Multivariate Models: Robust Maximum Likelihood and Relative Model Fit

EPSY 905: Fundamentals of Multivariate Modeling

Online Lecture #12

THE UNIVERSITY OF KANSAS

In This Lecture

- The version of ML used in lavaan to estimate multivariate linear models
- How that implies a scaled likelihood ratio test
- How R computes all of it for you automatically

EXAMPLE DATA SET

Today's Data Example

- Data are simulated based on the results reported in:

Pajares, F., & Miller, M. D. (1994). Role of self-efficacy and self-concept beliefs in mathematical problem solving: a path analysis. *Journal of Educational Psychology, 86*, 193-203.

- Sample of 350 undergraduates (229 women, 121 men)
  - In simulation, 10% of variables were missing (using missing completely at random mechanism)

- Note: simulated data characteristics differ from actual data (some variables extend beyond their official range)
  - Simulated using Multivariate Normal Distribution
    - Some variables had boundaries that simulated data exceeded
  - Results will not match exactly due to missing data and boundaries

Variables of Data Example

- Female (sex variable: 0 = male; 1 = female)
- Math Self-Efficacy (MSE)
  - Reported reliability of .91
  - Assesses math confidence of college students
- Perceived Usefulness of Mathematics (USE)
  - Reported reliability of .93
- Math Anxiety (MAS)
  - Reported reliability ranging from .86 to .90
- Math Self-Concept (MSC)
  - Reported reliability of .93 to .95
- Prior Experience at High School Level (HSL)
  - Self report of number of years of high school during which students took mathematics courses
- Prior Experience at College Level (CC)
  - Self report of courses taken at college level
- Math Performance (PERF)
  - Reported reliability of .788
  - 18-item multiple choice instrument (total of correct responses)

Model Notation From Last Time

- The multivariate model for PERF and USE is given by two regression models, which are estimated simultaneously:

$$PERF_i = \beta_{0,PERF} + e_{i,PERF}$$
$$USE_i = \beta_{0,USE} + e_{i,USE}$$

- As there are two variables, the error terms have a joint distribution that will be multivariate normal:

$$\begin{bmatrix} e_{i,PERF} \\ e_{i,USE} \end{bmatrix} \sim N_2\left(\mathbf{0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} \sigma^2_{e,PERF} & \sigma_{e,PERF,USE} \\ \sigma_{e,PERF,USE} & \sigma^2_{e,USE} \end{bmatrix}\right)$$

- Model 02 estimated $\sigma_{e,PERF,USE}$
- Model 03 set $\sigma_{e,PERF,USE} = 0$
- We compare these two models

