

Learning Lavaan

Pit Rieger

8/5/2021

1 Models created implicitly by lavaan

Lavaan fits more than just the specified model when fitting a confirmatory factor analysis model. These models are used for tests, etc. In the following, we consider the standard example for CFA that lavaan provides: The Holzinger Swineford dataset and a 3-factor model. We have 9 variables that load as follows:

$$\begin{aligned}visual &\sim x_1 + x_2 + x_3 \\text{tutual} &\sim x_4 + x_5 + x_6 \\speed &\sim x_7 + x_8 + x_9\end{aligned}$$

The **baseline** model is a simple model where we assume no latent structure and zero correlation among the manifest variables.

For the **saturated** model, lavaan allows for arbitrary correlation among all manifest variables such that the covariance matrix is estimated freely.

In this framework, model fit via χ^2 -test is a simple likelihood ratio test:

$$2 \log \frac{\mathcal{L}(\hat{\theta}_{\text{saturated}}^{ML})}{\mathcal{L}(\hat{\theta}_{\text{specified}}^{ML})} \sim \chi_{d.f.}^2$$
$$2 \left\{ \ell(\hat{\theta}_{\text{saturated}}^{ML}) - \ell(\hat{\theta}_{\text{specified}}^{ML}) \right\} \sim \chi_{d.f.}^2$$

The degrees of freedom ($d.f.$) can simply be calculated as the difference between the parameters in the specified model versus the saturated (which has $p(p+1)/2$ parameters).

```
# fit saturated model
HS.fit_sat = cfa(mod_saturated, HolzingerSwineford1939)

# manual likelihood ratio statistic
(LR_stat = 2 * (HS.fit_sat$loglik$loglik - HS.fit$loglik$loglik))
```

```
## [1] 85.30552
```

```
# compute degrees of freedom
(df = (9 * 10 / 2) - # parameters in saturated models
  (9 + # indicator variances
    9 - 3 + # loadings (first fixed to 1 for each LV)
    3 + # latent variable variances
    3)) # latent variable correlations
```

```
## [1] 24
```

```
# summary(HS.fit)

# compare p-values from manual and lavaan
pchisq(LR_stat, df, lower.tail = F)
```

```
## [1] 8.502553e-09
```

```
HS.fit@Fit@test$standard$pvalue
```

```
## [1] 8.502553e-09
```

Similarly, as one would expect, the implied covariance matrix of the saturated model equals the sample covariance matrix. Note that lavaan doesn't apply a Bessel correction (so computes variances with $1/n$ instead of $1/(n - 1)$).

```
X = as.matrix(HolzingerSwineford1939[, 7:15])
X = apply(X, 2, function(x) x - mean(x))
round(
  1/nrow(X) * (t(X) %*% X) - # sample covariance matrix
  HS.fit_sat@implied$cov[[1]], 2) # implied covariance matrix of saturated model
```

```
##      x1 x2 x3 x4 x5 x6 x7 x8 x9
## x1  0  0  0  0  0  0  0  0  0
## x2  0  0  0  0  0  0  0  0  0
## x3  0  0  0  0  0  0  0  0  0
## x4  0  0  0  0  0  0  0  0  0
## x5  0  0  0  0  0  0  0  0  0
## x6  0  0  0  0  0  0  0  0  0
## x7  0  0  0  0  0  0  0  0  0
## x8  0  0  0  0  0  0  0  0  0
## x9  0  0  0  0  0  0  0  0  0
```

These very principles also apply when comparing nested models, e.g. for testing measurement equivalence: Suppose we have one model that fits unconstrained parameters for subpopulations in our data and others which restrict the intercepts or slopes to be the same. We then have the following comparisons in lavaan:

```
fit1 <- cfa(HS.model, data = HolzingerSwineford1939, group = "school")

# weak invariance
fit2 <- cfa(HS.model, data = HolzingerSwineford1939, group = "school",
  group.equal = "loadings")

# strong invariance
fit3 <- cfa(HS.model, data = HolzingerSwineford1939, group = "school",
  group.equal = c("intercepts", "loadings"))

# model comparison tests
lavTestLRT(fit1, fit2, fit3)
```

```
## Chi-Squared Difference Test
##
##      Df      AIC      BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## fit1 48 7484.4 7706.8 115.85
## fit2 54 7480.6 7680.8 124.04      8.192      6      0.2244
## fit3 60 7508.6 7686.6 164.10     40.059      6 4.435e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Which we could also obtain by manually comparing their log likelihoods:

```
2 * (logLik(fit1) - logLik(fit2))
```

```
## 'log Lik.' 8.1922 (df=60)
```

```
2 * (logLik(fit2) - logLik(fit3))
```

```
## 'log Lik.' 40.05929 (df=54)
```

For measurement equivalence, we ideally want the differences between the constrained and unconstrained models to be small such that the H_0 of no difference cannot be rejected. The results therefore suggest that the null of weak invariance cannot be rejected while strong invariance certainly doesn't hold across schools.