blavaan: Bayesian Latent Variable Models with Stan and JAGS

Ed Merkle

Psychoco 2022
Collaborators and contributors related to this talk:
  ▶ Ellen Fitzsimmons, Missouri
  ▶ Daniel Furr, Berkeley
  ▶ Mauricio Garnier-Villareal, Amsterdam (Vrije U)
  ▶ Ben Goodrich, Columbia
  ▶ Terrence Jorgensen, Amsterdam (UvA)
  ▶ Sophia Rabe-Hesketh, Berkeley
  ▶ Yves Rosseel, Ghent
  ▶ James Uanhoro, North Texas
  ▶ Institute of Education Sciences (Grant R305D210044)
  ▶ Psychoco!
Introduction

- **blavaan**: An R package for Bayesian SEM, making use of `lavaan`, JAGS and Stan.

- Initial goal: Automatically generate JAGS code from a `lavaan` object (focus on Bollen Political Democracy model).

- Subsequent goals are based on tricky problems encountered during development, such as speed/efficiency of MCMC estimation.
Talk outline

- Brief introduction to \textit{blavaan}
- How \textit{blavaan} works
- “Advanced” features
- Future directions and conclusions
Package introduction
Why SEM?

This family of models overlaps with mixed models, time series models, models to assess causality, item response models, etc.

So, if you find good strategies for estimating these models, you have found good strategies for estimating many other models.
Why Bayesian SEM?

- Include prior information/expectations in analyses
- Handle uncertainty: Ease of describing uncertainty in key results (latent variables, functions of parameters)
- Flexibility/extensibility: As models increase in complexity, Bayesian methods can be easier to extend to new situations
blavaan

- *blavaan* is intended to work like *lavaan*, with some additional Bayesian options.

- This means that, if you already know how to do something in *lavaan*, you can probably also do something in *blavaan*. 
Path diagram

\[ \eta_1 \]
\[ \eta_2 \]

\[ y_1 \]
\[ y_2 \]
\[ y_3 \]
\[ y_4 \]
\[ y_5 \]
\[ y_6 \]
Model specification and estimation in *lavaan*:

```r
library("lavaan")

HS.model <- ' visual =~ x1 + x2 + x3
               verbal =~ x4 + x5 + x6 '

fit <- cfa(HS.model, data = HolzingerSwineford1939)
```
If you use all the defaults, `blavaan` is almost exactly the same:

```r
library("blavaan")

HS.model <- ' visual  =~ x1 + x2 + x3
             verbal =~ x4 + x5 + x6 '

bfit <- bcfa(HS.model, data = HolzingerSwineford1939)
```
But you shouldn’t rely on defaults! *blavaan* provides functionality for things like

- Choosing number of burnin (warmup) and sampling iterations.
- Specifying your own prior distributions.
- Sampling the latent variables, along with other parameters.
- Assessing chain convergence and plotting results.
Example without defaults:

```
library("blavaan")

HS.model <- ' visual =~ x1 + prior("normal(1,1)")*x2 + x3
  verbal =~ x4 + x5 + x6 ''

bfit <- bcfa(HS.model, data = HolzingerSwineford1939,
  dp = dpriors(lambda = "normal(1,5)",
  burnin = 500, sample = 500, n.chains = 4,
  save.lvs = TRUE,
  bcontrol = list(cores = 4))
```
blavaan

(credit to Richard McElreath)
How it works
Workflow

- How *blavaan* originally worked ($\approx 2016$):
  - Obtain a parameter table from a *lavaan* model specification
  - Write JAGS/Stan code that is specific to that model
  - Convert observed data to necessary JAGS/Stan format
  - Run MCMC in JAGS or Stan
  - Summarize the results (posterior point estimates/SDs, ppp, information criteria)
The original model estimation strategies treated latent variables as model parameters. This can be advantageous because, conditioned on latent variables, observed variables are often independent (leading to univariate distributions instead of multivariate distributions).

