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

Rebecca M. Kuiper and Yves Rosseel

08 december 2025

Contents

This is a tutorial for applying the AIC-type criterion GORICA, using the `goric` function, 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.

More information (tutorials and example R scripts) is available from this github page of Rebecca, 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.

```r
# To install restriktor in R: if (!require('restriktor'))
# install.packages('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.

To specify your hypotheses in terms of model parameters, you should give your own labels to estimates by including them in the `lavaan` model syntax.

```
# 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` in the `goric` function. As another note, it is also possible to compare the size of (absolute values of) various 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 in the `goric` function (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. Because of entering a lavaan object, we use GORICA (i.e., `type = "gorica"`) by default.

```
# 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 biased

```
# 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 less support (namely, 0.497 / 0.503 < 1 times more suppor

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

Thus, we did not find support for our hypothesis, that is, there is at least one standardized factor loading less than 0.6 (in absolute sense).

Example 2: Multiple Group CFA

In this example, we will extend the CFA with a multi-group structure. 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 hypothesis regarding the latent means of the two schools. Before we do, we need to establish measurement invariance, such that we can fairly compare the means.

Measurement invariance

We could use the GORICA to check for measurement invariance. When you can specify ranges of values for which the parameters, say, loadings, are the same for each group, then there is an added value of using the GORICA. Here, we will inspect the case where we set the the parameters, say, loadings, to be equal across groups. Then, one can use the AIC as well, where we will also determine the AIC weights for a more in-depth comparison.
Note that we compare parameters across groups, so they are already on a comparable scale, since they denote the same relationship in each group. Therefore, we do not need the standardized values.

```
# specify the model
HS.model_mgcfa <- "
             visual  =~ x1 + x2 + x3
             textual =~ x4 + x5 + x6
             speed   =~ x7 + x8 + x9
"

# configural invariance
fit_ci <- cfa(HS.model_mgcfa, data = HolzingerSwineford1939, group = "school")

# weak invariance
fit_wi <- cfa(HS.model_mgcfa, data = HolzingerSwineford1939, group = "school", group.equal = "loadings")

# strong invariance
fit_si <- cfa(HS.model_mgcfa, data = HolzingerSwineford1939, group = "school", group.equal = c("intercep
    "loadings"))

# strict invariance
fit_strict <- cfa(HS.model_mgcfa, data = HolzingerSwineford1939, group = "school",
    group.equal = c("intercepts", "loadings", "residuals"))

# model comparison with AIC
AIC_meas.invar <- lavTestLRT(fit_ci, fit_wi, fit_si, fit_strict)$AIC
hypo.names <- c("configural", "weak", "strong", "strict")
AIC_weights <- calc_ICweights(AIC_meas.invar, hypo.names)
# print(AIC_weights, use_scientific = FALSE, digits = 3)
AIC_weights$ratio_IC_weights
```

```
           vs_configural     vs_weak    vs_strong     vs_strict
configural  1.000000e+00 1.489865e-01 1.845625e+05 1.373106e+05
weak        6.712019e+00 1.000000e+00 1.238787e+06 9.216313e+05
strong      5.418218e-06 8.072412e-07 1.000000e+00 7.439787e-01
strict      7.282760e-06 1.085033e-06 1.344124e+00 1.000000e+00
```

```
AIC_weights$ratio_IC_weights["weak", ]
```

```
vs_configural        vs_weak     vs_strong     vs_strict
 6.712019e+00   1.000000e+00  1.238787e+06  9.216313e+05
```

The weak invariance (equal factor loadings) is the best of the set, since it has the lowest value. From the AIC weights, we can see that it is about 20 times more supported than configural invariance, and many times more supported than strong and strict invariance. Therefore, it is unwise to directly compare the values of the latent means across the two groups.

To still show how one can do that using the GORICA, we do proceed here.

Hypothesis evaluation using GORICA

From here on, we assume that strong (or even strict) measurement invariance was established, such that we can fairly compare the latent means. Let us say that our expectation is the latent means of our three factors are in absolute sense higher in the Pasteur group than in the Grant-White group.
First, we will create the model such that we obtain these latent means. Second, we specify our hypothesis. Third, we run the model. Fourth, we evaluate our hypothesis using the GORICA.

```r
# specify the model, using own labeling (per group!)  Specify model, such that
# we obtain the latent means (and not differences w.r.t. a reference category).
HS.model_mgcfa <- "
             visual  =~ x1 + x2 + x3
             textual =~ x4 + x5 + x6
             speed   =~ x7 + x8 + x9

# latent variable intercepts; here:
# the latent means
   visual  ~ c(mean_v_P, mean_v_GW) * 1
   textual ~ c(mean_t_P, mean_t_GW) * 1
   speed   ~ c(mean_s_P, mean_s_GW) * 1
"
# where below we ensure: intercepts equal across groups $ loadings equal across
# groups: group.equal = c('intercepts', 'loadings') and first intercept equal
# to O: marker.int.zero = TRUE


# Hypotheses
H1_mgcfa <- "
abs(mean_v_P) > abs(mean_v_GW);
abs(mean_t_P) > abs(mean_t_GW);
abs(mean_s_P) > abs(mean_s_GW)
"
# vs its complement (default)

# fit the model
fit_mgcfa <- cfa(HS.model_mgcfa, data = HolzingerSwineford1939, group = "school",
    group.equal = c("intercepts", "loadings"), marker.int.zero = 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))
```

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

```
# Altrenative: In the case you want to extract the estimates and their
# covariance matrix yourself and use that as input for the goric function:
# Extract estimates and their covariance matrix
names_param <- c("mean_v_P", "mean_v_GW", "mean_t_P", "mean_t_GW", "mean_s_P", "mean_s_GW")
est <- coef(fit_mgcfa)[names_param]
VCOV <- vcov(fit_mgcfa)[names_param, names_param]
# Calculate GORICA values and weights
set.seed(100)  # Needed for reproducibilty & sensitivity check
results_mgcfa <- goric(est, VCOV = VCOV, hypotheses = list(H1_mgcfa = H1_mgcfa))

# GORICA values and weights 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  -2.500    4.774  14.549           0.000            0.649           0.000
2  complement   9.594    5.391  -8.407           1.000            0.351           1.000

Conclusion:
The order-restricted hypothesis 'H1_mgcfa' has less support (namely,  0.000 / 1.000 < 1 times more supp
```

Conclusion: The order-restricted hypothesis 'H1_mgcfa' has less support (namely, $<$ 1 times more support) than its complement. That is, there is support for the latent means being in absolute sense higher in the Pasteur group than in the Grant-White group.

