Network analysis for modeling complex systems in SLA research

Contents

Load packages

```
library("bootnet")
library("psychTools")
library("qgraph")
library("NetworkComparisonTest")
```

Load and prepare data

```
Dataset <- read.delim("Dataset1.txt")
data <- Dataset[c(6,7,10:20,26:30,36:40)]
str(data)
```

```
## 'data.frame':    1297 obs. of  23 variables:
##  $ PriorL2ach  : int  29 29 80 100 43 100 95 91 10 85 ...
##  $ CurrentL2ach: int  42 25 81 100 22 100 100 83 27 92 ...
##  $ IdealSelf1  : int  1 2 5 3 2 6 5 5 1 5 ...
##  $ IdealSelf2  : int  2 2 5 4 3 4 3 5 3 1 ...
##  $ IdealSelf3  : int  4 1 4 4 4 6 5 6 1 4 ...
##  $ IdealSelf4  : int  4 2 5 4 2 2 5 3 7 5 ...
##  $ IdealSelf5  : int  1 1 3 4 3 5 5 5 4 5 ...
##  $ OughtSelf1  : int  2 2 4 2 3 3 3 5 6 1 ...
##  $ OughtSelf2  : int  3 2 4 2 3 1 5 3 6 1 ...
##  $ OughtSelf3  : int  4 2 4 4 5 2 5 7 4 1 ...
##  $ OughtSelf4  : int  2 3 5 3 2 4 5 3 5 1 ...
##  $ OughtSelf5  : int  1 2 5 2 2 2 3 6 6 4 ...
##  $ OughtSelf6  : int  2 4 4 2 2 1 3 5 7 5 ...
##  $ IntendEff1  : int  3 3 4 4 4 5 5 5 2 5 ...
```

```
##  $ IntendEff2  : int  1 1 4 3 3 3 3 4 5 4 ...
##  $ IntendEff3  : int  1 1 4 4 5 3 4 4 7 3 ...
##  $ IntendEff4  : int  1 1 4 5 5 5 6 5 5 3 ...
##  $ IntendEff5  : int  1 2 5 5 5 6 6 7 7 1 ...
##  $ VisualStyle1: int  1 4 5 5 5 7 5 6 6 5 ...
##  $ VisualStyle2: int  3 2 5 5 2 7 5 4 2 4 ...
##  $ VisualStyle3: int  1 4 4 4 3 7 5 6 1 3 ...
##  $ VisualStyle4: int  1 4 5 5 3 7 5 7 6 5 ...
##  $ VisualStyle5: int  4 3 4 2 3 4 4 4 2 4 ...
```

```
names(data)
```

```
##  [1] "PriorL2ach"   "CurrentL2ach" "IdealSelf1"   "IdealSelf2"   "IdealSelf3"
##  [6] "IdealSelf4"   "IdealSelf5"   "OughtSelf1"   "OughtSelf2"   "OughtSelf3"
## [11] "OughtSelf4"   "OughtSelf5"   "OughtSelf6"   "IntendEff1"   "IntendEff2"
## [16] "IntendEff3"   "IntendEff4"   "IntendEff5"   "VisualStyle1" "VisualStyle2"
## [21] "VisualStyle3" "VisualStyle4" "VisualStyle5"
```

