How to evaluate theory-based hypotheses in a lavaan model using the GORICA

Rebecca M. Kuiper

11 november 2025

Contents

This is a tutorial for applying the AIC-type criterion GORICA to a lavaan object. The GORICA is an information criterion that can be used to evaluate theory-driven, informative, order-restricted hypotheses. The GORICA selects the best hypothesis out of a given set. Below, you will find examples for the use of the `goric` function in the `restriktor` R package when you have a lavaan object.

Note: For (more) information (tutorials and example R scripts) is available from https://github.com/rebeccakuiper/Tutorials, including a tutorial regarding interpreting the GORIC(A) output (called 'Guidelines_output_GORIC').

Packages

First, install and call the `lavaan` library and the `restriktor` library (for the `goric` function). If needed, it is possible to view the description of the function with the `?` operator or the `help` command.

```
# To install restriktor in R: if (!require('restriktor'))
# install.packages('restriktor') To install restriktor from github: if
# (!require('devtools')) install.packages('devtools') library(devtools)
# install_github('LeonardV/restriktor')
library(restriktor)

# print docs in the help-tab to view arguments and explanations for the
# function ?goric

# To install lavaan in R: if (!require('lavaan')) install.packages('lavaan')
library(lavaan)
```

Example 1: Confirmatory Factor Analysis

In this example, we will look at the Confirmatory Factor Analysis (CFA) lavaan example (https://www.lavaan.ugent.be/tutorial/cfa.html).

To specify your hypotheses in terms of model parameters, it is often handy (and sometimes even necessary) to give your own labels to estimates by including them in the `lavaan` model.

```
# specify the model, using own labeling
HS.model <- " visual  =~ lambda_v1*x1 + lambda_v2*x2 + lambda_v3*x3
              textual =~ lambda_t1*x4 + lambda_t2*x5 + lambda_t3*x6
              speed   =~ lambda_s1*x7 + lambda_s2*x8 + lambda_s3*x9 "
```

Next, we need to formulate the hypothesis of interest. Let us say that the hypothesis is that all standardized factor loadings are at least (in absolute sense) .6. Note that we compare the factor loadings to a number, which is often the most meaningful when inspecting standardized factor loadings; then, use `standardized = TRUE`. As another note, it is also possible to compare the height of (absolute values of) standardized factor loadings. Since we are interested in one theory-based hypothesis, we compare it to its complement (reflecting all the other possibilities); which is done by default (i.e., by default: `comparison = "complement"`).

```
# Hypotheses
H1_cfa <- "
abs(lambda_v1) > .6; abs(lambda_v2) > .6; abs(lambda_v3) > .6;
abs(lambda_t1) > .6; abs(lambda_t2) > .6; abs(lambda_t3) > .6;
abs(lambda_s1) > .6; abs(lambda_s2) > .6; abs(lambda_s3) > .6;
"
# vs its complement (default)
```

Then, we fit the confirmatory factor analysis (CFA) model using the `cfa` function from `lavaan`:

```
# fit the model fixing the variances of all the latent variables in a CFA model
# to unity (std.lv = TRUE; https://www.lavaan.ugent.be/tutorial/syntax2.html)
fit_cfa <- cfa(HS.model, data = HolzingerSwineford1939, std.lv = TRUE)
```

Now, we can call the `goric` function. Here, we use GORICA (i.e.,`type = "gorica"`) by default, because of entering a lavaan object.

```
# Calculate GORICA values and weights for H1_cfa and its complement.

set.seed(100)  # Needed for reproducibilty & sensitivity check
results_cfa <- goric(fit_cfa, hypotheses = list(H1_cfa = H1_cfa), standardized = TRUE)
```

```
restriktor Message: The covariance matrix of the estimates was obtained via 'vcov()'. This is the biase
```

```
# summary(results_cfa)
results_cfa
```

```
restriktor (0.6-20): generalized order-restricted information criterion approximation:

Results:
        model  loglik  penalty   gorica  loglik.weights  penalty.weights  gorica.weights
1      H1_cfa  16.167    4.478  -23.378           0.011            0.989           0.497
2  complement  20.685    8.984  -23.402           0.989            0.011           0.503

Conclusion:
The order-restricted hypothesis 'H1_cfa' has 0.497 / 0.503 < 1 times more support than its complement.
```

Conclusion: The order-restricted hypothesis 'H1_cfa' has < 1 times more, so, less, support than its

complement.