Example 3: Structural equation modeling (SEM)

In this example, we will look at the structural equation modeling (SEM) lavaan example.

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` in the `goric` function. One can compare strengths of relationships of various independent variables for a specific dependent variable, but one over different dependent variables.
Here, we compare the strengths of relationships for one dependent variable, namely dem65. More specially, we expect that the predictive relationship of dem60 for dem65 (beta_dem65_dem60) is (in absolute sense) stronger than that of ind60 for dem65 (beta_dem65_ind60).

```
# 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 ~ ind60
    dem65 ~ beta_dem65_ind60*ind60 + beta_dem65_dem60*dem60
  # residual correlations
    y1 ~~ y5
    y2 ~~ y4 + y6
    y3 ~~ y7
    y4 ~~ y8
    y6 ~~ y8
```

```
"

# Hypotheses
H1_sem <- "
abs(beta_dem65_dem60) > 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     4.348    1.500   -5.697           1.000            0.500           1.000
2   complement   -14.069    1.500   31.137           0.000            0.500           0.000

Conclusion:
The order-restricted hypothesis 'H1_sem' has 99620500.72 times more support than its complement.
```

```
# Unrestricted & Order-restricted parameter estimates of interest Note:
# Hypotheses should be specified before seeing the data.  Looking afterwards at
# the standardized estimates can give you more insight.  Unrestricted parameter
# estimates of interest
subset(standardizedSolution(fit_sem), op == "~" & lhs == "dem65")
```

```
     lhs op    rhs              label est.std     se      z pvalue ci.lower ci.upper
13 dem65  ~ ind60 beta_dem65_ind60    0.182  0.070  2.587   0.01    0.044    0.320
14 dem65  ~ dem60 beta_dem65_dem60    0.885  0.051 17.405   0.00    0.786    0.985
```

```
# Order-restricted parameter estimates of interest; that is, the estimates of
# interest under the restrictions in the hypotheses:
results_sem$ormle
```

```
$b.restr
              beta_dem65_ind60 beta_dem65_dem60
H1_sem               0.1822593        0.8852290
complement           0.5959879        0.5959879
```

Conclusion: The order-restricted hypothesis 'H1_sem' has (> 1 times) more support than its complement. Thus, we found support that the predictive relationship of ind60 for dem60 is (in absolute sense) higher that that for dem65.

Example 4: Multilevel SEM

In this example, we will look at the multilevel/two-level SEM lavaan example.

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` in the `goric` function. We expect that the order of the (absolute) strength/size in predictive relationships for fw (beta) is highest for x1, then x2, and then x3, that is, abs(beta_x1) > abs(beta_x2) > abs(beta_x3).

```
# specify the model, using own labeling
model_msem <- "
    level: 1
        fw =~ y1 + y2 + y3
        fw ~ beta_x1*x1 + beta_x2*x2 + beta_x3*x3
    level: 2
        fb =~ y1 + y2 + y3
        fb ~ w1 + w2
"

# Hypotheses
H1_msem <- "
abs(beta_x1) > abs(beta_x2) > abs(beta_x3)
"
# 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 biase
```

```
# 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 (> 1 times) more support than its complement. Thus, we found support that the order of the (absolute) strength in predictive relationships for fw (beta) is highest for x1, then x2, and then x3, that is, abs(beta_x1) > abs(beta_x2) > abs(beta_x3).

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.

Here, we assume that we are interested in the predictive strength of the two predictors (x1 and x2) on the slope (s); so, we only need to label these. We expect that the strength of x2 on the slope (s_2) is (in absolute sense) higher than that of x1 (s_1).

```
# 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 ~ s_1*x1 + s_2*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(s_2) > abs(s_1)
"
# 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	5.210	1.500	-7.419	1.000	0.500	1.000
2	complement	-8.912	1.500	20.824	0.000	0.500	0.000

Conclusion:
The order-restricted hypothesis 'H1_growth' has 1358030.63 times more support than its complement.

Conclusion: The order-restricted hypothesis 'H1_growth' has many more (>> 1 times more) support than its complement. Thus, we found support that the intercept of the slope s is (in absolute sense) higher than the intercept of the intercept i.

Example 6: Mediation

In this example, we will look at the mediation lavaan example.

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. More examples can found on this github page of Rebecca.

```r
# 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: 1. based on the 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 biase

```r
# 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.wei
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 we
  * 'H_full' is 2.306 times more supported than 'H_part'.

```r
# 2. based on extracted estimates Extract standardized estimates of the defined
# parameters and their covariance 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
```

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

Results:

	model	loglik	penalty	gorica	loglik.weights	penalty.weights	gorica.weights	gorica.weigh
1	H_part	3.098	0.922	-4.351	0.249	0.349	0.262	
2	H_full	3.511	0.500	-6.022	0.376	0.532	0.604	
3	unconstrained	3.511	2.000	-3.022	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 we
  * 'H_full' is 2.306 times more supported than 'H_part'.