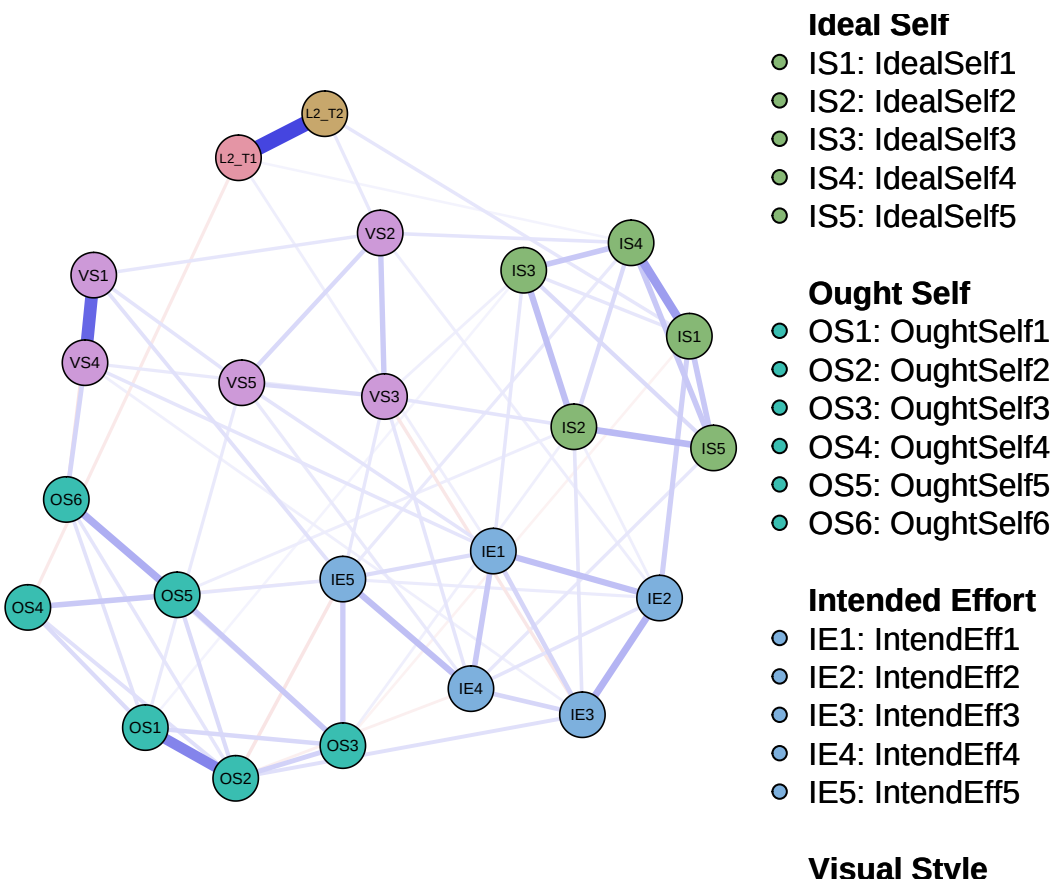
```
groups_data <- list(
  "Prior L2" = 1,
  "CurrentL2" = 2,
  "Ideal Self" = 3:7,
  "Ought Self" = 8:13,
  "Intended Effort" = 14:18,
  "Visual Style" = 19:23
)

nodeNames <- c("L2_T1",
               "L2_T2",
               "IS1",
               "IS2",
               "IS3",
               "IS4",
               "IS5",
               "OS1",
               "OS2",
               "OS3",
               "OS4",
               "OS5",
               "OS6",
               "IE1",
               "IE2",
               "IE3",
               "IE4",
               "IE5",
               "VS1",
               "VS2",
               "VS3",
               "VS4",
               "VS5")
```

Create network model

```
net_Bootnet <- estimateNetwork(data, default = "ggmModSelect", corMethod = "spearman")

Figure1<-
  plot(net_Bootnet,
     palette = "pastel",
     groups = groups_data,
     vsize = 4,
     maximum =1,
     labels = nodeNames,
     nodeNames = names(data),
     legend = TRUE,
     legend.cex = .5)
```




```
pdf(file="Figure1.pdf", paper = "a4r",
    width = 11.69, height = 8.27)

plot(Figure1)

dev.off()
```

```
## pdf
##   2
```

Check stability of network structure

- Compute edge-weight accuracy

```
boot1 <- bootnet(net_Bootnet, nBoots = 5000, nCores = 8)
plot(boot1, labels = FALSE, order = 'sample')

save(boot1, file="boot1.Rdata")

load("boot1.Rdata")
summary(boot1)
```

- Compute differences between edge weights

```
plot(boot1,plot = "difference", onlyNonZero = TRUE, order = "sample")
```

- Compute node centrality indices

```
centralityPlot(net_Bootnet, include = c("Strength","Closeness","Betweenness"))
```

- Compute centrality stability.

```
boot2 <- bootnet(net_Bootnet, nBoots = 5000, type = "case", nCores = 8)
plot(boot2)
save(boot2, file="boot2.Rdata")
load("boot2.Rdata")
```

- Calculate the CS-coefficient

```
corStability(boot2)
```

Network analysis for modeling complex systems in SLA research

Contents

Load packages

```
library("bootnet")
library("psychTools")
library("qgraph")
```

Creating the first model with 7 variables

Load and prepare data

```
Dataset <- read.delim("Dataset2.txt")
DataSeven <- Dataset[c(3,14,17:19,21:22)]
str(DataSeven)
```

```
## 'data.frame':    90 obs. of  7 variables:
##  $ Education     : int  17 13 11 17 12 12 13 11 11 11 ...
##  $ GrammarE_Cor  : num  91.7 95.8 95.8 100 95.8 ...
##  $ VocabRaw      : int  48 58 58 53 55 48 39 48 31 42 ...
##  $ CollocRaw     : int  30 35 31 37 36 21 29 33 22 29 ...
##  $ IQ_Raw        : int  16 11 5 20 16 8 8 10 7 9 ...
##  $ PrintExp_Raw  : int  17 31 38 26 31 15 7 10 6 6 ...
##  $ LgAnalysis_Raw: int  15 13 5 15 14 3 4 5 2 6 ...
```