Example 2: Multiple Group CFA

In this example, we will extend the CFA with a multiple-group structure (cf. https://www.lavaan.ugent.be/tutorial/groups.html). There are two groups: 1. Pasteur and 2. Grant-White. Each will have there own parameter estimates and thus also need there own labeling. In this example, we will evaluate whether the standardized factor loadings in the second group are higher than the corresponding ones in the first group (i.e., in the first group lower than in the second).

```
# specify the model, using own labeling (per group!)
HS.model_mgcfa <- " visual  =~ c(lambda_v1_P, lambda_v1_GW)*x1 + c(lambda_v2_P, lambda_v2_GW)*x2 + c(lam
              textual =~ c(lambda_t1_P, lambda_t1_GW)*x4 + c(lambda_t2_P, lambda_t2_GW)*x5 + c(lambda_t3
              speed   =~ c(lambda_s1_P, lambda_s1_GW)*x7 + c(lambda_s2_P, lambda_s2_GW)*x8 + c(lambda_s3

# Hypotheses
H1_mgcfa <- "
abs(lambda_v1_P) < abs(lambda_v1_GW); abs(lambda_v2_P) < abs(lambda_v2_GW); abs(lambda_v3_P) < abs(lambd
abs(lambda_t1_P) < abs(lambda_t1_GW); abs(lambda_t2_P) < abs(lambda_t2_GW); abs(lambda_t3_P) < abs(lambd
abs(lambda_s1_P) < abs(lambda_s1_GW); abs(lambda_s2_P) < abs(lambda_s2_GW); abs(lambda_s3_P) < abs(lambd
"
# vs its complement (default)

# fit the model fixing the variances of all the latent variables in a CFA model
# to unity (std.lv = TRUE; https://www.lavaan.ugent.be/tutorial/syntax2.html)
fit_mgcfa <- cfa(HS.model_mgcfa, data = HolzingerSwineford1939, group = "school",
    std.lv = TRUE)

# Calculate GORICA values and weights
set.seed(100)  # Needed for reproducibilty & sensitivity check
results_mgcfa <- goric(fit_mgcfa, hypotheses = list(H1_mgcfa = H1_mgcfa), standardized = TRUE)
```

```
restriktor Message: The covariance matrix of the estimates was obtained via 'vcov()'. This is the biased
```

```
# summary(results_mgcfa)
results_mgcfa
```

```
restriktor (0.6-20): generalized order-restricted information criterion approximation:

Results:
        model  loglik  penalty   gorica  loglik.weights  penalty.weights  gorica.weights
1    H1_mgcfa  33.392   13.450  -39.884           0.146            0.989           0.941
2  complement  35.159   17.985  -34.348           0.854            0.011           0.059

Conclusion:
The order-restricted hypothesis 'H1_mgcfa' has 15.92 times more support than its complement.
```

Conclusion: The order-restricted hypothesis 'H1_mgcfa' has (> 1 times) more support than its complement.

Example 3: Structural equation modeling (SEM)

In this example, we will look at the structural equation modeling (SEM) lavaan example (cf. https://www.lavaan.ugent.be/tutorial/sem.html).

Here, we assume that we are interested in (some of) the regression parameters; so, we only need to label these. We want to compare the strengths of the predictive relationships; hence, we need standardized estimates for a meaningful comparison; that is, add `standardized = TRUE`. We expect that the predictive relationship of ind60 for dem60 is (in absolute sense) higher that that for dem65.

```
# specify the model, using own labeling
model_sem <- "
  # measurement model
    ind60 =~ x1 + x2 + x3
    dem60 =~ y1 + y2 + y3 + y4
    dem65 =~ y5 + y6 + y7 + y8
  # regressions
    dem60 ~ beta_dem60_ind60*ind60
    dem65 ~ beta_dem65_ind60*ind60 + dem60
  # residual correlations
    y1 ~~ y5
    y2 ~~ y4 + y6
    y3 ~~ y7
    y4 ~~ y8
    y6 ~~ y8
"

# Hypotheses
H1_sem <- "
abs(beta_dem60_ind60) > abs(beta_dem65_ind60)
"
# vs its complement (default)

# fit the model
fit_sem <- sem(model_sem, data = PoliticalDemocracy)

# Calculate GORICA values and weights
set.seed(100)  # Needed for reproducibilty & sensitivity check
results_sem <- goric(fit_sem, hypotheses = list(H1_sem = H1_sem), standardized = TRUE)
```

```
restriktor Message: The covariance matrix of the estimates was obtained via 'vcov()'. This is the biase
```

```
# summary(results_sem)
results_sem
```

```
restriktor (0.6-20): generalized order-restricted information criterion approximation:

Results:
        model  loglik  penalty  gorica  loglik.weights  penalty.weights  gorica.weights
1      H1_sem   3.113    1.500  -3.226           0.862            0.500           0.862
2  complement   1.279    1.500   0.443           0.138            0.500           0.138

Conclusion:
The order-restricted hypothesis 'H1_sem' has 6.26 times more support than its complement.
```

