Lecture 3: Bayes + LVM/FA

Jeff Rouder

June, 2024

Why Bayes For FA and SEM?

- lavaan is quick, easy, seamless, well documented
- you certainly don't need me here to show you how to use it
- And Bayes FA/SEM
    - not quick
    - not easy
    - not seamless
    - not well documented

Why Bayes For FA and SEM

1. Fixes Heywood cases.

- Variances are negative
- Correlations are bigger than 1.0 or negative

Why Bayes For FA and SEM

2. Most of our data are not cute little matrices $\boldsymbol{Y}_{[I \times J]}$
3. Preprocessing the data to make $\boldsymbol{Y}$

- could be simple, like aggregation
- could be complicated, like deriving a drift rate in a diffusion model
- resulting $\boldsymbol{Y}$ in real data is often too noisy to support FA/SEM

4. Needed: a fully integrated approach where FA affects preprocessing and preprocessing affects FA.
5. Enter Bayes

Libraries

```r
library(corrplot)
library(mvtnorm)
library(R2jags)
library(infinitefactor)
library(abind)
```

Bayes For Cute Little Score Matrices

Here is our data set:

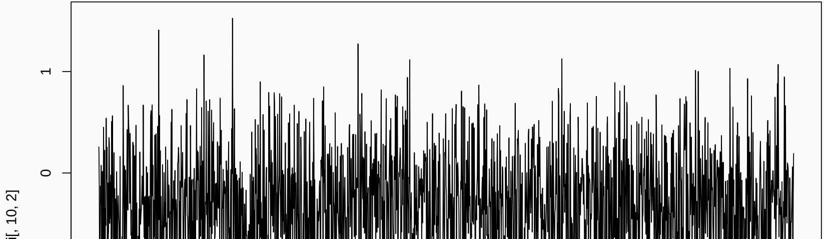
```
set.seed(123)
I=200
J=8
D=2
lambda=matrix(nrow=J,ncol=D)
lambda[,1]=seq(1,0,length=J)
lambda[,2]=seq(0,1,length=J)
Sigma=crossprod(t(lambda))+diag(rep(1^2,J))
y=rmvnorm(I,rep(0,J),Sigma)
```

Your turn

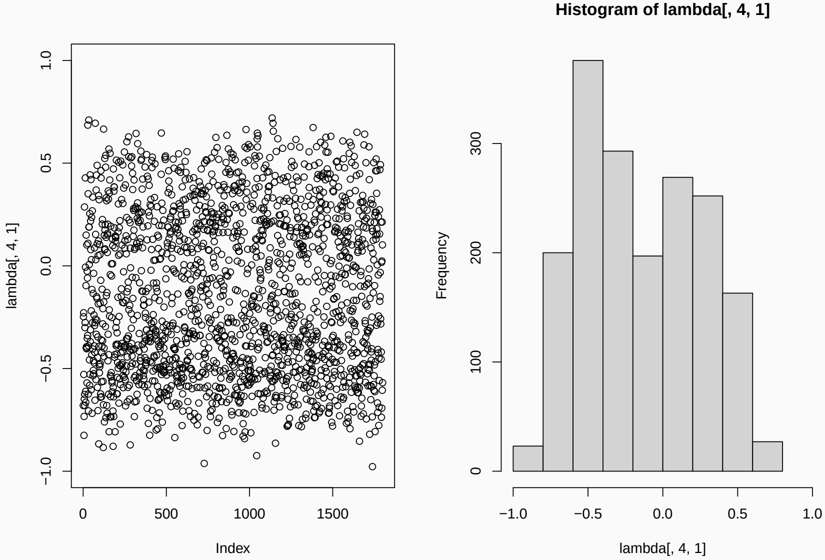
- Program up a Bayes sampler in JAGS or stan (I will do JAGS) for recovering lambda
- Use the conditional formulation
    - write out the model
    - implement it in JAGS or stan
    - run it and see if you can document issues
- When I program up all Gibbs steps, I get a lot of autocorrelation.

