Generalized Network Psychometrics

Combining Network and Latent Variable Models

Sacha Epskamp

SEM 2 Symposium 2017

Psychometrika
pp 1–24

Generalized Network Psychometrics: Combining Network and Latent Variable Models

Authors | Authors and affiliations

Sacha Epskamp, Mijke Rhemtulla, Denny Borsboom

Article

First Online: 13 March 2017

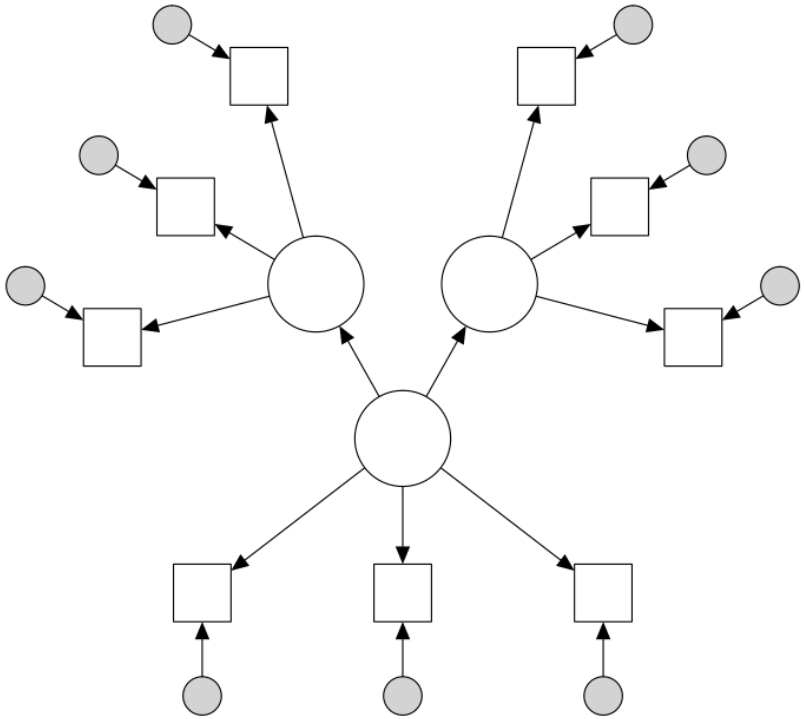
DOI: 10.1007/s11336-017-9557-x

Cite this article as:
Epskamp, S., Rhemtulla, M. & Borsboom, D. Psychometrika (2017). doi:10.1007/s11336-017-9557-x

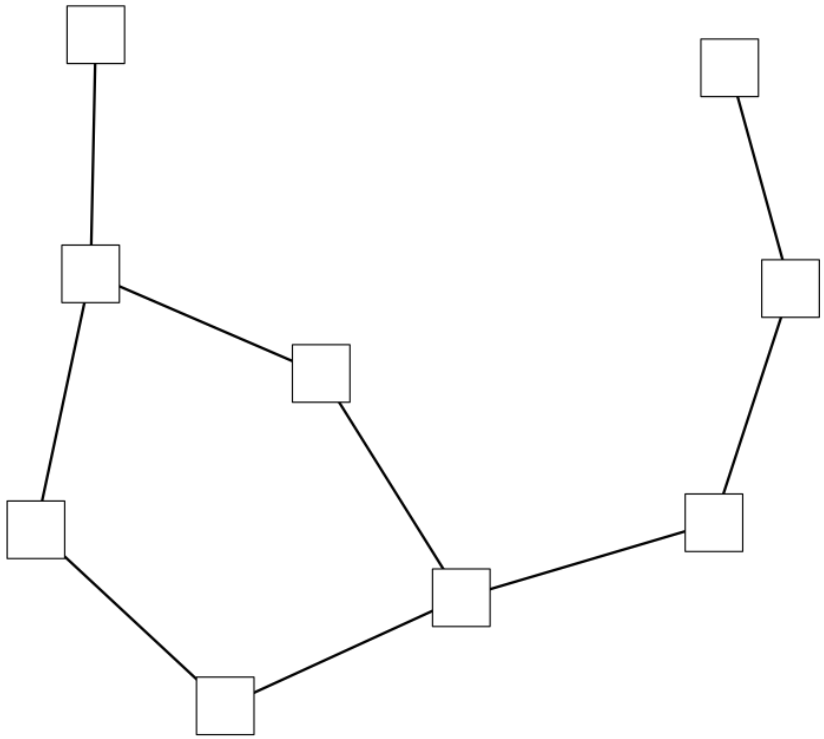
3 Citations | 10 Shares | 239 Downloads

- Published in Psychometrika
  - doi:10.1007/s11336-017-9557-x
- http://rdcu.be/p3Vj (pre-print also on my website)

Structural Equation Modeling



Network Modeling

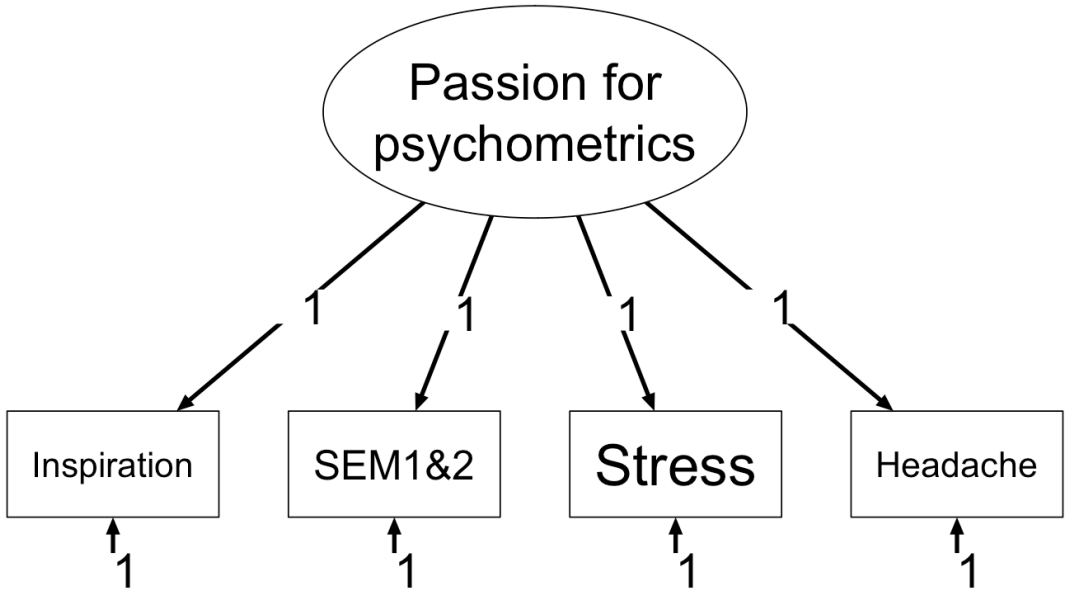


- Both models imply a variance-covariance matrix $\hat{\boldsymbol{\Sigma}}$, aimed to closely resemble the sample variance-covariance matrix $\boldsymbol{S}$ with positive degrees of freedom.

Structural Equation Modeling

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Lambda}(\boldsymbol{I} - \boldsymbol{B})^{-1}\boldsymbol{\Psi}(\boldsymbol{I} - \boldsymbol{B})^{-1\top}\boldsymbol{\Lambda}^{\top} + \boldsymbol{\Theta}$$

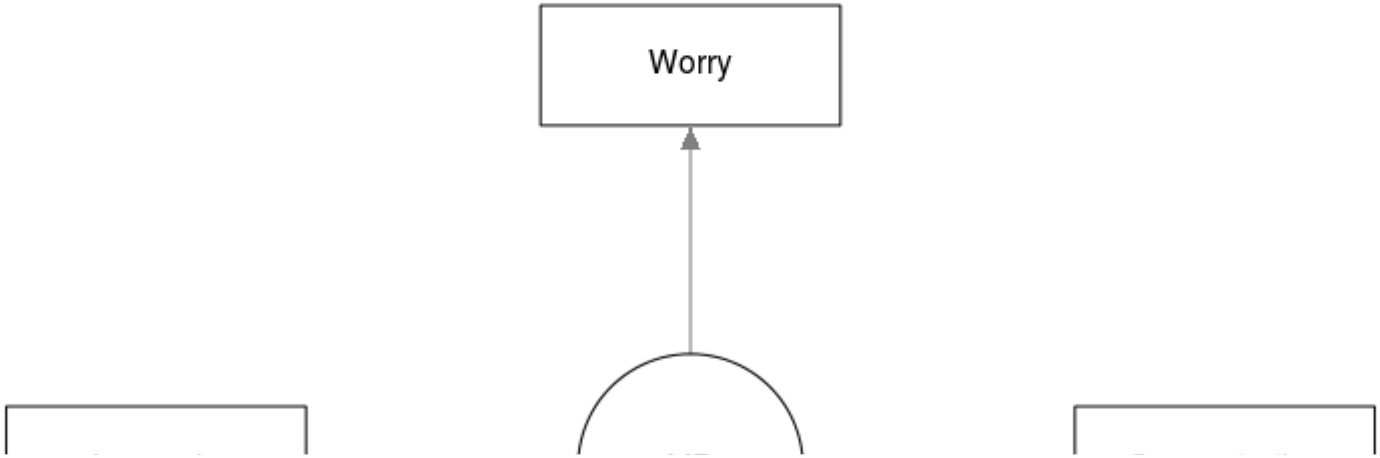
- $\hat{\boldsymbol{\Sigma}}$: model implied variance-covariance matrix
- $\boldsymbol{\Lambda}$: matrix containing factor loadings
- $\boldsymbol{B}$: matrix containing structural relationships between latents
- $\boldsymbol{\Psi}$: variance-covariance matrix of latents or latent residuals
- $\boldsymbol{\Theta}$: matrix containing residual variances and covariances
  - Usually diagonal!