Conclusion: The order-restricted hypothesis 'H1_sem' has 6.06 times more support than its complement.

Example 4: Multilevel SEM

In this example, we will look at the multilevel/two-level SEM lavaan example (cf. https://www.lavaan.ugent.be/tutorial/multilevel.html).

Here, we assume that we are interested in (some of) the regression parameters; so, we only need to label these. We want to compare the strengths of the predictive relationships; hence, we need standardized estimates for a meaningful comparison; that is, add `standardized = TRUE`. We expect that the order of the (absolute) height in predictive relationships for fw is x1, x2, x3.

```
# specify the model, using own labeling
model_msem <- "
    level: 1
        fw =~ y1 + y2 + y3
        fw ~ beta_1*x1 + beta_2*x2 + beta_3*x3
    level: 2
        fb =~ y1 + y2 + y3
        fb ~ w1 + w2
"

# Hypotheses
H1_msem <- "
abs(beta_1) > abs(beta_2) > abs(beta_3)
"
# vs its complement (default)

# fit the model
fit_msem <- sem(model = model_msem, data = Demo.twolevel, cluster = "cluster")

# Calculate GORICA values and weights
set.seed(100)  # Needed for reproducibilty & sensitivity check
results_msem <- goric(fit_msem, hypotheses = list(H1_msem = H1_msem), standardized = TRUE)
```

```
restriktor Message: The covariance matrix of the estimates was obtained via 'vcov()'. This is the biased
```

```
# summary(results_msem)
results_msem
```

```
restriktor (0.6-20): generalized order-restricted information criterion approximation:

Results:
        model  loglik  penalty   gorica  loglik.weights  penalty.weights  gorica.weights
1     H1_msem   9.146    1.827  -14.637           0.995            0.700           0.998
2  complement   3.848    2.673   -2.350           0.005            0.300           0.002

Conclusion:
The order-restricted hypothesis 'H1_msem' has 465.55 times more support than its complement.
```

Conclusion: The order-restricted hypothesis 'H1_msem' has 465.55 times more support than its complement.

Example 5: linear growth model with a time-varying covariate

In this example, we will look at the linear growth model with a time-varying covariate lavaan example (cf. https://www.lavaan.ugent.be/tutorial/growth.html).

Here, we assume that we are interested in the two intercepts; so, we only need to label these. For this, we now

need extra code. We expect that the intercept of the slope s is (in absolute sense) higher than the intercept of the intercept i.

```
# specify the model, using own labeling a linear growth model with a
# time-varying covariate
model_growth <- "
  # intercept and slope with fixed coefficients
    i =~ 1*t1 + 1*t2 + 1*t3 + 1*t4
    s =~ 0*t1 + 1*t2 + 2*t3 + 3*t4
  # regressions
    i ~ x1 + x2
    s ~ x1 + x2
  # time-varying covariates
    t1 ~ c1
    t2 ~ c2
    t3 ~ c3
    t4 ~ c4
  # Extra: Label intercepts
  i ~ intercept_i * 1
  s ~ intercept_s * 1
"

# Hypotheses
H1_growth <- "
abs(intercept_s) > abs(intercept_i)
"
# vs its complement (default)

# fit the model
fit_growth <- growth(model_growth, data = Demo.growth)

# Calculate GORICA values and weights
set.seed(100)  # Needed for reproducibilty & sensitivity check
results_growth <- goric(fit_growth, hypotheses = list(H1_growth = H1_growth), standardized = TRUE)
```

```
restriktor Message: The covariance matrix of the estimates was obtained via 'vcov()'. This is the biase
```

```
# summary(results_growth)
results_growth
```

```
restriktor (0.6-20): generalized order-restricted information criterion approximation:

Results:
        model   loglik  penalty  gorica  loglik.weights  penalty.weights  gorica.weights
1   H1_growth    3.977    1.500  -4.954           1.000            0.500           1.000
2  complement  -47.034    1.500  97.068           0.000            0.500           0.000

Conclusion:
The order-restricted hypothesis 'H1_growth' has 14251239710235289453804.00 times more support than its
```