My turn

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1600
##    Unobserved stochastic nodes: 432
##    Total graph size: 7056
##
## Initializing model
```



Lambda [4,1]



lambda[, 4, 1]
Index
1.0
0.5
0.0
−0.5
−1.0
0
500
1000
1500
Histogram of lambda[, 4, 1]
Frequency
300
200
100
0
−1.0
−0.5
0.0
0.5
1.0
lambda[, 4, 1]

Rotations!

- each iteration is corresponding to a different rotation.
- what to do?
- old way, fix loading to lower triangle

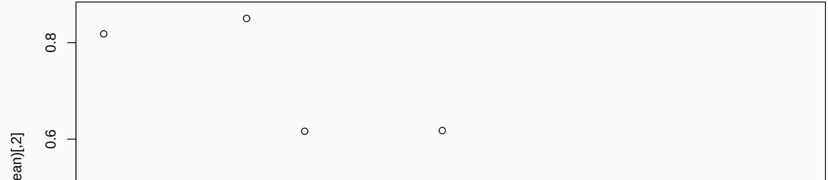
Lower Triangle for 3 Factors

$+$	0	0
*	$+$	0
*	*	$+$
*	*	*
*	*	*
*	*	*
*	*	*
*	*	*

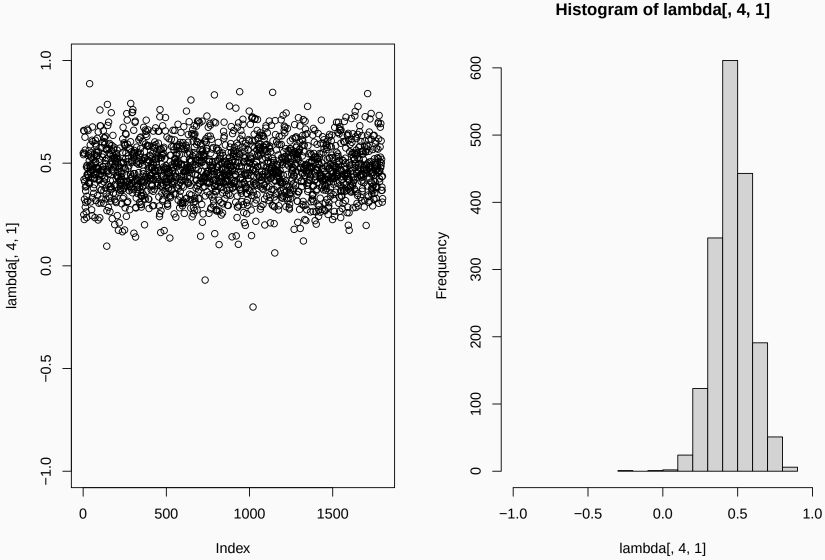
Your Turn

Adapt your code for lower triangle.

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1600
##    Unobserved stochastic nodes: 432
##    Total graph size: 7103
##
## Initializing model
```



Lower Triangle, Lambda [4,1]