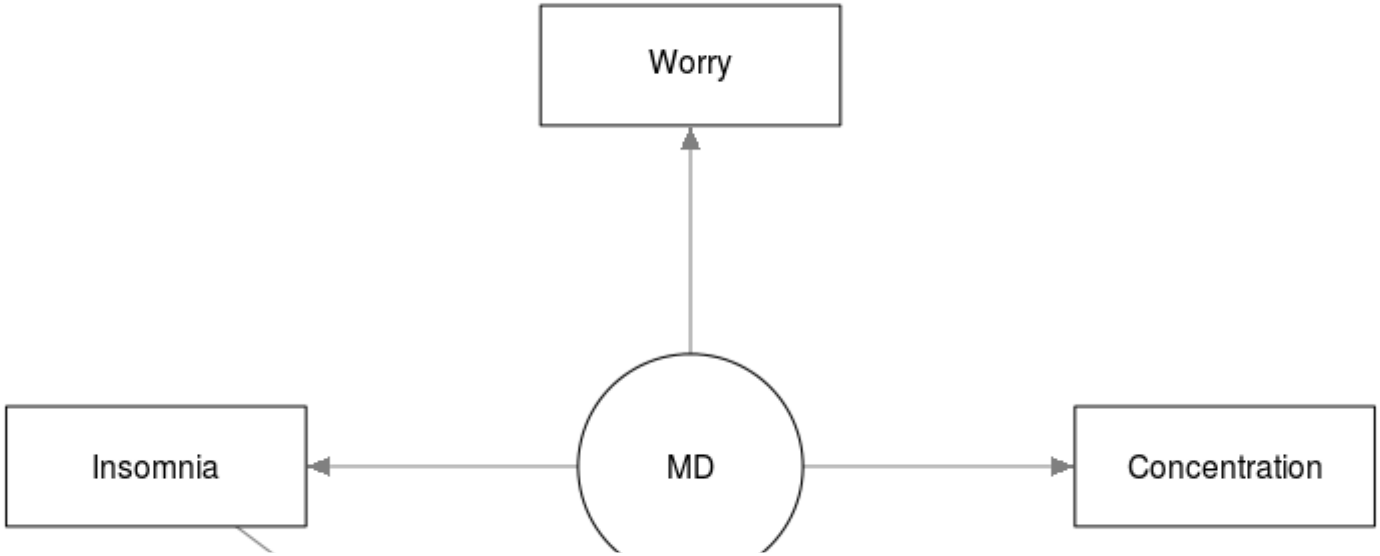

Passion for psychometrics
1
1
1
1
Inspiration
SEM1&2
Stress
Headache
1
1
1
1

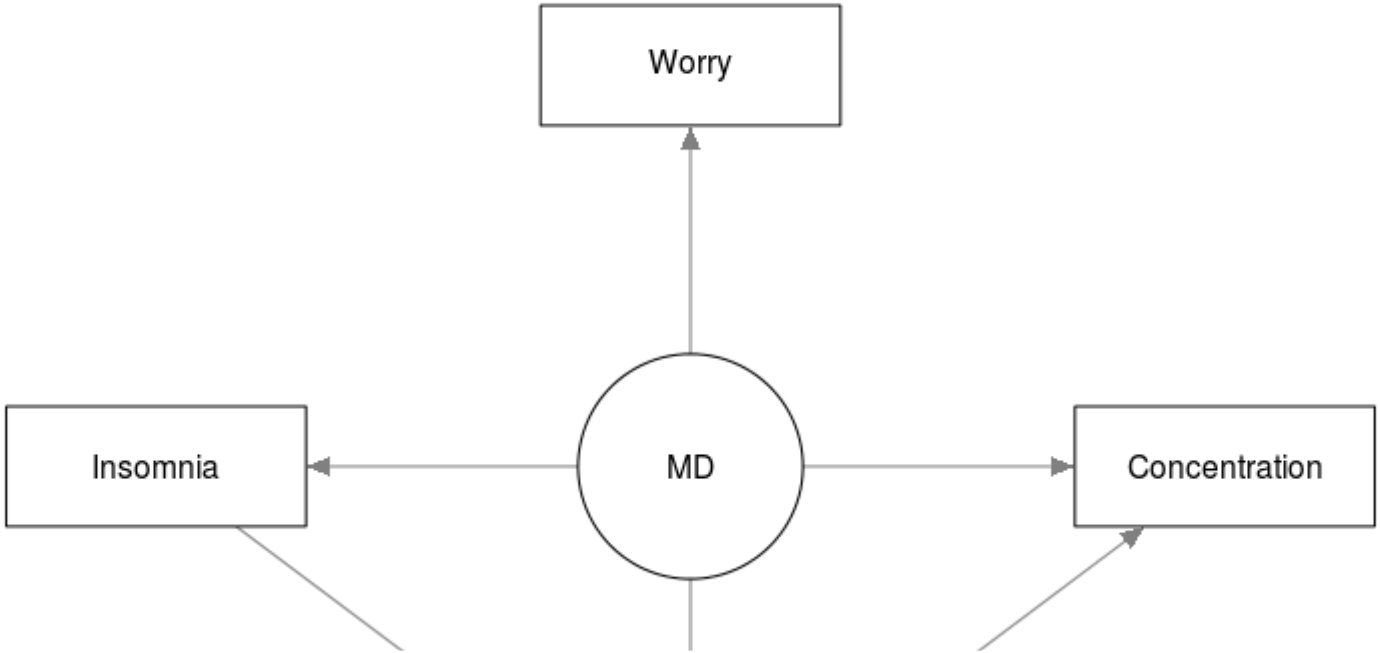
$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Lambda}\boldsymbol{\Psi}\boldsymbol{\Lambda}^\top + \boldsymbol{\Theta}$$

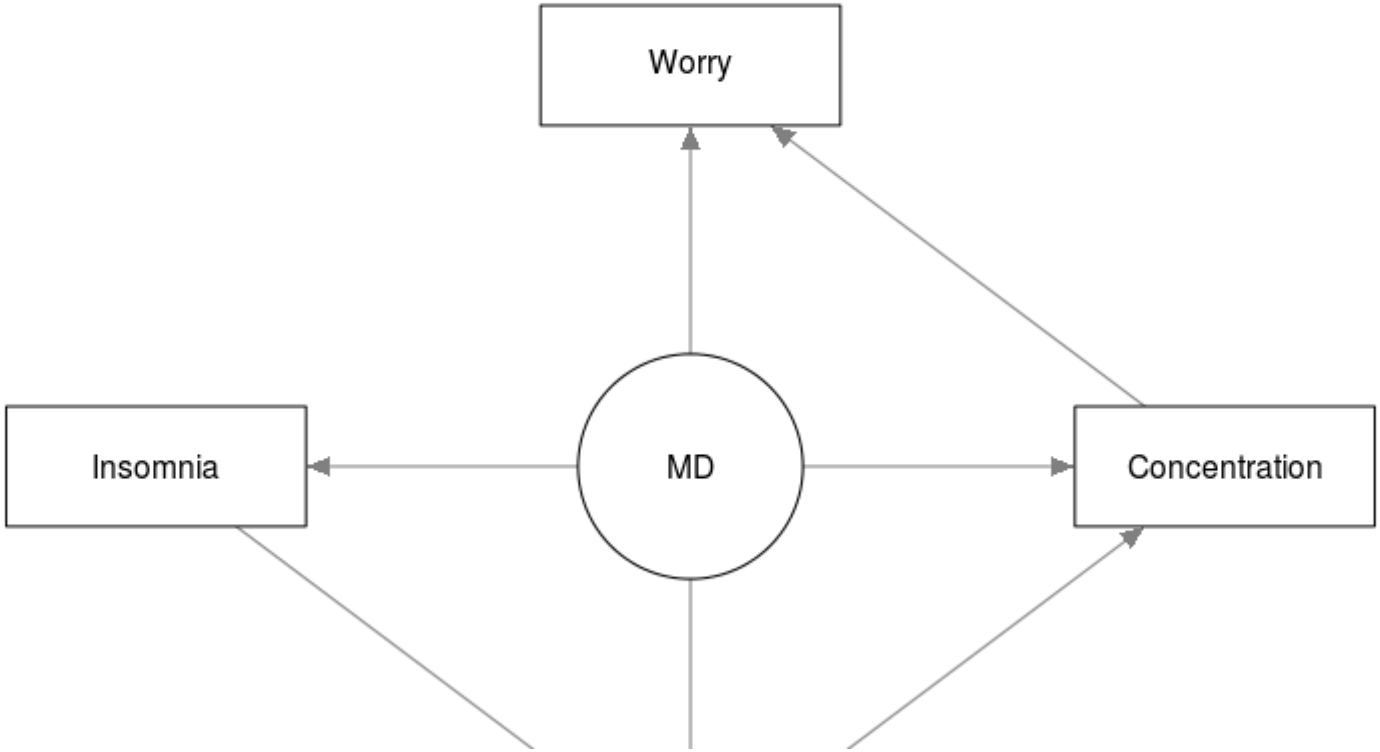
$$\begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