ROBUST MAXIMUM LIKELIHOOD:
AN ESTIMATOR EXCLUSIVE TO PATH ANALYSIS MODELING SOFTWARE

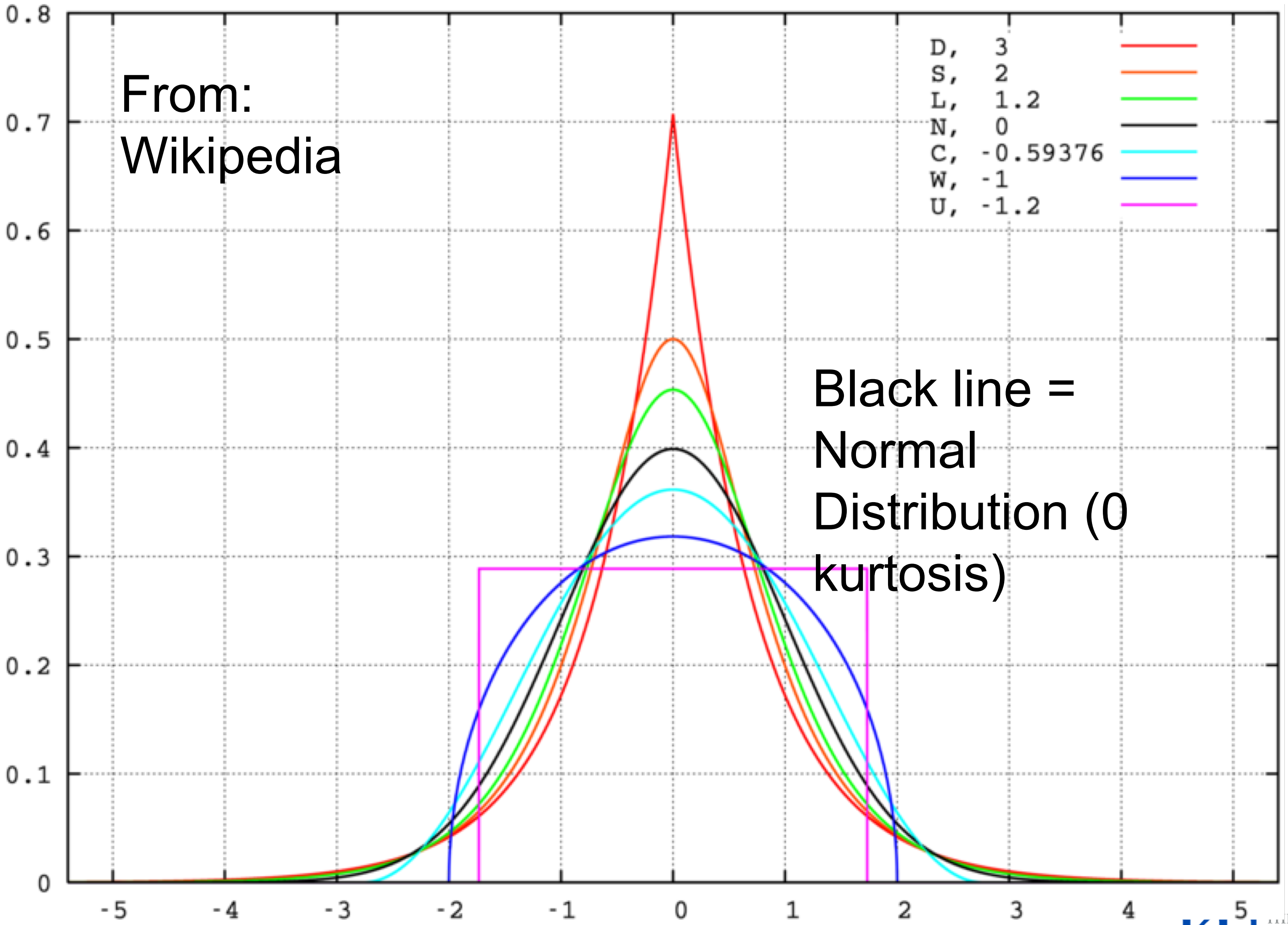
Robust Estimation: The Basics

- Robust estimation in ML still assumes the data follow a multivariate normal distribution
  - But that the data have more or less kurtosis than would otherwise be common in a normal distribution

- Kurtosis: measure of the shape of the distribution
  - From Greek word for bulging
  - Can be estimated for data (either marginally for each item or jointly across all items)

- The degree of kurtosis in a data set is related to how incorrect the log-likelihood value will be
  - Leptokurtic data (too-fat tails): χ^2 inflated, SEs too small
  - Platykurtic data (too-thin tails): χ^2 depressed, SEs too large

Visualizing Kurtosis



Robust ML for Non-Normality in lavaan: MLR

- Robust ML can be specified very easily in lavan:
  - Add estimator = "MLR" to your sem() function call

```
#empty model estimation
model03.fit = sem(model03.syntax, data=job_data, mimic="MPLUS", fixed.x=TRUE, estimator = "MLR")
```

- The model parameter **estimates** will all be identical to those found under regular maximum likelihood
  - And…if data are MVN – then no adjustment is made (so we can use MLR for everything!)

- MLR adjusts:
  - Model χ^2 (and associated fit statistics that use it: RMSEA, CFI, TLI) – closely related to Yuan-Bentler T_2 (permits MCAR or MAR missing data)
  - Model **standard errors**: uses Huber-White "sandwich" estimator to adjust standard errors
    - Sandwich estimator found using information matrix of the partial first derivatives to correct information matrix from the partial second derivatives

Adjusted Model Fit Statistics

- Under MLR, model fit statistics are adjusted based on an estimated scaling factor:
  - Scaling factor = 1.000
    - Perfectly MVN data
  - Scaling factor > 1.000
    - Leptokurtosis (too-fat tails; fixes too big χ^2)
  - Scaling factor < 1.000
    - Platykurtosis (too-thin tails; fixes too small χ^2)

- The scaling factor will now show up in all likelihood ratio tests (deviance tests)
  - So you must add it to your calculations

Adjusted Standard Errors

- The standard errors for all parameter estimates will be different under MLR
  - Remember, these are used in Wald tests

- If the data show leptokurtosis (too-fat tails):
  - Increases information matrix
  - Fixes too small SEs

- If the data show platykurtosis (too-thin tails):
  - Lowers values in information matrix
  - Fixes too big SEs

Data Analysis Example with MLR

- To demonstrate, we will revisit our analysis of the example data for today's class using MLR
- So far, we have estimated two models:
  - Saturated model (Model 02)
  - Independence model (Model 03)
- We compare results of the two estimators (ML v. MLR)
- Because MLR is something does not affect our results if we have MVN data, we can be using MLR for each analysis

lavaan Output: Log-likelihoods Under ML and MLR

- Model 3 Results (Covariance = 0)

Under ML   Under MLR

```
> summary(model03.fit, fit.measures=TRUE)
lavaan (0.5-23.1097) converged normally after  18 iterations

                                                  Used       Total
  Number of observations                           348         350

  Number of missing patterns                         3

  Estimator                                         ML      Robust
  Minimum Function Test Statistic                6.064       5.573
  Degrees of freedom                                 1           1
  P-value (Chi-square)                           0.014       0.018
  Scaling correction factor                                  1.088
    for the Yuan-Bentler correction (Mplus variant)
```

- The actual log-likelihoods are the same

```
Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -2088.064   -2088.064
  Scaling correction factor                                  1.012
    for the MLR correction
  Loglikelihood unrestricted model (H1)      -2085.032   -2085.032
  Scaling correction factor                                  1.028
    for the MLR correction
```

lavaan Output: Log-likelihoods Under ML and MLR

- Model 2 (estimated covariance)

