package is a supplement for the article 'A Theory of States and Traits – Revised' (SMGC; Steyer, Mayer, Geiser, & Cole, in press). It is based on the structural equation modeling R package `lavaan` (Rosseel, 2012) and provides a convenient interface to compute some common models of the revised latent state-trait theory (LST-R theory). The main function of the package allows for easy specification of multistate, multistate-singletrait, and multistate-multitrait models. It automatically generates `lavaan` syntax for these models, runs the models, and returns model estimates together with reliability, occasion specificity, and consistency coefficients for the respective models.

Keywords: states, traits, reliability, occasion specificity, consistency

Cautionary Note: The package is currently under development and some things may change in the future. We are at an early stage of development and it is likely that the structure and key aspects of the two packages will change. We also plan to release the package on CRAN, once we are a bit furhter in the develpment process. Please report any bugs.

Contents

1 Introduction

lsttheory allows for easy specification of multistate, multistate-singletrait, and multistate-multitrait models. This vignette is structured as follows: We first describe the installation process in detail for nonexperienced users of R. Users who are familiar with package installation from local .zip files or from source (via Github) may wish to skip this section. We then present various kinds of LST-R theory models with syntax and model results.

2 Installation

The R package lsttheory can be installed from a local file. Therefore, we need to install the dependencies first. The installation has been tested under Windows 7 and under Linux (Ubuntu 11.10). It should also work under Mac OS X but we haven't tested it yet. Please make sure that you are using R version 3.0.1 or higher.

2.1 Windows Installation

For a Windows installation (without Rtools) we suggest installing the dependencies from a CRAN mirror first by executing

```
install.packages("lavaan")  ## for lsttheory
install.packages("shiny")  ## for user Interface
install.packages("semPlot")  ## for plots
```

in the console and then selecting a mirror next to you. After that, the R package lsttheory can be installed using the Windows binary file (with file ending .zip) as follows:

```
install.packages("D:/workspace/lsttheory_0.1-1.zip")
```

Please adjust the file path and version number accordingly.

2.2 Linux Installation

We assume that Linux users are familiar with installing R packages from source. The source files is lsttheory_0.1-1.tar.gz and can be downloaded from GitHub.

2.3 Loading the Package

After having succesfully installed the package, we need to load them:

```r
library("lsttheory")

## Loading required package:  lavaan
## This is lavaan 0.5-16
## lavaan is BETA software!  Please report any bugs.
```

3 Multistate Models

The lsttheory pacakge contains several simulated example datasets. The first one that we use is the dataset `multistate`. It contains 4 manifest variables Y_{it}, where the index refers to the ith manifest variable assessed at occasion t. This dataset has been used in Supplement C of Steyer, Mayer, Geiser & Cole (in press) to describe software input for `lavaan` for LST-R models.

```
data(multistate)
head(round(multistate, 2))

##     y11   y21   y12   y22
## 1  3.03  3.76  4.15  5.26
## 2 -1.95 -1.11 -1.99 -0.52
## 3  0.94  0.53  1.10  1.85
## 4  2.44  2.45  1.36  2.03
## 5  3.72  3.59  0.33  1.56
## 6  3.16  3.64  2.25  2.79
```

3.1 Multistate Model with η_t-Congenericity

First, we use this dataset to fit a multistate model with η_t-congenericity and conditional mean independence (see Box 4.1 of SMGC). The main function of our package to be called by the user is `lsttheory`. See `?lsttheory` for details. It is used to fit all models. The multistate model with η_t-congenericity can be specified as follows:

```
m1 <- lsttheory(neta = 2, data = multistate)
print(m1)

##
##  Multistate Model
##
##     rely spey cony
## y11 0.92   NA   NA
## y21 0.92   NA   NA
## y12 0.93   NA   NA
## y22 0.91   NA   NA
##
##
```

The lsttheory function just requires two mandatory arguments: The number of common state variables η_t and the dataset to use. In the current version of our package, the dataset may only include the manifest variables Y_{it} and these should be ordered by occasion t and indicator i, i.e., $Y_{11}, Y_{21}, \ldots, Y_{12}, Y_{22}, \ldots, Y_{13}, Y_{23}, \ldots$.