Create network model

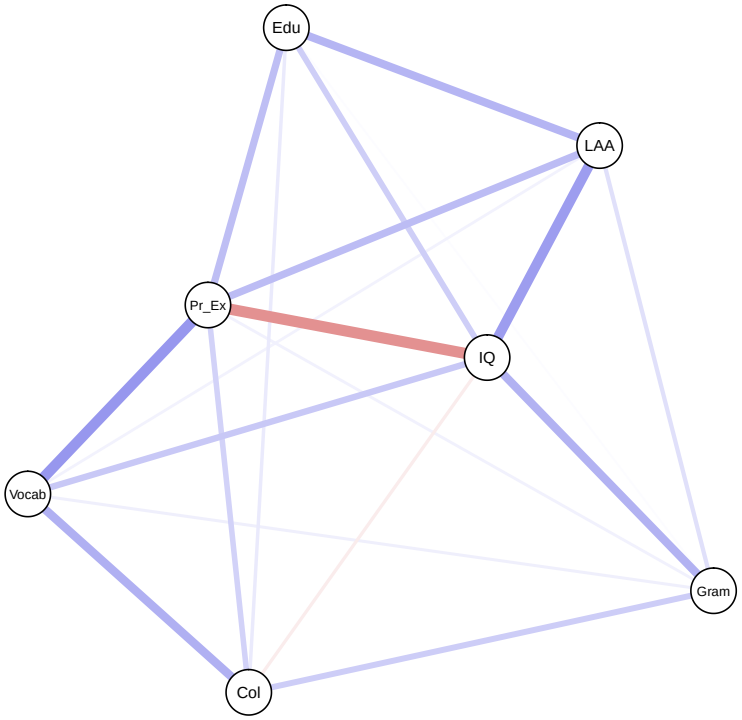
```
nodeNames = c("Education",
              "Grammatical comprehension",
              "Receptive vocabulary size",
              "Collocations knowledge",
              "Nonverbal IQ",
              "Print exposure",
              "Language analytic ability"
)

nodeLabels <- c("Edu",
                "Gram",
                "Vocab",
                "Col",
                "IQ",
                "Pr_Ex",
                "LAA")


net_Bootnet_example2 <- estimateNetwork(DataSeven, default = "EBICglasso", corMethod = "cor")
```

```
## Warning in EBICglassoCore(S = S, n = n, gamma = gamma, penalize.diagonal =
## penalize.diagonal, : A dense regularized network was selected (lambda < 0.1 *
## lambda.max). Recent work indicates a possible drop in specificity. Interpret the
## presence of the smallest edges with care. Setting threshold = TRUE will enforce
## higher specificity, at the cost of sensitivity.
```

```
Figure3 <-
  plot(net_Bootnet_example2,
     maximum = 1,
     theme = "colorblind",
     vsize = 4,
     labels = nodeLabels,
     nodeNames = nodeNames,
     legend = TRUE,
     legend.cex = .5)
```



Edu: Education
Gram: Grammatical comp
Vocab: Receptive vocabu
Col: Collocations knowlec
IQ: Nonverbal IQ
Pr_Ex: Print exposure
LAA: Language analytic a

```
pdf(file="Figure3.pdf", paper = "a4r",
    width = 11.69, height = 8.27)

plot(Figure3)

dev.off()
```

```
## pdf
##   2
```

Check stability of network structure

- Compute edge-weight accuracy

```
boot_example2 <- bootnet(net_Bootnet_example2, nBoots = 5000, nCores = 8)

save(boot_example2, file="boot_example2.Rdata")

plot(boot_example2, labels = FALSE, order = 'sample')
```

- Compute differences between edge weights

```
plot(boot_example2,plot = "difference", onlyNonZero = TRUE, order = "sample")
```

Creating the second model with 8 variables (including age)

Prepare the data

```
DataEight <- Dataset[c(2:3,14,17:19,21:22)]
str(DataEight)
```

```
## 'data.frame':    90 obs. of  8 variables:
##  $ Age           : int  21 38 55 26 55 58 31 58 42 59 ...
##  $ Education     : int  17 13 11 17 12 12 13 11 11 11 ...
##  $ GrammarE_Cor  : num  91.7 95.8 95.8 100 95.8 ...
##  $ VocabRaw      : int  48 58 58 53 55 48 39 48 31 42 ...
##  $ CollocRaw     : int  30 35 31 37 36 21 29 33 22 29 ...
##  $ IQ_Raw        : int  16 11 5 20 16 8 8 10 7 9 ...
##  $ PrintExp_Raw  : int  17 31 38 26 31 15 7 10 6 6 ...
##  $ LgAnalysis_Raw: int  15 13 5 15 14 3 4 5 2 6 ...
```