But there are often many latent variables, and each person has their own latent variables. This can lead to a parameter explosion and inefficiency.
New workflow

For a video presentation to the University of Kansas on 2018-05-04

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
<th>Author</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>get working for presentation</td>
<td>bgoodri</td>
<td>last year</td>
</tr>
<tr>
<td>inst/include</td>
<td>minimal version of R package</td>
<td>bgoodri</td>
<td>last year</td>
</tr>
<tr>
<td>src</td>
<td>get working for presentation</td>
<td>bgoodri</td>
<td>last year</td>
</tr>
<tr>
<td>tools</td>
<td>minimal version of R package</td>
<td>bgoodri</td>
<td>last year</td>
</tr>
<tr>
<td>.Rbuildignore</td>
<td>get working for presentation</td>
<td>bgoodri</td>
<td>last year</td>
</tr>
<tr>
<td>.gitignore</td>
<td>minimal version of R package</td>
<td>bgoodri</td>
<td>last year</td>
</tr>
<tr>
<td>Bollen.png</td>
<td>upload PNGs</td>
<td>bgoodri</td>
<td>last year</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>get working for presentation</td>
<td>bgoodri</td>
<td>last year</td>
</tr>
<tr>
<td>LERSIL.Rmd</td>
<td>get working for presentation</td>
<td>bgoodri</td>
<td>last year</td>
</tr>
<tr>
<td>LERSIL.html</td>
<td>get working for presentation</td>
<td>bgoodri</td>
<td>last year</td>
</tr>
</tbody>
</table>
To avoid parameter explosion, we can work with the marginal likelihood (marginal over latent variables).

We can still sample latent variables, but they are not official model parameters (sampled in Stan’s “generated quantities” block, or in R after model estimation).

So we are using the same likelihood as the frequentists, but MCMC affords us a different set of tricks for model estimation.
New workflow

- Instead of writing Stan code that is unique to a user’s model, we now have one Stan file that handles (almost) any model the user requests.

- We compile this file once during package installation, and reuse it for all models.

- This reduces some flexibility (in, e.g., prior distributions) but avoids the need to compile for each model estimation. Users can also modify the Stan file if desired.

- Older *blavaan* approaches are still available, via `target = "jags"` and `target = "stanclassic"`. 
Estimation efficiencies (from 2021 JSS paper)
Advanced features
“Advanced” features

- There is a large Bayesian ecosystem within R. *blavaan* can often make use of other packages to provide model metrics/assessments that are difficult or impossible outside of R.

- Examples include information criteria for model comparison (including ordinal models), and general posterior predictive assessment.
Information criteria

- Package `loo` provides methods for computing WAIC and leave-one-out cross-validation metrics for model comparison.

- For these, we should supply *marginal* likelihoods of the estimated model, which are not always available from Bayesian SEMs (see Merkle, Furr, Rabe-Hesketh, 2019).

- `blavaan` automates these computations, allowing for model comparisons that incorporate uncertainty.
Information criteria

hsm1 <- ' visual  =~ x1 + x2 + x3 + x4
textual =~ x4 + x5 + x6
speed   =~ x7 + x8 + x9 '

fit1 <- bcfa(hsm1, data=HolzingerSwineford1939)

hsm2 <- ' visual  =~ x1 + x2 + x3
textual =~ x4 + x5 + x6 + x7
speed   =~ x7 + x8 + x9 '

fit2 <- bcfa(hsm2, data=HolzingerSwineford1939)
Information criteria

blavCompare(fit1, fit2)

##
## WAIC estimates:
## object1: 7540.766
## object2: 7541.606
##
## WAIC difference & SE:
## -0.420 1.375
##
## LOO estimates:
## object1: 7540.861
## object2: 7541.758
##
## LOO difference & SE:
## -0.449 1.383
##
## Laplace approximation to the log-Bayes factor
## (experimental; positive values favor object1): 0.746
Ordinal models

- Models with ordinal observed variables are a recent addition (2 weeks on CRAN). We use a data augmentation strategy for estimation, augmenting ordinal data with underlying, continuous values.