The lsttheory function returns an object of class lsttheory for which several methods are available. `print(m1)` shows reliability, occasion specificity, and consistency coefficients (see Box 3.1 of SMGC). For the multistate model only reliability coefficients are available, because traits are not modeled.

The function lsttheory has automatically generated lavaan input syntax:

```
cat(m1@lavaansyntax)

## eta1 =~ 1*y11
## eta1 =~ la211*y21
## eta2 =~ 1*y12
## eta2 =~ la221*y22
## y11 ~ 0*1
## y21 ~ la210*1
## y12 ~ 0*1
## y22 ~ la220*1
## eta1 ~ ga10*1
## eta2 ~ ga20*1
## y11 ~~ eps11*y11
## y21 ~~ eps21*y21
## y12 ~~ eps12*y12
## y22 ~~ eps22*y22
## eta1 ~~ psi1*eta1
## eta2 ~~ psi2*eta2
##
##
##
##
## vareta1 := psi1
## vareta2 := psi2
## vary11 := 1^2 * vareta1 + eps11
## vary21 := la211^2 * vareta1 + eps21
## vary12 := 1^2 * vareta2 + eps12
## vary22 := la221^2 * vareta2 + eps22
## rely11 := 1^2 * vareta1 / vary11
## rely21 := la211^2 * vareta1 / vary21
## rely12 := 1^2 * vareta2 / vary12
## rely22 := la221^2 * vareta2 / vary22
```

and the lavaan output can be seen by calling:

```
summary(m1@lavaanres)

## lavaan (0.5-16) converged normally after  38 iterations
##
```

```
##   Number of observations                           400
##
##   Estimator                                         ML
##   Minimum Function Test Statistic                0.894
##   Degrees of freedom                                 1
##   P-value (Chi-square)                           0.344
##
## Parameter estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
##                    Estimate  Std.err  Z-value  P(>|z|)
## Latent variables:
##   eta1 =~
##     y11               1.000
##     y21    (l211)     0.968    0.030   32.385    0.000
##   eta2 =~
##     y12               1.000
##     y22    (l221)     0.952    0.029   33.020    0.000
##
## Covariances:
##   eta1 ~~
##     eta2              2.240    0.208   10.794    0.000
##
## Intercepts:
##     y11               0.000
##     y21    (l210)     0.518    0.038   13.504    0.000
##     y12               0.000
##     y22    (l220)     0.732    0.045   16.215    0.000
##     eta1   (ga10)     0.466    0.092    5.076    0.000
##     eta2   (ga20)     0.952    0.095    9.989    0.000
##
## Variances:
##     y11    (ep11)     0.261    0.074
##     y21    (ep21)     0.265    0.070
##     y12    (ep12)     0.254    0.079
##     y22    (ep22)     0.285    0.072
##     eta1   (psi1)     3.103    0.248
##     eta2   (psi2)     3.380    0.268
##
## Defined parameters:
##     vareta1           3.103    0.248   12.523    0.000
##     vareta2           3.380    0.268   12.633    0.000
##     vary11            3.365    0.238   14.142    0.000
```

```
##     vary21             3.176    0.225   14.142    0.000
##     vary12             3.634    0.257   14.142    0.000
##     vary22             3.347    0.237   14.142    0.000
##     rely11             0.922    0.023   40.825    0.000
##     rely21             0.917    0.023   40.472    0.000
##     rely12             0.930    0.022   42.008    0.000
##     rely22             0.915    0.022   41.052    0.000
```

The slot lavaanres in the m1 object contains the fitted lavaan object of class lavaan. See `?"lavaan-class"` for more information and available methods.