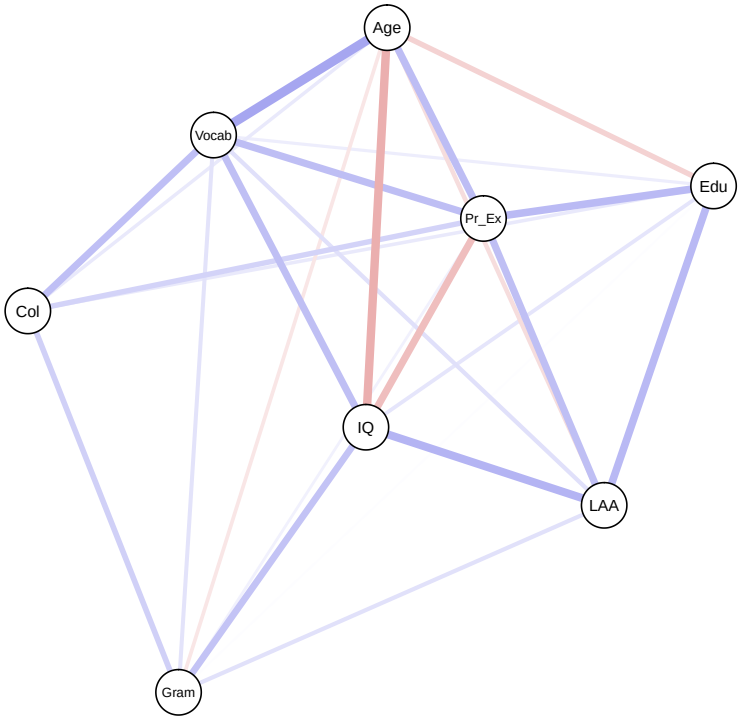
Create network model

```
net_Bootnet_age <- estimateNetwork(DataEight, default = "EBICglasso", corMethod = "cor")

nodeNames2 <- c("Age", nodeNames)

nodeLabels2 <- c("Age", nodeLabels)

Figure4<-
  plot(net_Bootnet_age,
     maximum = 1,
     theme = "colorblind",
     vsize = 4,
     labels = nodeLabels2,
     nodeNames = nodeNames2,
     legend = TRUE,
     legend.cex = .5)
```



Age: Age
Edu: Education
Gram: Grammatical comp
Vocab: Receptive vocabu
Col: Collocations knowlec
IQ: Nonverbal IQ
Pr_Ex: Print exposure
LAA: Language analytic a

```
pdf(file="Figure4.pdf", paper = "a4r",
    width = 11.69, height = 8.27)

plot(Figure4)

dev.off()
```

```
## pdf
##   2
```

Check stability of network structure

- Compute edge-weight accuracy

```
boot_example2_age <- bootnet(net_Bootnet_age, nBoots = 5000, nCores = 8)
save(boot_example2_age, file="boot_example2_age.Rdata")
plot(boot_example2_age, labels = FALSE, order = 'sample')
```

- Test for differences between edge weights

```
plot(boot_example2_age,plot = "difference", onlyNonZero = TRUE, order = "sample")
```

- Compute centrality indices

```
centralityPlot(net_Bootnet_age, include = c("Strength","Closeness","Betweenness"))
```

- Compute centrality stability

```
boot4 <- bootnet(net_Bootnet_age, nBoots = 5000, type = "person", nCores = 2)
plot(boot4)
```

- Calculate the CS-coefficient

```
corStability(boot4)
```

Compare correlations between Print Exposure and the raw and transformed IQ scores

```
attach(Dataset)
cor.test(IQ_Raw, PrintExp_Raw)
```

```
##
##  Pearson's product-moment correlation
##
## data:  IQ_Raw and PrintExp_Raw
## t = -0.36019, df = 88, p-value = 0.7196
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2435245  0.1700741
## sample estimates:
##         cor
## -0.03836841
```

```
cor.test(IQ_Stand, PrintExp_Raw)
```

```
##
##  Pearson's product-moment correlation
##
## data:  IQ_Stand and PrintExp_Raw
## t = 0.77436, df = 88, p-value = 0.4408
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1269870  0.2845115
## sample estimates:
##        cor
## 0.08226755
```