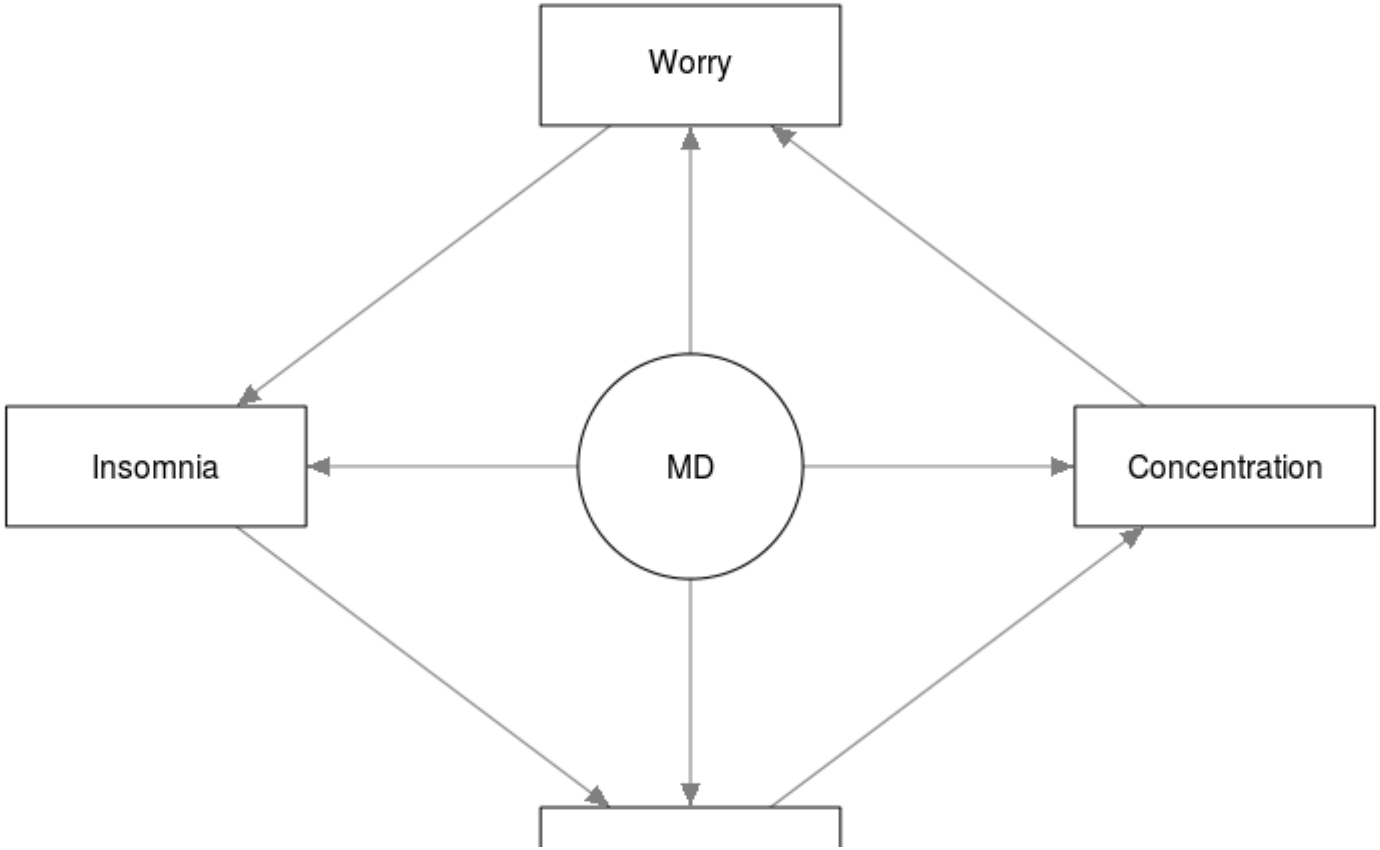
- Degrees of freedom: 2

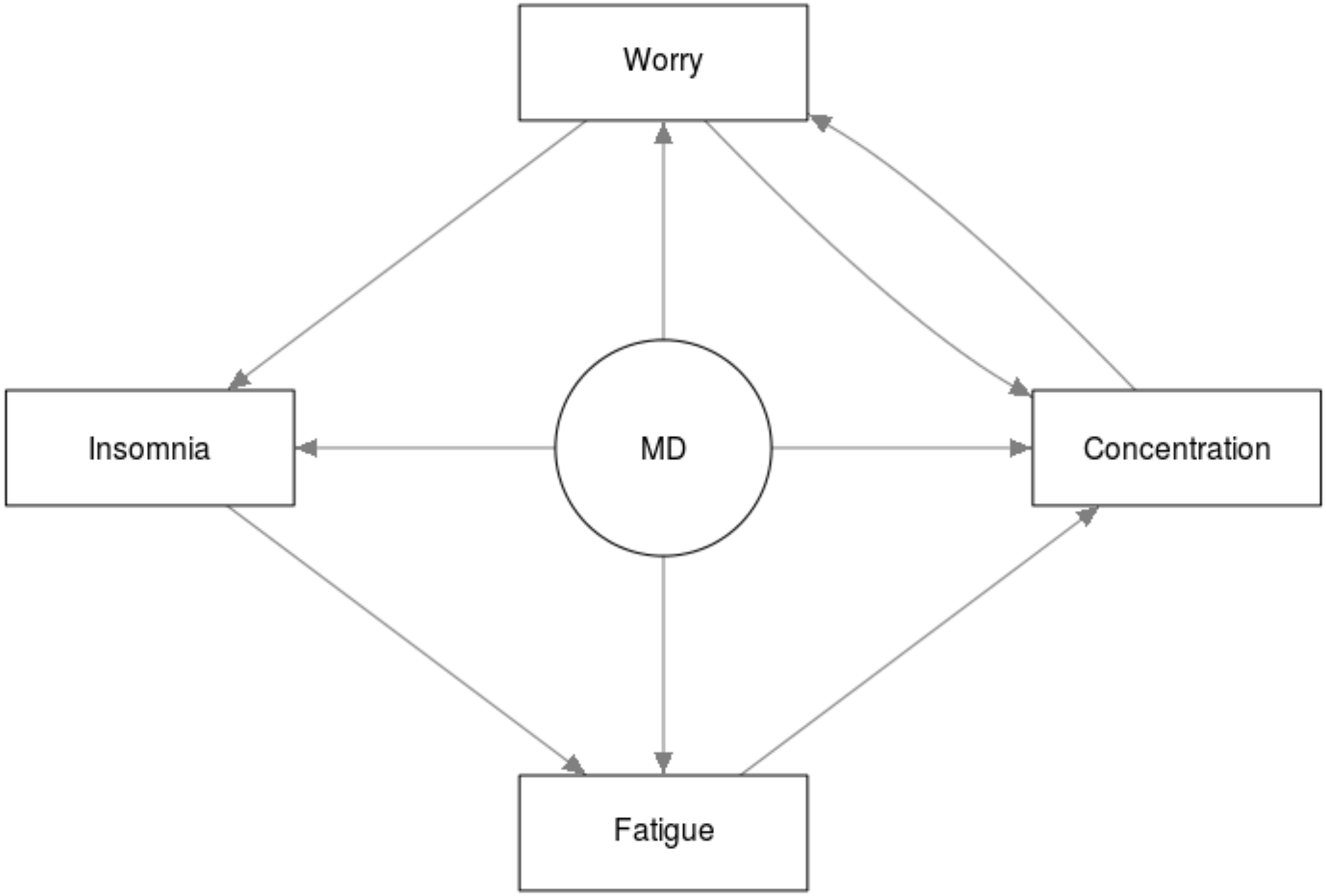

Worry

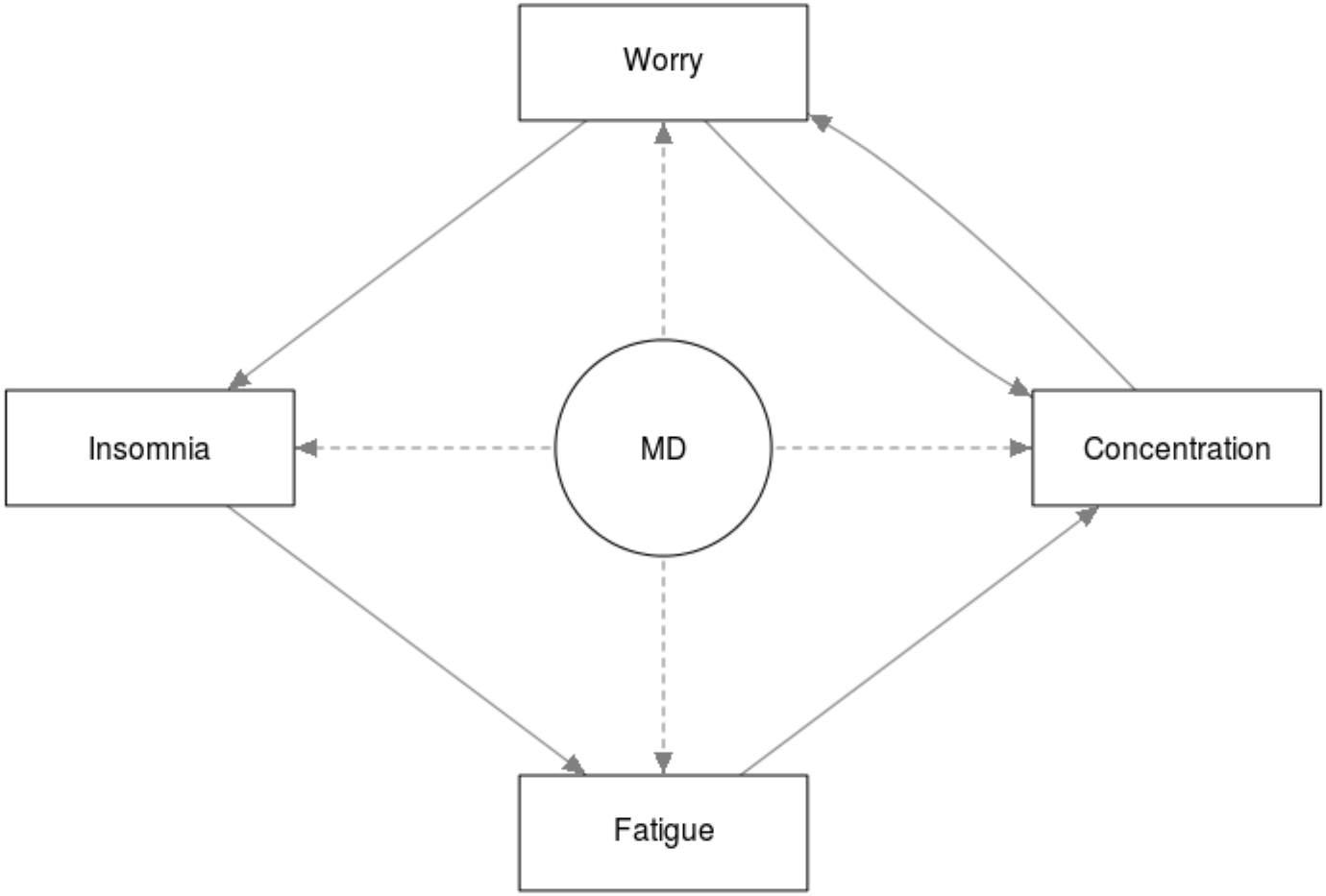

Worry
Insomnia
MD
Concentration

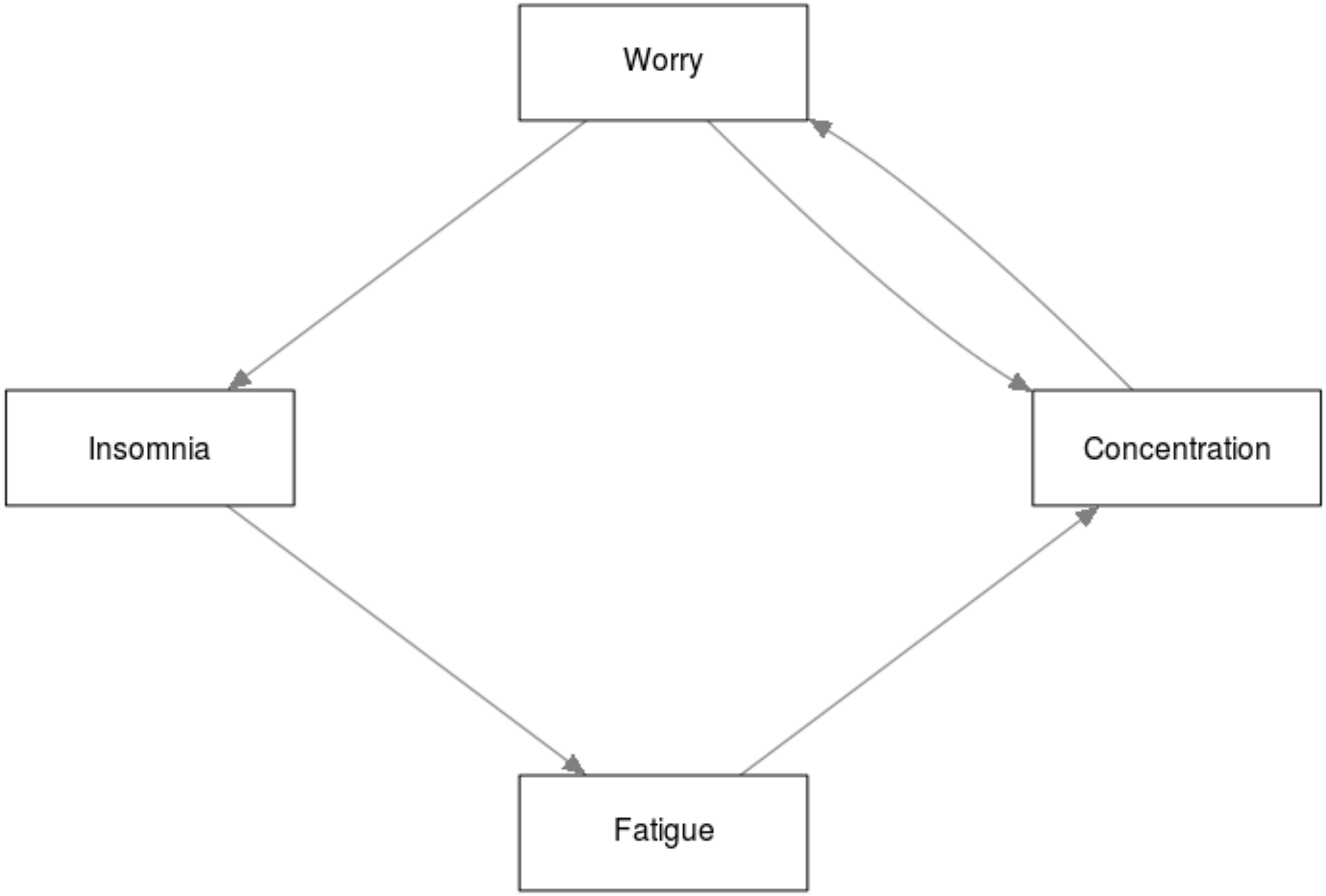

Worry
Insomnia
MD
Concentration

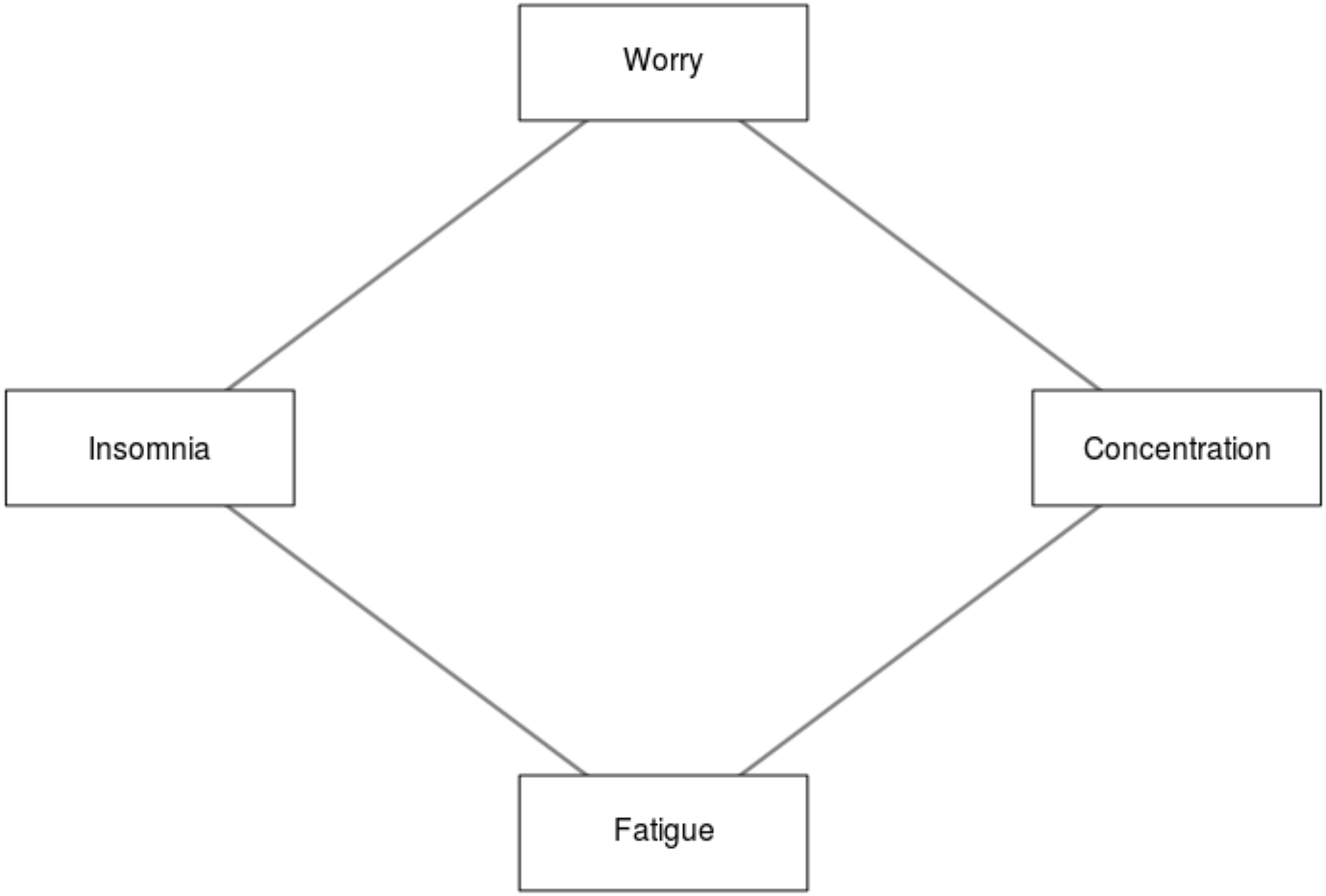

Worry
Insomnia
MD
Concentration

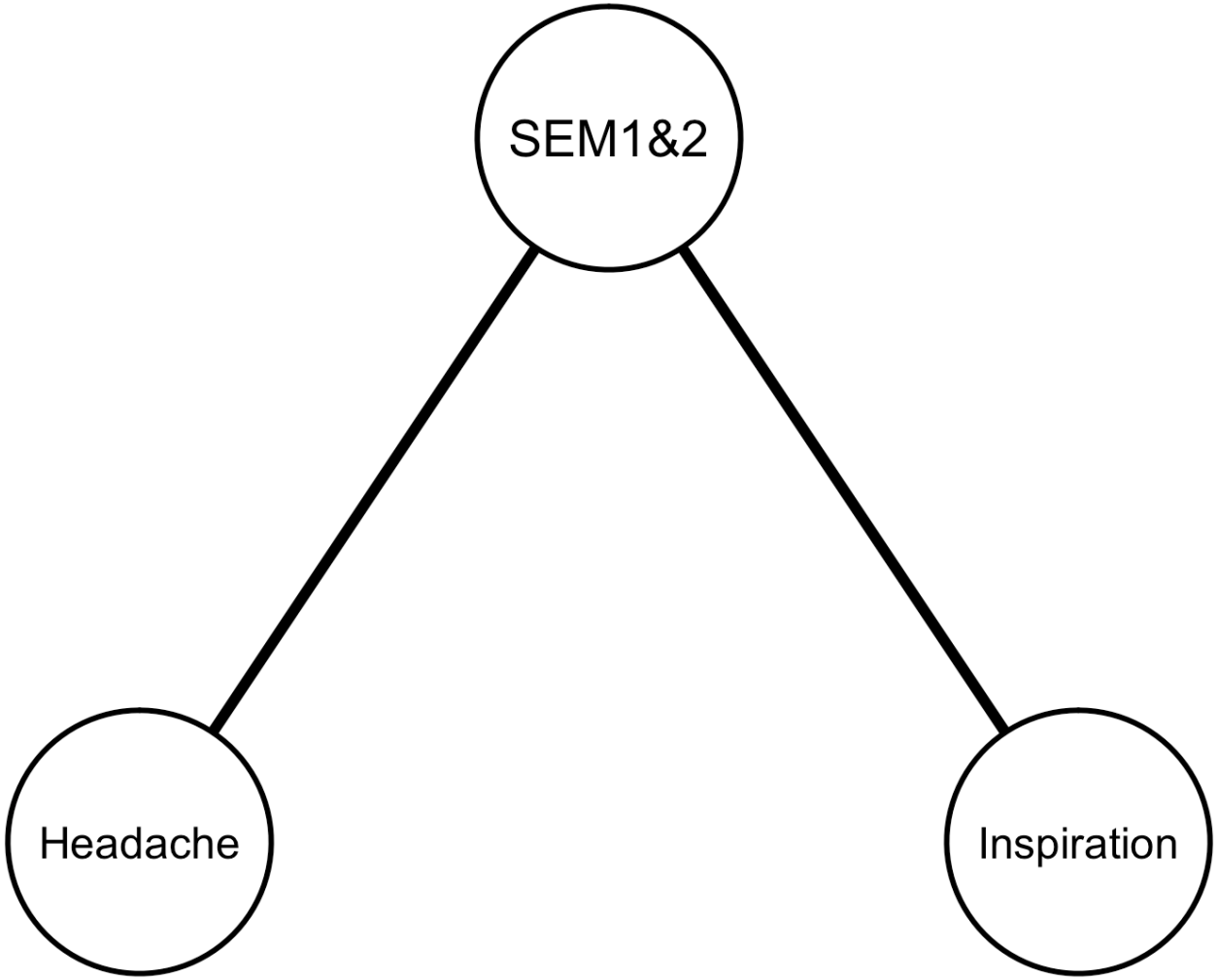

Worry
Insomnia