3.2 Multistate Model with Essential η_t-Equivalence

The default setting of the lsttheory function is to assume η_t-congenericity. If we want to assume essential η_t-equivalence, we need to specify an additional argument, namely the `equiv.assumption` argument, which is a list of equivalence assumptions. For the multistate model, the theta argument will be ignored. By specifying `tau="ess"`, we assume essential η_t-equivalence:

```
m1 <- lsttheory(neta = 2, data = multistate, equiv.assumption = list(tau = "ess",
    theta = "equi"))
coef(m1@lavaanres, type = "user")

##   eta1=~y11  eta1=~y21  eta2=~y12  eta2=~y22      y11~1      la210
##       1.000      1.000      1.000      1.000      0.000      0.503
##       y12~1      la220       ga10       ga20      eps11      eps21
##       0.000      0.687      0.466      0.952      0.313      0.217
##       eps12      eps22       psi1       psi2 eta1~~eta2    vareta1
##       0.343      0.204      2.997      3.198      2.132      2.997
##     vareta2     vary11     vary21     vary12     vary22     rely11
##       3.198      3.310      3.214      3.541      3.402      0.906
##      rely21     rely12     rely22
##       0.933      0.903      0.940
```

3.3 Multistate Model with η_t-Equivalence

Similarly, if we want to assume η_t-equivalence, we specify the equivalence assumption as follows:

```
m1 <- lsttheory(neta = 2, data = multistate, equiv.assumption = list(tau = "equi",
    theta = "equi"))
coef(m1@lavaanres, type = "user")

##   eta1=~y11  eta1=~y21  eta2=~y12  eta2=~y22      y11~1      y21~1
```

```
##      1.000      1.000      1.000      1.000      0.000      0.000
##      y12~1      y22~1       ga10       ga20      eps11      eps21
##      0.000      0.000      0.762      1.383      0.462      0.320
##      eps12      eps22       psi1       psi2 eta1~~eta2    vareta1
##      0.639      0.379      2.936      3.088      2.132      2.936
##    vareta2     vary11     vary21     vary12     vary22     rely11
##      3.088      3.398      3.256      3.727      3.467      0.864
##     rely21     rely12     rely22
##      0.902      0.829      0.891
```

3.4 Multistate Models with Scale Invariance

In order to add scale invariance assumptions over time, we need to specify the scale.invariance argument. The default is not to assume scale invariance. The scale invariance argument is a list of four entries. For the multistate models, only the first and the second entry are relevant. The first entry refers to scale invariance of intercepts and the second entry refers to scale invariance of loadings. For example, if we want to specify a multistate model with η_t-congenericity and scale invariance of intercepts and loadings, the function call is:

```
m1 <- lsttheory(neta = 2, data = multistate, scale.invariance = list(lait0 = TRUE,
    lait1 = TRUE, lat0 = TRUE, lat1 = TRUE))
coef(m1@lavaanres, type = "user")

##  eta1=~y11      la211  eta2=~y12      la211      y11~1      la210
##      1.000      0.979      1.000      0.979      0.000      0.608
##      y12~1      la210       ga10       ga20      eps11      eps21
##      0.000      0.608      0.415      1.007      0.282      0.254
##      eps12      eps22       psi1       psi2 eta1~~eta2    vareta1
##      0.310      0.243      3.068      3.280      2.194      3.068
##    vareta2     vary11     vary21     vary12     vary22     rely11
##      3.280      3.350      3.193      3.590      3.385      0.916
##     rely21     rely12     rely22
##      0.920      0.914      0.928
```

Of course, the scale invariance argument can also be used for a multistate model with essential η_t-equivalence. Then, the lait1 entry is ignored.

```
m1 <- lsttheory(neta = 2, data = multistate, equiv.assumption = list(tau = "ess",
    theta = "equi"), scale.invariance = list(lait0 = TRUE, lait1 = TRUE, lat0 = TRUE,
    lat1 = TRUE))
```

For a multistate model with η_t-equivalence, all scale invariance settings are ignored.