- Marginal likelihood computations are more complicated for these models, requiring us to evaluate (approximate) the CDF of a multivariate normal.

- *blavaan* currently uses an importance sampling approach from package *tmvnsim* to evaluate this CDF, after model estimation.
Ordinal version of Holzinger-Swineford data:

```r
m1 <- bcfa(HS.model, data = hs39, ordered = TRUE,
            dp = dpriors(lambda = "normal(1, .5)"),
            bcontrol = list(cores = 3),
            mcmceextra = list(data = list(llnsamp = 50)))
```
Example

fitMeasures(m1)

## blavaan NOTE: These criteria involve likelihood approximations that may be imprecise.
## You could try running the model again to see how much the criteria fluctuate.
## You can also manually set llnsamp for greater accuracy (but also greater runtime).

## npar logl ppp bic dic p_dic waic
## 30.000 -1837.227 0.510 3833.254 3717.351 21.448 3741.129
## p_waic se_waic looic p_loo se_loo margloglik
## 43.555 35.495 3741.821 43.901 35.550 NA
Posterior assessment

- Along with information criteria, \textit{blavaan} allows for posterior predictive assessments involving any user-defined function.

- Functionality is available via \texttt{ppmc()}, contributed by Terrence Jorgensen.
Posterior assessment

▶ Example: Posterior predictive assessment of item-total correlation in ordinal SEM, as described by Bonifay & Depaoli, 2022.

▶ Compare the observed item-total correlations to the model’s posterior predictive distributions of item-total correlations.
Posterior predictive assessment of item-total correlations:

```r
itemtot <- function(fit) {
  tmpdata <- lavInspect(fit, "data")
  sapply(1:ncol(tmpdata),
     function(i) cor(tmpdata[,i], rowSums(tmpdata[, -i])))
}

out1 <- ppmc(m1, discFUN = itemtot)
```
```r
summary(out1, dist="sim", central.tendency="mean")
```

##
## Posterior summary statistics and highest posterior density (HPD) 95% credible intervals for the posterior predictive distribution (PPP).
##
## EAP   SD lower upper   PPP_sim_GreaterThan_obs   PPP_sim_LessThan_obs
## 1 0.233 0.080 0.077 0.382        0.381                  0.619
## 2 0.235 0.082 0.074 0.391        0.428                  0.572
## 3 0.242 0.083 0.077 0.401        0.594                  0.406
## 4 0.211 0.084 0.044 0.373        0.566                  0.434
## 5 0.221 0.084 0.065 0.393        0.623                  0.377
## 6 0.200 0.082 0.033 0.348        0.572                  0.428
## 7 0.189 0.085 0.011 0.348        0.679                  0.321
## 8 0.191 0.086 0.021 0.353        0.125                  0.875
## 9 0.180 0.082 0.021 0.340        0.733                  0.267
Future & conclusions
Conclusions

- So far, blavaan has led to some improvements and tightening in Bayesian SEM estimation and model comparison. It has also provided other researchers with tools for developing/implementing new procedures.

- Current/future development is supported by the Institute of Education Sciences, U.S. Department of Education.
Future

- The near future: Refinement of ordinal SEM; multilevel SEM (start of 2023?)

- Other possibilities
  - Parallelization in Stan
  - Latent variable interactions and quadratic effects
  - Modeling framework closer to GLLAMM
  - Your contribution!
Some References


Other References


▶ Bonifay W, Depaoli S (in press). Model evaluation in the presence of categorical data: Bayesian model checking as an alternative to traditional methods. *Prevention Science*.


Thank you!

Try it yourself:

`install.packages("blavaan")`

Further information:

https://ecmerkle.github.io/blavaan/