MD
Concentration


Worry
Insomnia
MD
Concentration
Fatigue

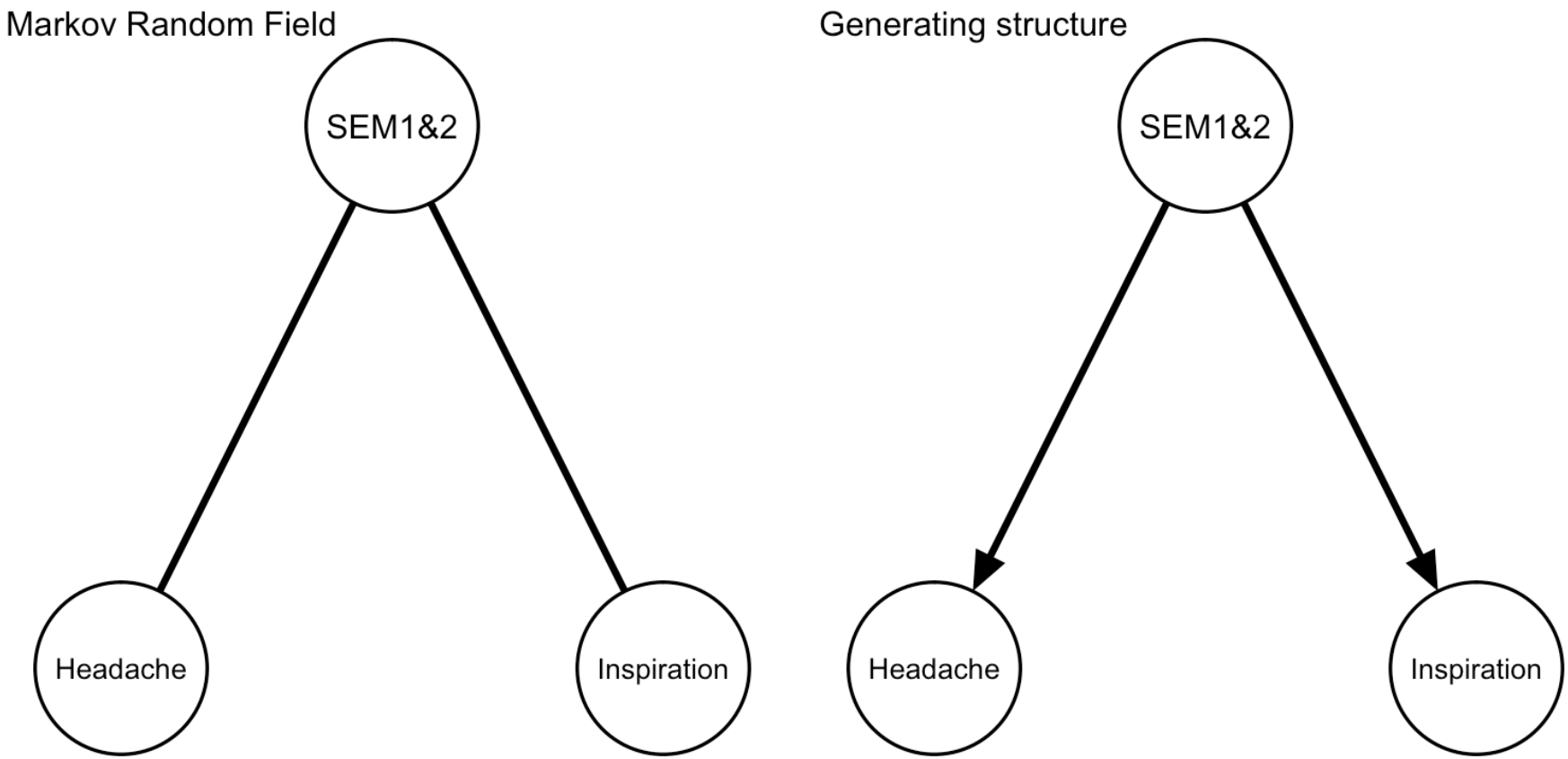

Worry
Insomnia
MD
Concentration
Fatigue


Worry
Insomnia
Concentration
Fatigue

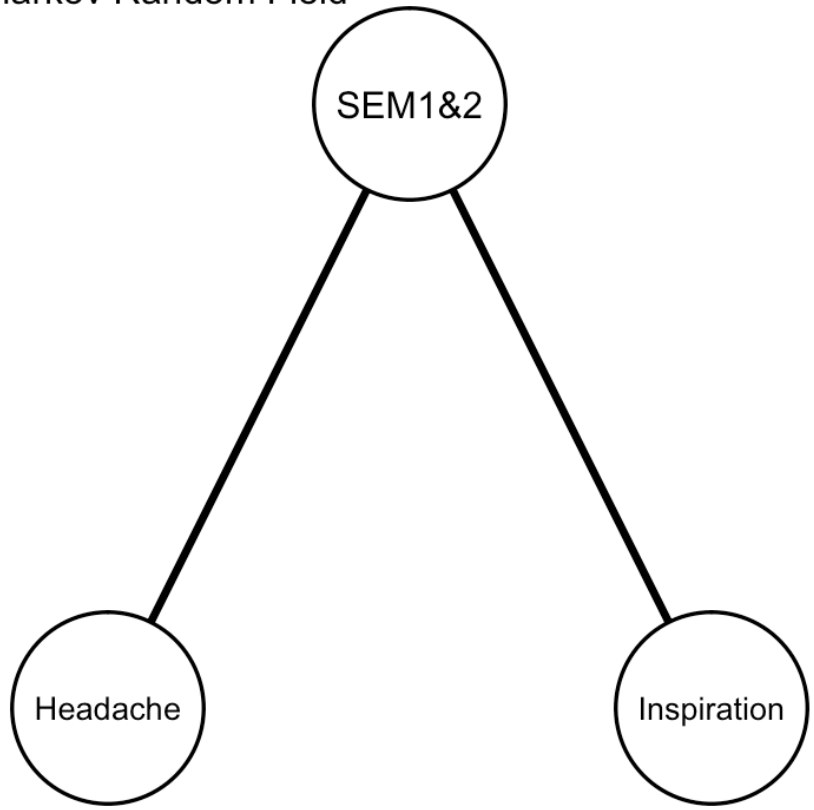

Worry
Insomnia
Concentration
Fatigue



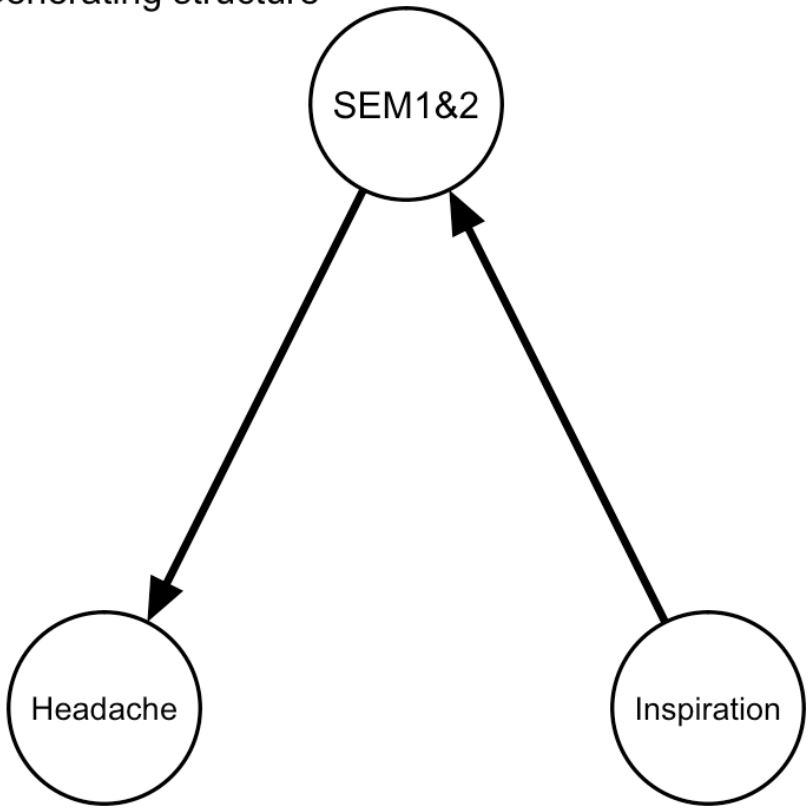
- Inspiration and headache are independent after conditioning on attending SEM1&2