Loglikelihood and Information Criteria:	Under ML	Under MLR
Loglikelihood user model (H0)	-2085.032	-2085.032
Loglikelihood unrestricted model (H1)	-2085.032	-2085.032
Number of free parameters	5	5
Akaike (AIC)	4180.064	4180.064
Bayesian (BIC)	4199.325	4199.325
Sample-size adjusted Bayesian (BIC)	4183.464	4183.464

- The actual log-likelihoods are the same
  - But, under MLR, the log-likelihood gets re-scaled

Adding Scaling Factors to the Analysis

- The MLR-estimated scaling factors are used to rescale the log-likelihoods under LR test model comparisons
  - Extra calculations are needed

- The rescaled LR test is given by:

$$LR_{RS} = \frac{-2\left(\log L_{restricted} - \log L_{full}\right)}{c_{LR}}$$

- The denominator is found by the scaling factors (c) and number of parameters (q) in each model:

$$c_{LR} = \left| \frac{(q_{restricted})(c_{restricted}) - (q_{full})(c_{full})}{\left(q_{restricted} - q_{full}\right)} \right|$$

- Sometimes c_{LR} can be negative - so take the absolute value

THE UNIVERSITY OF KANSAS

Model Comparison: Independence v. Saturated Model

```
> summary(model03.fit, fit.measures=TRUE)
lavaan (0.5-23.1097) converged normally after  18 iterations

                                                  Used       Total
  Number of observations                           348         350

  Number of missing patterns                         3

  Estimator                                         ML      Robust
  Minimum Function Test Statistic                6.064       5.573
  Degrees of freedom                                 1           1
  P-value (Chi-square)                           0.014       0.018
  Scaling correction factor                                  1.088
    for the Yuan-Bentler correction (Mplus variant)
```

```
Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -2088.064   -2088.064
  Scaling correction factor                                  1.012
    for the MLR correction
  Loglikelihood unrestricted model (H1)      -2085.032   -2085.032
  Scaling correction factor                                  1.028
    for the MLR correction
```

- To compare the independence model against the saturated model we must first calculate the scaling factor
  - $q_{full} = 5$ – number of parameters in saturated model
  - $c_{full} = 1.028$ – scaling factor from saturated model
  - $q_{reduced} = 4$ – number of parameters in one-factor model
  - $c_{reduced} = 1.012$ – scaling factor from one-factor model

- The scaling factor for the LR test is then:

$$c_{LR} = \frac{4 * 1.012 - 5 * 1.028}{4 - 5} = \frac{-1.092}{-1} \approx 1.088$$

Model Comparison #1: Independence v. Saturated Model

- The next step is to calculate the re-scaled likelihood ratio test using the original log-likelihoods and the scaling factor:

$$LR_{RS} = \frac{-2\left(\log L_{restricted} - \log L_{full}\right)}{c_{LR}}$$

$$= \frac{-2(-2088.064 - -2085.032)}{1.088} = 107.0225$$

- Finally, we use the rescaled LR test as we would in any other LR test- compare it to a χ^2 with df = difference in number of parameters (here 1)

```
> anova(model03.fit,model02.fit)
Scaled Chi Square Difference Test (method = "satorra.bentler.2001")

            Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
model02.fit  0 4180.1 4199.3 0.0000
model03.fit  1 4184.1 4199.5 6.0641     5.5729       1    0.01824 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Standard Errors/Wald Tests Under MLR

Under ML

Intercepts:

	Estimate	Std.Err	z-value	P(>|z|)
perf	13.966	0.174	80.397	0.000
use	52.500	0.874	60.047	0.000

Variances:

	Estimate	Std.Err	z-value	P(>|z|)
perf	8.751	0.727	12.042	0.000
use	249.201	19.519	12.767	0.000

Under MLR

Intercepts:

	Estimate	Std.Err	z-value	P(>|z|)
perf	13.966	0.174	80.397	0.000
use	52.500	0.874	60.047	0.000

Variances:

	Estimate	Std.Err	z-value	P(>|z|)
perf	8.751	0.756	11.581	0.000
use	249.201	19.212	12.971	0.000

- The SEs of our model under MLR are smaller than the SEs under ML
  - As such, the values of the Wald tests are larger (SEs are the denominator)

MLR: The Take-Home Point

- If you feel you have continuous data that are (tenuously) normally distributed, use MLR
  - Any time you use SEM/CFA/Path Analysis as we have to this point
  - In general, likert-type items with 5 or more categories are treated this way
  - If data aren't/cannot be considered normal we should still use different distributional assumptions

- If data truly are MVN, then MLR doesn't adjust anything

- If data are not MVN (but are still continuous), then MLR adjusts the important inferential portions of the results

WRAPPING UP

Wrapping Up

- Today was an introduction to ML and “Robust” ML estimation for multivariate outcomes using path analysis

- These topics are important when using lavaan as there are quite a few different estimators in the package
  - ML is not always the default