Conclusion: The order-restricted hypothesis ‘H1_growth’ has many more (nl., 14251239710235289453804.00) times more support than its complement.

Example 6: Mediation

In this example, we will look at the mediation lavaan example (cf. https://www.lavaan.ugent.be/tutorial/mediation.html).

Let us say we want to evaluate whether there is partial mediation or full mediation. It can be helpful to specify what you believe is the minimum effect. Here, we assume that an (in)direct effect between -0.1 and 0.1 can be seen as no effect. Note: More examples can found on https://github.com/rebeccakuiper/Tutorials/tree/main/GORICA%20for%20mediation.

```
# Create data
set.seed(1234)
X <- rnorm(100)
M <- 0.5 * X + rnorm(100)
Y <- 0.7 * M + rnorm(100)
Data <- data.frame(X = X, Y = Y, M = M)

# specify the model, using own labeling
model_med <- " # direct effect
              Y ~ c*X
            # mediator
              M ~ a*X
              Y ~ b*M
            # indirect effect (a*b)
              indirect := a*b
            # direct effect (c)
              direct := c
          "

# Hypotheses
H_part <- "abs(indirect) > 0.1; abs(direct) > 0.1"
H_full <- "abs(indirect) > 0.1; -0.1 < direct < 0.1"
# and unconstrained as failsafe (default)

# fit the model
fit_med <- sem(model_med, data = Data)
# summary(fit_med, standardized = TRUE)

# Calculate GORICA values and weights Based on object
set.seed(100)  # Needed for reproducibilty & sensitivity check
results_med <- goric(fit_med, hypotheses = list(H_part = H_part, H_full = H_full),
    standardized = TRUE)
```

```
restriktor Message: The covariance matrix of the estimates was obtained via 'vcov()'. This is the biased
```

```
# summary(results_med)
results_med
```

```
restriktor (0.6-20): generalized order-restricted information criterion approximation:

Results:
           model  loglik  penalty   gorica  loglik.weights  penalty.weights  gorica.weights  gorica.weig
1         H_part   8.382    3.922   -8.919           0.249            0.349           0.262
2         H_full   8.795    3.500  -10.590           0.376            0.532           0.604
3  unconstrained   8.795    5.000   -7.590           0.376            0.119           0.135
```

```
Conclusion:
- The order-restricted hypothesis 'H_full' is the best in the set, as it has the highest GORIC(A) weigh
- Since 'H_full' has a higher GORIC(A) weight than the unconstrained hypothesis, it is not considered w
  * 'H_full' is 2.306 times more supported than 'H_part'.
```

```
# # Based on extracted estimates # Extract standardized estimates of the
# defined parameters and their var-cov matrix label_names <- c('direct',
# 'indirect') est <-
# as.vector(standardizedSolution(fit_med)['est.std'])$est.std names(est) <-
# standardizedSolution(fit_med)$label est <- est[label_names] #est VCOV <-
# lavInspect(fit_med, 'vcov.def.std.all')[label_names, label_names] # VCOV
# matrix of parameters # GORICA set.seed(123) # for reproducibility & possibly
# sensitivity check results_med <- goric(est, VCOV = VCOV, hypotheses =
# list(H_part = H_part, H_full = H_full)) #summary(results_med) results_med
```

Conclusion: - The order-restricted hypothesis ‘H_full’ is the best in the set, as it has the highest GORIC(A) weight. - Since ‘H_full’ has a higher GORIC(A) weight than the unconstrained hypothesis, it is not considered weak. We can now inspect the relative support for ‘H_full’ against the other order-restricted hypotheses: * ‘H_full’ is 2.306 times more supported than ‘H_part’.