Markov Random Field
SEM1&2
Headache
Inspiration
Generating structure
SEM1&2
Headache
Inspiration

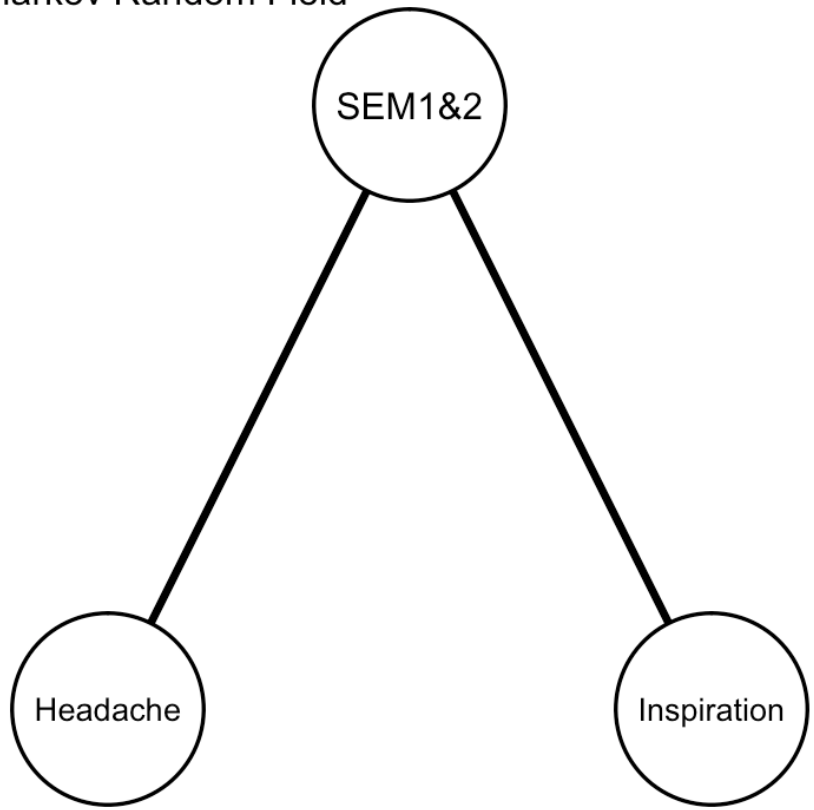
Markov Random Field



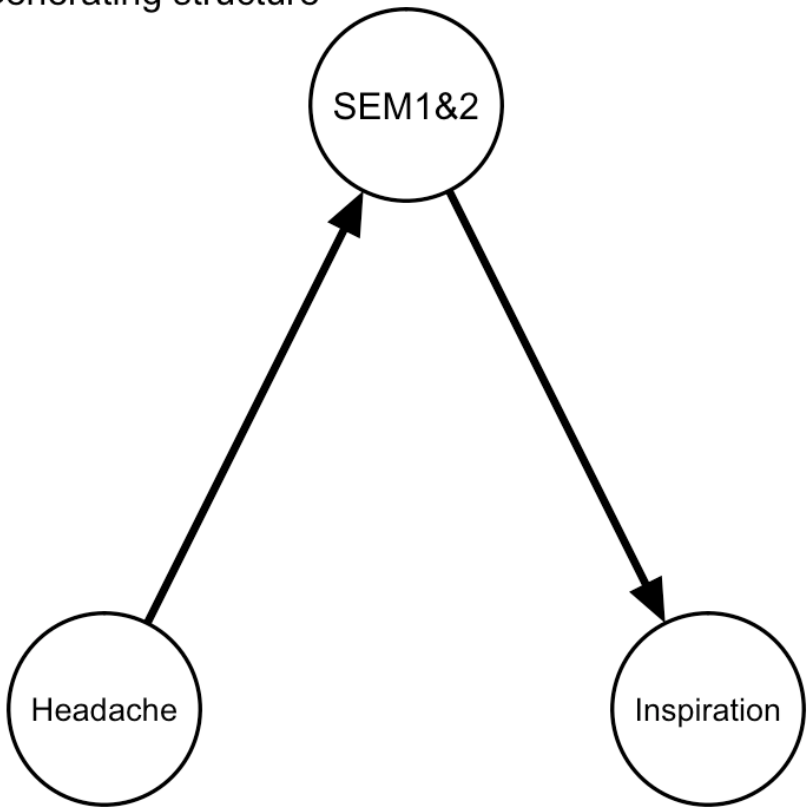
Generating structure



Markov Random Field



Generating structure



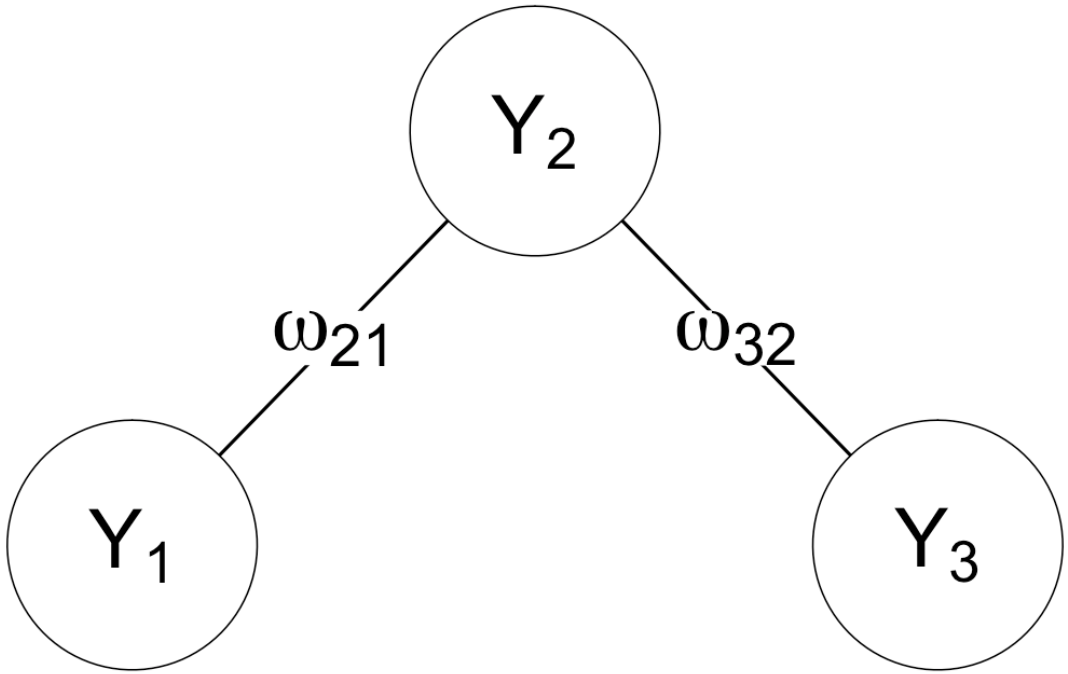
Gaussian Graphical Model

In network analysis, multivariate Gaussian data is modeled with the Gaussian Graphical Model (GGM):

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Delta}(\boldsymbol{I} - \boldsymbol{\Omega})^{-1}\boldsymbol{\Delta}$$

- $\boldsymbol{\Delta}$ is a diagonal scaling matrix
- $\boldsymbol{\Omega}$ is a symmetrical matrix with 0 on the diagonal and *partial correlation coefficients* on offdiagonal elements
  - $\omega_{ij} = \omega_{ji} = \mathrm{Cor}\left(Y_i, Y_j \mid \boldsymbol{Y}^{-(i,j)}\right)$

- Encodes a network; there is no edge between node Y_i and Y_j if $\omega_{ij} = 0$
- A GGM is saturated if all offdiagonal elements in $\mathbf{\Omega}$ are non-zero



$$\mathbf{\Omega} = \begin{bmatrix} 0 & & \\ \omega_{21} & 0 & \\ 0 & \omega_{32} & 0 \end{bmatrix}$$

Sparse configurations of $\mathbf{\Omega}$ can often lead to dense configurations of $\hat{\mathbf{\Sigma}}$

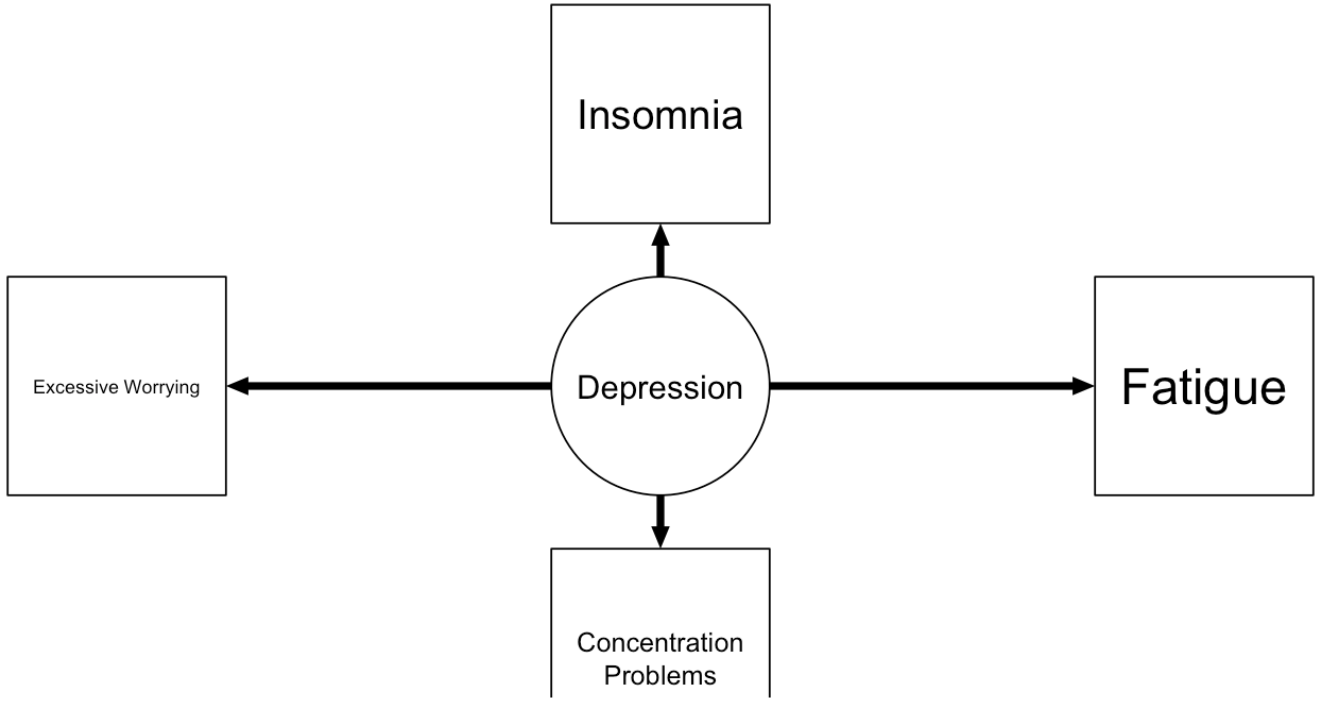
$$\mathbf{\Omega} = \begin{bmatrix} 0 & 0.5 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 0.5 & 0 \end{bmatrix}, \mathbf{\Delta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Results in:

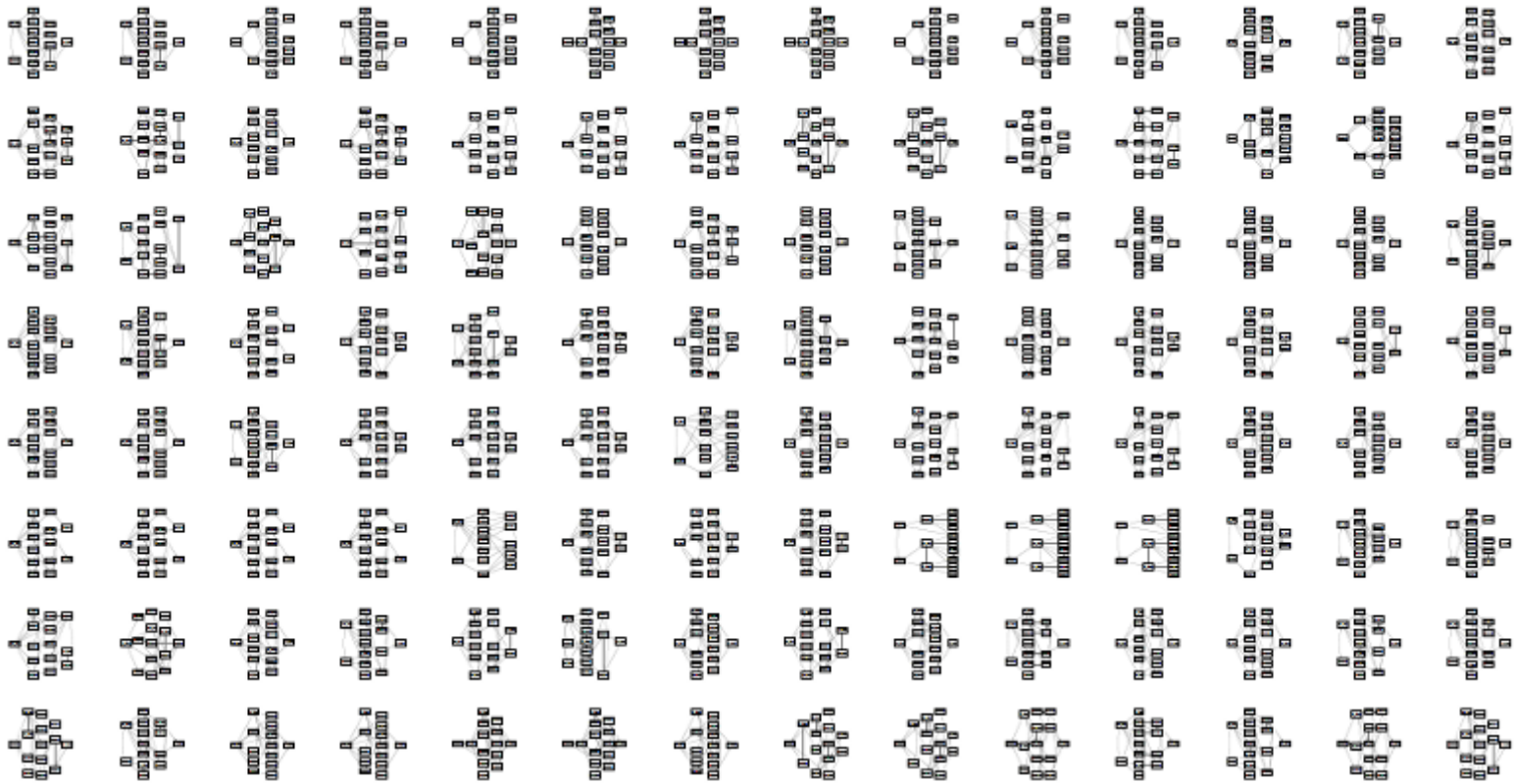
$$\hat{\mathbf{\Sigma}} = \begin{bmatrix} 1.5 & 1 & 0.5 \\ 1 & 2 & 1 \\ 0.5 & 1 & 1.5 \end{bmatrix}$$

- If all nodes are connected, $\hat{\boldsymbol{\Sigma}}$ will be dense
- Degrees of freedom in this example: 1
  - Parameters: 2 edges (ω_{21} and ω_{32}) and 3 scaling parameters (δ_{11}, δ_{22} and δ_{33}).