4 Multistate-Singletrait Models

4.1 Multistate-Singletrait Models with θ-Congenericity

The same function lsttheory can also be used to fit multistate-singletrait models in LST-R theory. We only need to specify that there is one θ variable in addition to the specification of the corresponding multistate model. The following syntax specifies a multistate-singletrait model with these assumptions:

- η_t-congenericity (Box 4.1 of SMGC)
- conditional mean independence (Box 4.1 of SMGC)
- θ-congenericity (Box 5.1 of SMGC)

```
m1 <- lsttheory(neta = 3, ntheta = 1, data = multistate02)
print(m1)

##
##  Singletrait-Multistate Model
##
##     rely spey cony
## y11 0.73 0.25 0.47
## y21 0.83 0.29 0.54
## y31 0.64 0.22 0.42
## y12 0.77 0.20 0.57
## y22 0.81 0.21 0.60
## y32 0.65 0.17 0.48
## y13 0.74 0.26 0.48
## y23 0.79 0.28 0.51
## y33 0.68 0.24 0.44
##
##
```

Note that we use the `multistate02` data set for this model – it contains three indicators at three occasions of measurement. We now also get estimates for the occasion specificity and consistency coefficents in addition to the reliability coefficients. To see all parameters of the model:

```
coef(m1@lavaanres, type = "user")

##    eta1=~y11        la211        la311    eta2=~y12        la221
##        1.000        1.221        0.801        1.000        1.211
##        la321    eta3=~y13        la231        la331        y11~1
##        0.772        1.000        1.233        0.828        0.000
##        la210        la310        y12~1        la220        la320
```

```
##         0.217        0.570        0.000        0.212        0.578
##        y13~1        la230        la330       eta1~1         ga20
##        0.000        0.262        0.574        0.000        0.400
##         ga30        eps11        eps21        eps31        eps12
##       -0.115        1.070        0.874        1.029        0.932
##        eps22        eps32        eps13        eps23        eps33
##        1.036        0.956        1.028        1.146        0.944
##         psi1         psi2         psi3  theta1=~eta1        ga21
##        0.988        0.792        1.014        1.000        1.099
##         ga31    vartheta1      mtheta1      vareta1      vareta2
##        0.999        1.857        0.611        2.845        3.035
##      vareta3       vary11       vary21       vary31       vary12
##        2.869        3.914        5.116        2.853        3.967
##       vary22       vary32       vary13       vary23       vary33
##        5.483        2.765        3.897        5.511        2.912
##       rely11       rely21       rely31       rely12       rely22
##        0.727        0.829        0.639        0.765        0.811
##       rely32       rely13       rely23       rely33       spey11
##        0.654        0.736        0.792        0.676        0.252
##       spey21       spey31       spey12       spey22       spey32
##        0.288        0.222        0.200        0.212        0.171
##       spey13       spey23       spey33       cony11       cony21
##        0.260        0.280        0.239        0.474        0.541
##       cony31       cony12       cony22       cony32       cony13
##        0.417        0.565        0.599        0.483        0.476
##       cony23       cony33
##        0.512        0.437
```

4.2 Multistate-Singletrait Models with θ-Equivalence

We don't show all possible combinations of assumptions. We just give one more example of a multistate-singletrait model with this set of assumptions:

- essential η_t-equivalence (Box 4.1 of SMGC)
- scale invariance over time
- conditional mean independence (Box 4.1 of SMGC)
- θ-equivalence (Box 5.1 of SMGC)

```r
m1 <- lsttheory(neta = 3, ntheta = 1, data = multistate02, equiv.assumption = list(tau = "es
    theta = "equi"), scale.invariance = list(lait0 = TRUE, lait1 = TRUE, lat0 = TRUE,
    lat1 = TRUE))
coef(m1@lavaanres, type = "user")
```

```
##     eta1=~y11     eta1=~y21     eta1=~y31     eta2=~y12     eta2=~y22
##         1.000         1.000         1.000         1.000         1.000
##     eta2=~y32     eta3=~y13     eta3=~y23     eta3=~y33         y11~1
##         1.000         1.000         1.000         1.000         0.000
##         la210         la310         y12~1         la210         la310
##         0.385         0.421         0.000         0.385         0.421
##         y13~1         la210         la310        eta1~1        eta2~1
##         0.000         0.385         0.421         0.000         0.000
##        eta3~1         eps11         eps21         eps31         eps12
##         0.000         1.064         1.381         0.961         1.004
##         eps22         eps32         eps13         eps23         eps33
##         1.591         0.878         1.090         1.673         0.831
##          psi1          psi2          psi3 theta1=~eta1 theta1=~eta2
##         0.861         0.939         0.963         1.000         1.000
## theta1=~eta3     vartheta1       mtheta1       vareta1       vareta2
##         1.000         1.855         0.727         2.716         2.794
##       vareta3        vary11        vary21        vary31        vary12
##         2.817         3.780         4.097         3.677         3.797
##        vary22        vary32        vary13        vary23        vary33
##         4.384         3.672         3.908         4.490         3.648
##        rely11        rely21        rely31        rely12        rely22
##         0.719         0.663         0.739         0.736         0.637
##        rely32        rely13        rely23        rely33        spey11
##         0.761         0.721         0.627         0.772         0.228
##        spey21        spey31        spey12        spey22        spey32
##         0.210         0.234         0.247         0.214         0.256
##        spey13        spey23        spey33        cony11        cony21
##         0.246         0.214         0.264         0.491         0.453
##        cony31        cony12        cony22        cony32        cony13
##         0.504         0.488         0.423         0.505         0.475
##        cony23        cony33
##         0.413         0.508
```

5 Multistate-Doubletrait Models

For the mulistate doubletrait models, we need to use a different data set, because we need at least two common state variables for each of the θ variables. The simulated data set is called **multitraitmultistate** and contains 8 manifest variables Y_{it} distributed across 4 occasions of measurement:

```
data(multitraitmultistate)
head(round(multitraitmultistate, 2))

##     y11   y21   y12   y22   y13   y23   y14   y24
## 1  1.55  1.99  0.63  0.02  1.61  2.26  4.85  3.55
## 2  1.92  3.43 -0.66 -0.58  2.81  4.27  1.58  0.97
## 3 -0.07  0.32  1.81  1.83  1.73  4.05  0.70  2.93
## 4 -0.67 -1.67  1.01  1.55 -1.79 -2.35 -0.67  0.79
## 5  0.53  0.65  0.11 -0.47 -1.10 -1.03  2.91 -0.56
## 6 -1.90 -2.47  1.46  3.04  1.02 -0.08  0.88  1.39
```

5.1 Multistate-Doubletrait Models with θ_1, θ_2-Congenericity

The first model that we want to show with this dataset is a multistate-doubletrait model with these assumptions:

- η_t-congenericity (Box 4.1 of SMGC)
- conditional mean independence (Box 4.1 of SMGC)
- θ_1-congenericity (Box 6.1 of SMGC)
- θ_2-congenericity (Box 6.1 of SMGC)