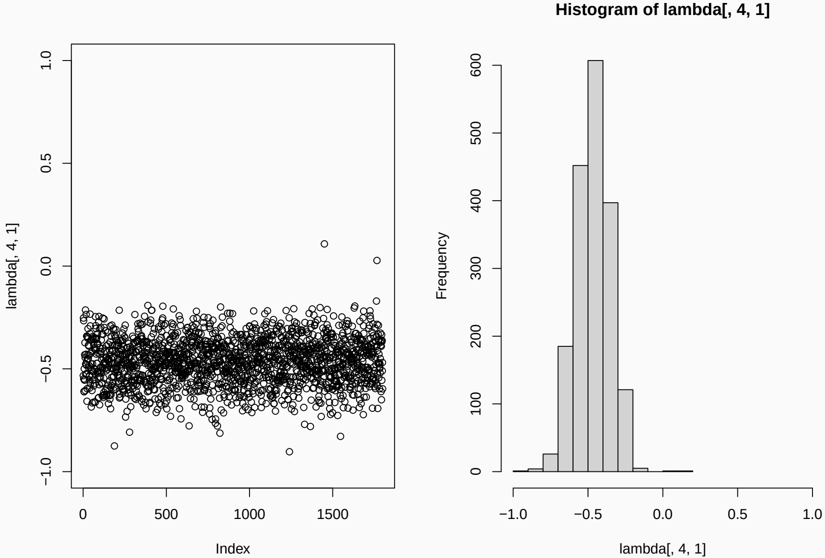


Post-Sampling Rotations

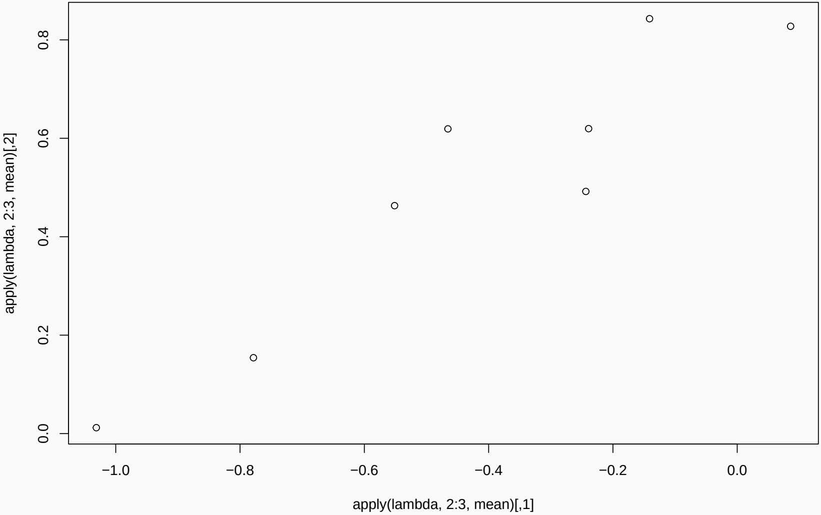
- Very new approach
- Align each iteration to a common rotation after the fact.
- Papastamoulis, P., & Ntzoufras, I. (2022). On the identifiability of Bayesian factor analytic models. Statistics and Computing, 32(2), 23. doi:10.1007/s11222-022-10084-4
- Poworoznek, E., Ferrari, F., & Dunson, D. (2021, July 29). Efficiently resolving rotational ambiguity in Bayesian matrix sampling with matching. Retrieved November 21, 2023, from http://arxiv.org/abs/2107.13783

```r
phi=outCM$BUGSoutput$sims.list$phi
lambda=outCM$BUGSoutput$sims.list$lambda
M=dim(lambda)[1]
lambdaList=lapply(1:M,function(x) lambda[x,,])
etaList=lapply(1:M,function(x) phi[x,,])
aligned=jointRot(lambda = lambdaList, eta = etaList)
lambda=aperm(abind(aligned$lambda, along=3),c(3,1,2))
phi=aperm(abind(aligned$eta, along=3),c(3,1,2))
```

Joint Rotation, Lambda [4,1]



All Lambdas



0.0
0.2
0.4
0.6
0.8
apply(lambda, 2:3, mean)[,2]
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
apply(lambda, 2:3, mean)[,1]

Make Positive

```
makePositive=function(lambda){
  dimMean=apply(lambda,3,mean)
  D=length(dimMean)
  for (i in 1:D) lambda[,,i]=sign(dimMean[i])*lambda[,,i]
  return(lambda)}
```

Me? I use `jointRot` in `infinitefactor`

Hierarchical Factor Models

Illusions Data:

- 138 people
- 10 tasks
    - 5 Illusions
    - 2 versions each
- 15 trials per task

Q? How many factors mediate visual illusions.

Model 1, Exploratory Factor Model

$$
\begin{aligned}
Y_{ijk} | \theta_{ij} &\sim \mathrm{N}(\theta_{ij}, \tau_j^2) \\
\boldsymbol{\theta}_i &\sim \mathrm{N}_J(\boldsymbol{\mu},\ \boldsymbol{\lambda}\boldsymbol{\lambda}' + D(\boldsymbol{\delta}^2))
\end{aligned}
$$

Model 2, Confirmatory Factor Model

$$\boldsymbol{\lambda} = \begin{pmatrix} \lambda_{11} & 0 & 0 & 0 & 0 & \lambda_{16} & \lambda_{17} \\ \lambda_{21} & 0 & 0 & 0 & 0 & \lambda_{26} & \lambda_{27} \\ 0 & \lambda_{32} & 0 & 0 & 0 & \lambda_{36} & \lambda_{37} \\ 0 & \lambda_{42} & 0 & 0 & 0 & \lambda_{46} & \lambda_{47} \\ 0 & 0 & \lambda_{53} & 0 & 0 & \lambda_{56} & \lambda_{57} \\ 0 & 0 & \lambda_{63} & 0 & 0 & \lambda_{66} & \lambda_{67} \\ 0 & 0 & 0 & \lambda_{74} & 0 & \lambda_{76} & \lambda_{77} \\ 0 & 0 & 0 & \lambda_{84} & 0 & \lambda_{86} & \lambda_{87} \\ 0 & 0 & 0 & 0 & \lambda_{9,5} & \lambda_{96} & \lambda_{97} \\ 0 & 0 & 0 & 0 & \lambda_{10,5} & \lambda_{10,6} & \lambda_{10,7} \end{pmatrix}$$