SEM problem 1: local indepence

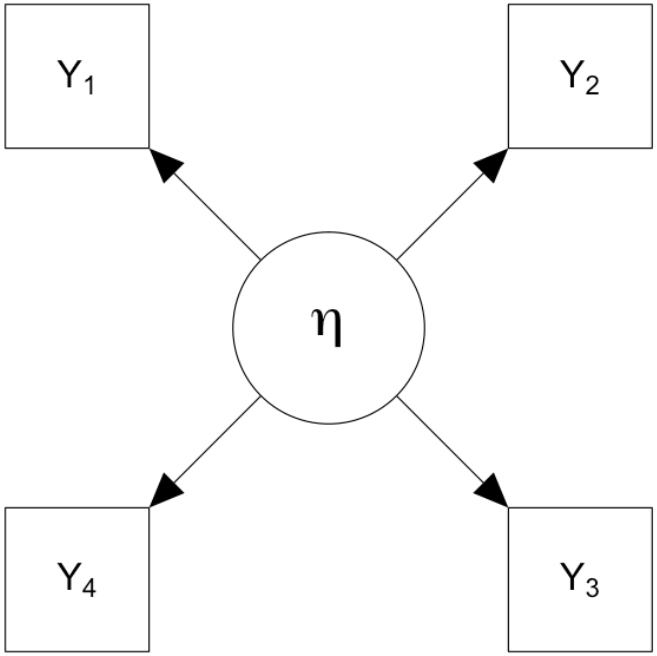


SEM problem 2: exploratory search

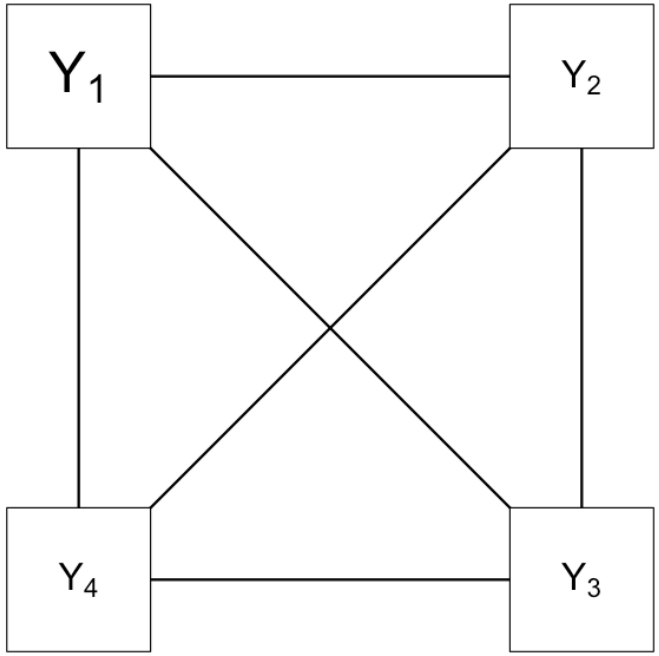


Networks problem 1: latent variables

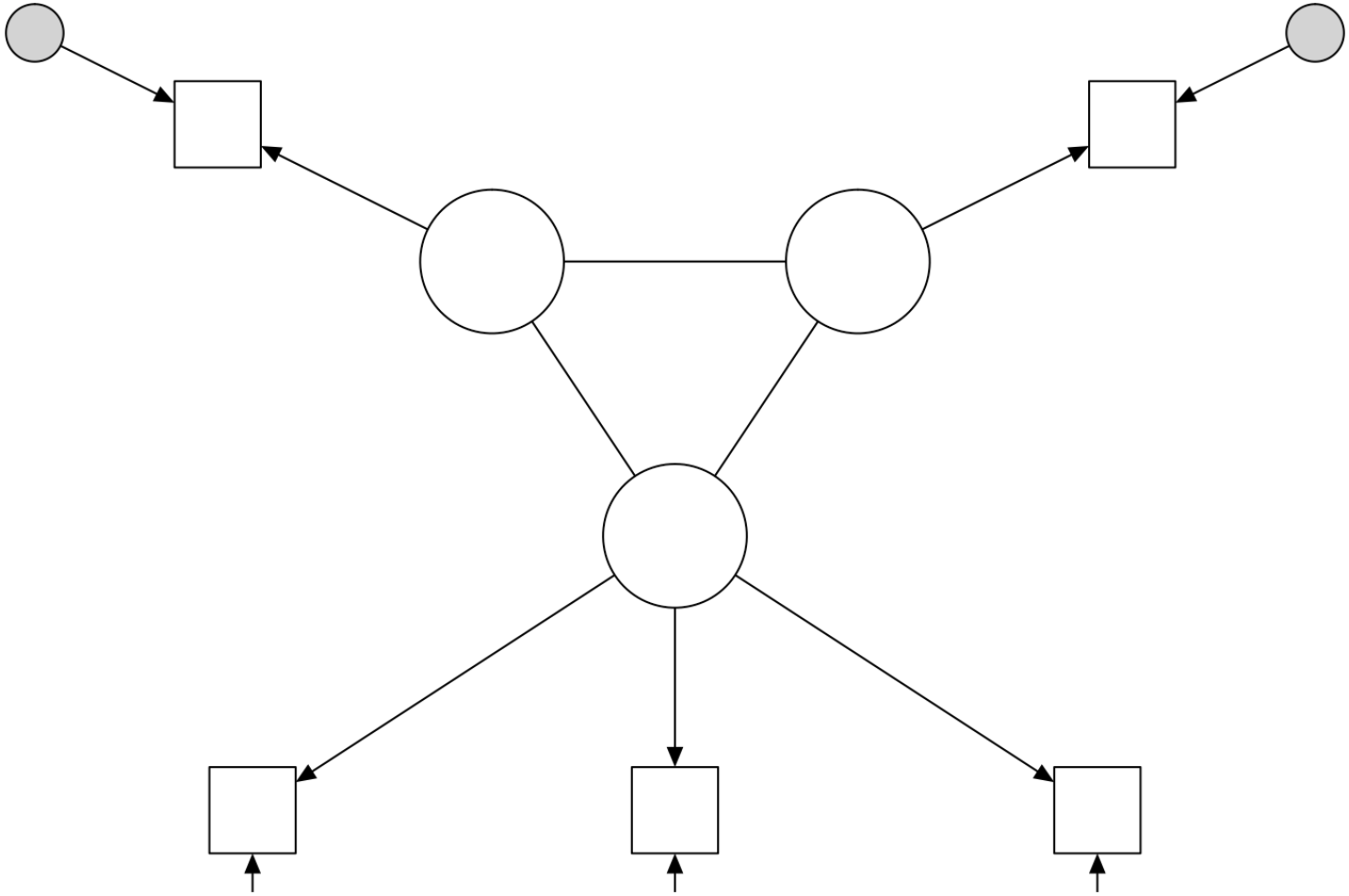
Data generating structure



Markov Random Field



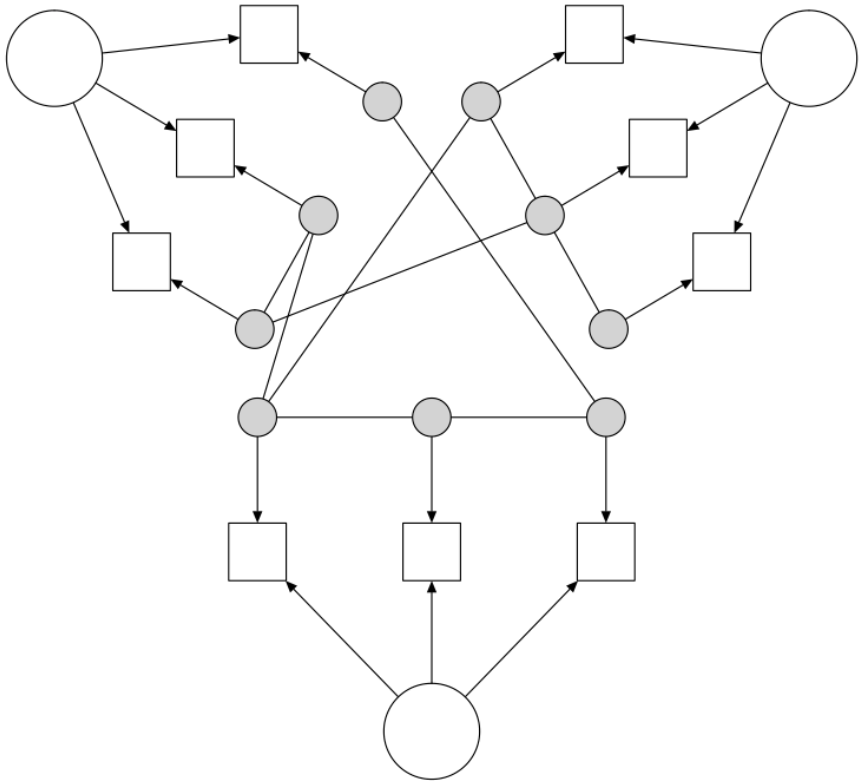
Networks problem 2: Measurement error



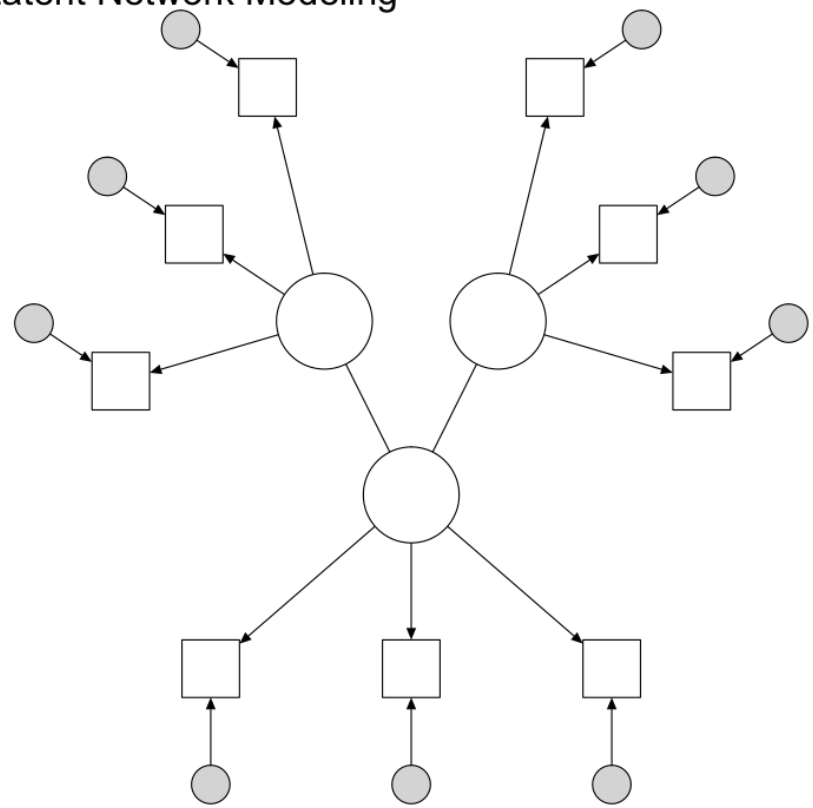
Generalized Network Modeling

Augment Structural Equation Models (SEM) by modeling either the residuals or latent covariances as a Gaussian Graphical model (GGM):

Residual Network Modeling



Latent Network Modeling



Implementing networks in SEM

The variance-covariance matrices in a SEM model can be modeled as a network.

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Lambda}(\boldsymbol{I} - \boldsymbol{B})^{-1}\boldsymbol{\Psi}(\boldsymbol{I} - \boldsymbol{B})^{-1\top}\boldsymbol{\Lambda}^{\top} + \boldsymbol{\Theta}$$

Residual networks:

$$\boldsymbol{\Theta} = \boldsymbol{\Delta}_{\boldsymbol{\Theta}}(\boldsymbol{I} - \boldsymbol{\Omega}_{\boldsymbol{\Theta}})^{-1}\boldsymbol{\Delta}_{\boldsymbol{\Theta}}$$

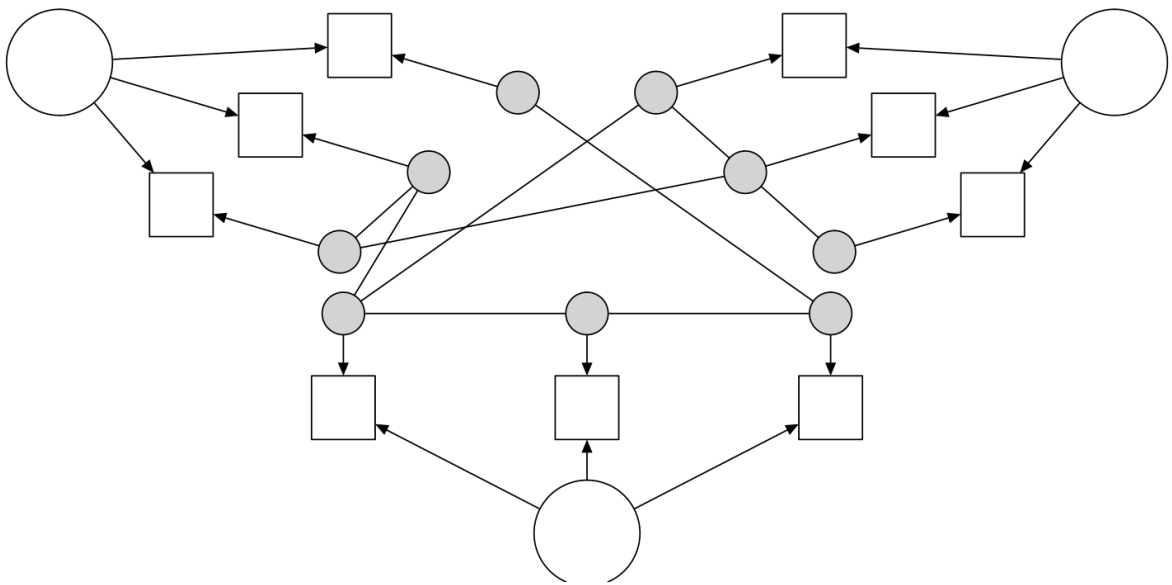
Latent networks:

$$\boldsymbol{\Psi} = \boldsymbol{\Delta}_{\boldsymbol{\Psi}}(\boldsymbol{I} - \boldsymbol{\Omega}_{\boldsymbol{\Psi}})^{-1}\boldsymbol{\Delta}_{\boldsymbol{\Psi}}$$

Residual Network Modeling (RNM)

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Lambda}(\boldsymbol{I} - \boldsymbol{B})^{-1}\boldsymbol{\Psi}(\boldsymbol{I} - \boldsymbol{B})^{-1\top}\boldsymbol{\Lambda}^{\top} + \boldsymbol{\Delta}_{\boldsymbol{\Theta}}(\boldsymbol{I} - \boldsymbol{\Omega}_{\boldsymbol{\Theta}})^{-1}\boldsymbol{\Delta}_{\boldsymbol{\Theta}}$$

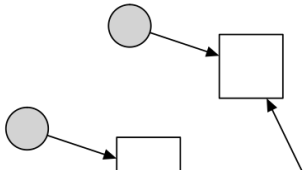
- Network is formed at the residuals of SEM
- Model a network while not assuming no unobserved common causes
- Model a latent variable structure without the assumption of local independence

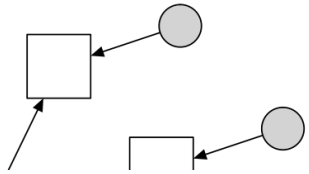


Latent Network Modeling (LNM)

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Lambda}\boldsymbol{\Delta}_{\boldsymbol{\Psi}}(\boldsymbol{I} - \boldsymbol{\Omega}_{\boldsymbol{\Psi}})^{-1}\boldsymbol{\Delta}_{\boldsymbol{\Psi}}\boldsymbol{\Lambda}^{\top} + \boldsymbol{\Theta}$$

- Models conditional independence relations between latent variables as a network
- Model networks between latent variables
- Exploratory search for conditional independence relationships between latents





Exploratory estimation

Two methods for exploratory estimation:

- Stepwise model search
  - Start with empty residual network (RNM) or saturated latent network (LNM) and add/remove edges to improve fit
  - Works for ~8 variables
- Penalized maximum likelihood estimation
  - LASSO regularization
  - Works for ~25 variables
  - Optimize penalized likelihood:

$$\min_{\hat{\boldsymbol{\Sigma}}} \left[\log\det\left(\hat{\boldsymbol{\Sigma}}\right) + \text{Trace}\left(\boldsymbol{S}\hat{\boldsymbol{\Sigma}}^{-1}\right) - \log\det\left(\hat{\boldsymbol{S}}\right) - P + \nu\text{Penalty} \right]$$

Algorithm 1 Stepwise network estimation by χ^2 difference testing.

```
Start with initial setup for Ω
repeat
  for all Unique elements of Ω do
    Remove edge if present or add edge if absent
```


Algorithm 2 Stepwise network estimation by AIC, BIC or EBIC optimization.

```
Start with initial setup for Ω
repeat
  for all Unique elements of Ω do
    Remove edge if present or add edge if absent
    Fit model with changed edge
  end for
  if Any changed edge improved AIC, BIC or EBIC then
    Change edge that improved AIC, BIC or EBIC the most
  end if
until No changed edge improves AIC, BIC or EBIC
```

Algorithm 3 LASSO estimation for exploratory network search.

for all Sequence of tuning parameters $\nu_1, \nu_2, \ldots$ **do**

Estimate LASSO regularized model using given tuning parameter

Count the number of parameters for which the absolute estimate is larger than ϵ

Determine information criterion (AIC, BIC or EBIC) given fit and number of parameters

end for

Select model with best AIC, BIC or EBIC

Refit this model without LASSO in which absolute parameters smaller than ϵ are fixed to zero

Performance

Simulation studies (included in paper) show:

- High specificity (few false edges)
- Sensitivity increases with sample size (correct edges are picked up)
- Slow estimation with increasing number of variables

lvnet package

- Package website: http://github.com/SachaEpskamp/lvnet
- Uses OpenMx for estimating confirmatory latent variable network models
  - `lvnet` function
- Includes exploratory search algorithms

- `lvnetSearch` for step-wise search
- `lvnetLasso` for penalized maximum likelihood

- Also include earlier work on lvglasso `lvglasso` and `EBIClvglasso` functions

Installation:

```
library("devtools")
install_github("sachaepskamp/lvnet")
```

Load dataset:

```
# Load package:
library("lvnet")

# Load dataset:
```

```r
library("lavaan")
data(HolzingerSwineford1939)
Data <- HolzingerSwineford1939[,7:15]
```


Setup lambda:

```r
# Measurement model:
Lambda <- matrix(0, 9, 3)
```

```r
Lambda[1:3,1] <- NA
Lambda[4:6,2] <- NA
Lambda[7:9,3] <- NA

Lambda
```

```
##       [,1] [,2] [,3]
##  [1,]   NA    0    0
##  [2,]   NA    0    0
##  [3,]   NA    0    0
##  [4,]    0   NA    0
##  [5,]    0   NA    0
##  [6,]    0   NA    0
##  [7,]    0    0   NA
##  [8,]    0    0   NA
##  [9,]    0    0   NA
```