The model syntax is:

```
m1 <- lsttheory(neta = 4, ntheta = 2, data = multitraitmultistate)
coef(m1@lavaanres, type = "user")

##      eta1=~y11          la211      eta2=~y12          la221      eta3=~y13
##          1.000          1.203          1.000          1.195          1.000
##          la231      eta4=~y14          la241          y11~1          la210
##          1.204          1.000          1.080          0.000          0.313
##          y12~1          la220          y13~1          la230          y14~1
##          0.000          0.380          0.000          0.307          0.000
##          la240          eta1~1           ga20          eta3~1           ga40
##          0.382          0.000          0.398          0.000          0.234
##          eps11          eps21          eps12          eps22          eps13
##          0.941          1.020          1.136          0.857          0.915
```

```
##           eps23           eps14           eps24            psi1            psi2
##           1.038           0.901           1.158           0.955           1.141
##            psi3            psi4    theta1=~eta1            ga21    theta2=~eta3
##           1.390           0.681           1.000           0.649           1.000
##            ga41       vartheta1       vartheta2         mtheta1         mtheta2
##           0.858           2.694           2.580           0.450           0.928
## theta1~~theta2         vareta1         vareta2         vareta3         vareta4
##           1.721           3.649           2.277           3.970           2.583
##          vary11          vary21          vary12          vary22          vary13
##           4.590           6.303           3.413           4.106           4.885
##          vary23          vary14          vary24          rely11          rely21
##           6.795           3.484           4.173           0.795           0.838
##          rely12          rely22          rely13          rely23          rely14
##           0.667           0.791           0.813           0.847           0.741
##          rely24          spey11          spey21          spey12          spey22
##           0.722           0.208           0.219           0.334           0.397
##          spey13          spey23          spey14          spey24          cony11
##           0.284           0.297           0.196           0.191           0.587
##          cony21          cony12          cony22          cony13          cony23
##           0.619           0.333           0.395           0.528           0.551
##          cony14          cony24
##           0.546           0.532
```

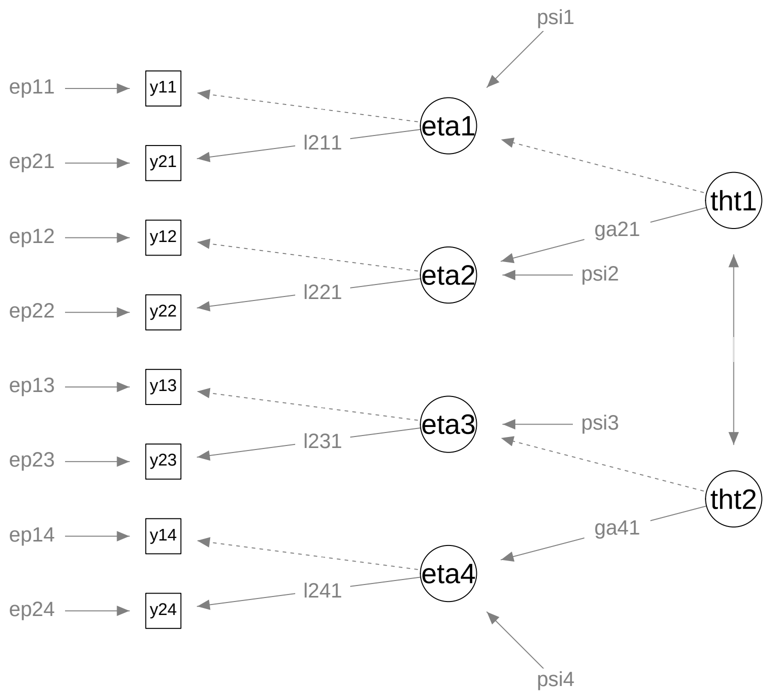

Figure 1: Multistate-Doubletrait Model.

6 Plot LST-R Theory Models with `semPlot`

The package **`semPlot`** by Sacha Epskamp can be used to plot such LST-R models as shown in this vignette. To plot the multistate-doubletrait model shown in last section, we call:

```
library(semPlot)
semPaths(m1@lavaanres, style = "lisrel", intercepts = F, layout = "tree2", rotation = 4,
    nCharNodes = 4, nCharEdges = 4, optimizeLatRes = F, residScale = 10)
```

which gives the figure shown in Figure 1.

References

Epskamp, S. (2013). semPlot: Path diagrams and visual analysis of various SEM packages' output. R package version 0.3.2. http://CRAN.R-project.org/package=semPlot

R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Rosseel, Y. (2012). lavaan: An R Package for Structural Equation Modeling. Journal of Statistical Software, 48(2), 1–36. URL http://www.jstatsoft.org/v48/i02/.

Steyer, R., Mayer, A., Geiser, C., & Cole, D.A. (in press). A theory of states and traits – revised. Annual Review of Clinical Psychology.