Fit model:

```
# Fit CFA model:
CFA <- lvnet(Data, lambda = Lambda)
CFA
```

```
##
## lvnet estimation completed.:
##      - Chi-square (24) = 85.31, p = 0
##      - RMSEA = 0.09 (95% CI: 0.07 - 0.11)
##
##  Use summary(object) to inspect more fitmeasures and parameter estimates (see ?summar
##  Use plot(object) to plot estimated networks and factor structures (see ?plot.lvnet)
##  Use lvnetCompare(object1, object2) to compare lvnet models (see ?lvnetCompare)
```

```
> summary(CFA)
========== lvnet ANALYSIS RESULTS ==========

Input:
        Model:
        Number of manifests:     9
        Number of latents:       3
        Number of parameters:    21
        Number of observations   301

Test for exact fit:
        Chi-square:              85.306
        DF:                      24
        p-value:                 0

Information criteria:
        AIC:                     7517.49
        BIC:                     7595.339
        Adjusted BIC:            7528.739
        Extended BIC:            7835.038

Fit indices:
        CFI:                     0.931
        NFI:                     0.907
        TLI:                     0.896
        RFI:                     0.861
        IFI:                     0.931
        RNI:                     0.931
        RMR:                     0.075
        SRMR:                    0.06

RMSEA:
        RMSEA:                   0.092
        90% CI lower bound:      0.071
        90% CI upper bound:      0.114
        p-value:                 0.001

Parameter estimates:
 matrix row col Estimate
 lambda   1   1    0.900
 lambda   2   1    0.498
 lambda   3   1    0.656
 lambda   4   2    0.990
 lambda   5   2    1.102
 lambda   6   2    0.917
 lambda   7   3    0.619
 lambda   8   3    0.731
 lambda   9   3    0.670
  theta   1   1    0.549
  theta   2   2    1.134
  theta   3   3    0.844
  theta   4   4    0.371
  theta   5   5    0.446
  theta   6   6    0.356
  theta   7   7    0.799
  theta   8   8    0.488
  theta   9   9    0.566
    psi   1   2    0.459
    psi   1   3    0.471
    psi   2   3    0.283
```

CFA fit is comparable to lavaan:

```
HS.model <- ' visual  =~ x1 + x2 + x3
              textual =~ x4 + x5 + x6
              speed   =~ x7 + x8 + x9 '

cfa(HS.model, data=HolzingerSwineford1939)
```

```
## lavaan (0.5-23.1097) converged normally after  35 iterations
##
##   Number of observations                           301
##
##   Estimator                                         ML
##   Minimum Function Test Statistic               85.306
##   Degrees of freedom                                24
##   P-value (Chi-square)                           0.000
```

Latent network:

```r
# Latent network:
Omega_psi <- matrix(c(
  0,NA,NA,
  NA,0,0,
  NA,0,0
),3,3,byrow=TRUE)
Omega_psi
```

```
##      [,1] [,2] [,3]
## [1,]    0   NA   NA
## [2,]   NA    0    0
## [3,]   NA    0    0
```

```r
# Fit model:
LNM <- lvnet(Data, lambda = Lambda, omega_psi=Omega_psi)

# Compare fit:
lvnetCompare(cfa=CFA,lnm=LNM)
```

```
##           Df      AIC      BIC     EBIC    Chisq Chisq diff Df diff
## Saturated  0       NA       NA       NA  0.00000         NA      NA
## cfa       24 7517.490 7595.339 7835.038 85.30552  85.305522      24
## lnm       25 7516.494 7590.637 7818.921 86.31009   1.004565       1
##             Pr(>Chisq)
## Saturated           NA
## cfa       8.502553e-09
## lnm       3.162085e-01
```

Exploratory search for latent network:

```
Res <- lvnetSearch(Data, lambda = Lambda,
            matrix = "omega_psi", verbose = FALSE)
Res$best$matrices$omega_psi
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.0000000 0.4222516 0.4420685
## [2,] 0.4222516 0.0000000 0.0000000
## [3,] 0.4420685 0.0000000 0.0000000
```

Empirical Example

Dataset from the *psych* package on the Big 5 personality traits. This dataset consists of 2800 observations of 25 items designed to measure the 5 central personality traits with 5 items per trait:

```
library("psych")

# Load BFI data:
data(bfi)
bfi <- bfi[,1:25]

# Lavaan model:
Mod <- '
A =~ A1 + A2 + A3 + A4 + A5
E =~ E1 + E2 + E3 + E4 + E5
C =~ C1 + C2 + C3 + C4 + C5
```

```
N =~ N1 + N2 + N3 + N4 + N5
O =~ O1 + O2 + O3 + O4 + O5
'
```


Empirical Example

Dataset from the *psych* package on the Big 5 personality traits. This dataset consists of 2800 observations of 25 items designed to measure the 5 central personality traits with 5 items per trait:

```
library("psych")

# Load BFI data:
data(bfi)
bfi <- bfi[,1:25]

# Lavaan model:
Mod <- '
A =~ A1 + A2 + A3 + A4 + A5
```

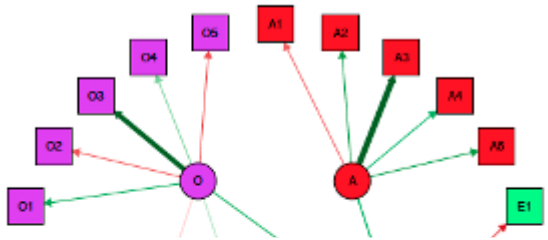
```
E =~ E1 + E2 + E3 + E4 + E5
C =~ C1 + C2 + C3 + C4 + C5
N =~ N1 + N2 + N3 + N4 + N5
O =~ O1 + O2 + O3 + O4 + O5
'
```



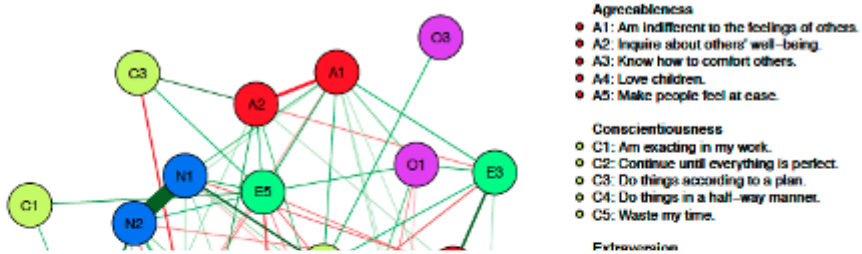
```
# Transform lavaan model to lvnet model matrices:
mod <- lav2lvnet(Mod, bfi)

# Best RNM model:
RNM <- lvnetLasso(bfi, lambda = mod$lambda,
                  lassoMatrix = "omega_theta",
                  nCores = 8,
                  tuning.max = 0.5, nTuning = 100,
                  criterion = "ebic")
RNM_best <- RNM$best
```

	df	chisq	AIC	BIC	EBIC	RMSEA	TLI	CFI
CFA	265	4715.62	230250.58	230606.82	231559.30	0.08	0.75	0.78
RNM	163	688.74	226427.69	227389.55	229961.26	0.03	0.95	0.97
RNM+LNM	168	723.89	226452.85	227385.01	229877.35	0.03	0.95	0.97

Factor structure & latent network



Residual network



Psychometrika
pp 1–24

Generalized Network Psychometrics: Combining Network and Latent Variable Models

Authors | Authors and affiliations

Sacha Epskamp, Mijke Rhemtulla, Denny Borsboom

Article

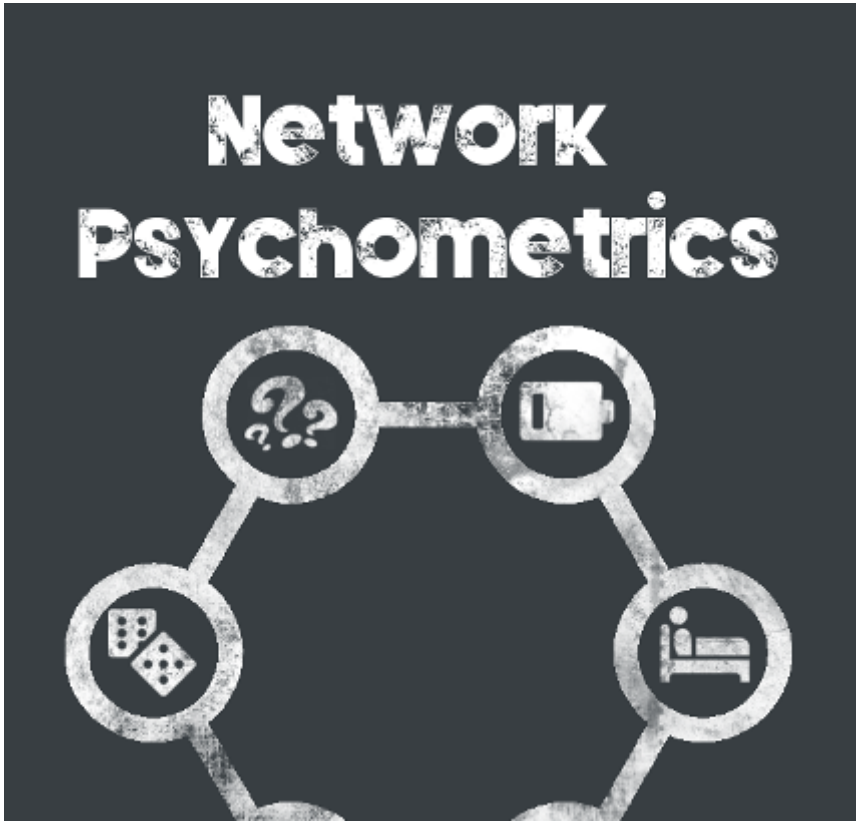
First Online: 13 March 2017

DOI: 10.1007/s11336-017-9557-x

Cite this article as:
Epskamp, S., Rhemtulla, M. & Borsboom, D. Psychometrika (2017). doi:10.1007/s11336-017-9557-x

3 Citations
10 Shares
239 Downloads

- Published in Psychometrika



Latent Variables Combined with Network Analysis

Network Analysis raises the issue of the local independence of manifestations required by Latent Variable approaches. As explained before, considering that manifestations reveal an underlying mechanism (a common cause) is not in contradiction with the hypothesis that manifestations can interact. We consider that Latent Variables and Network Analysis are not mutually exclusive approaches, they are complementary. In fact, some authors who previously advocated Network Analysis rather than Latent Variable models seem to be returning to Latent Variable models. Epskamp et al. (2017, p. 2) today consider that "network modeling and latent variable modeling can complement — rather than exclude — one another... we think the assumption of no underlying latent traits ... may often be too strict." The new approach proposed by Epskamp et al. (2017) seems an interesting approach combining the Latent Variable approach and Network Analysis. It retains the notion of the latent variable and thus views psychological attributes as realities (common causes of manifestations).

- Guyon, H., Falissard, B., & Kop, J. L. (2017). Modelling mental attributes in Psychology–an epistemological discussion: Network Analysis vs Latent Variables. *Frontiers in Psychology, 8*, 798-. Chicago

would also be amenable to this approach. Specifically, we would suggest—as others have (Eaton, 2015; Epskamp et al., in press; Markus, 2010; Stapel, 2015; Young, 2015)—that integrating latent variables into network analysis is the best way forward. We echo the suggestions made by Epskamp et al. (in press), who recently proposed *latent network modeling*, in which latent variables are used to extract the most reliable variance from multiple measures of a symptom, and these latent variables subsequently act as the nodes in a network analysis. This method helps to ensure that the conditional independence networks

- Forbes, M. K., Wright, A. G., Markon, K., & Krueger, R. (2017). Evidence That Psychopathology Symptom Networks ~~do not replicate~~ Have Limited Replicability. *Journal of Abnormal Psychology* (https://osf.io/5dfc4/)

Conclusion

- Generalizing SEM and GGM leads to two frameworks:

  - Residual network model (RNM)
  - Latent network model (LNM)
- RNM and LNM allow for specifying many models
  - CFA with residual network
  - Graphical VAR
  - Network with (some) nodes latent
- Confirmatory fit and exploratory search are possible
- Future directions:
  - Exploratory search algorithms need to be optmized
  - Estimation needs to be *fleshed out*
  - SEM extensions need to be included (e.g., ordinal data, FIML)

Thank you for your attention!