Structural Models: Examining the Distribution of Attributes in Diagnostic Classification Models

NCME 2014 Training Session

Diagnostic Measurement –
Theory, Methods, and Applications

Session 4

Course Website: http://wp.me/p3nkOf-nu

THE UNIVERSITY OF KANSAS

Session Overview

- Session 4 will provide an understanding of structural models used in DCMs
  - What they are: estimates of how attributes are distributed in a sample of examinees
  - How to summarize them: by attribute (marginal probabilities) and by attribute pairs (correlations)
  - Differing types of structural models
    - Mplus: log-linear structural models
    - Other methods not readily available in commercial software

Notation Used Throughout Session

- **Attributes**: $a = 1,\ldots, A$
- **Respondents**: $r = 1,\ldots,R$
- **Attribute Profiles**: $\boldsymbol{\alpha}_r = [\alpha_{r1}, \alpha_{r2},\ldots, \alpha_{rA}]$
  - α_{ra} is 0 or 1
- **Latent Classes**: $c = 1,\ldots,C$
  - We have $C = 2^A$ latent classes – one for each possible attribute profile
- **Items**: $i = 1,\ldots,I$
  - Restricted to dichotomous item responses (X_{ri} is 0 or 1)
- **Q-matrix**: Elements q_{ia} for an item i and attribute a
  - q_{ia} is 0 or 1

Session 4: Diagnostic Classification Structural Models

STRUCTURAL MODELS

DCM Structural Models

- Throughout the workshop, attribute profile base-rates have been mentioned as being influential in DCMs
  - Part of respondent diagnoses: the attribute "base rates"
  - Describes distribution of attribute profiles in a sample
    - Proportion of masters for any given attribute
    - Correlation of attributes

- The base-rates represent the probability any respondent has a given attribute profile

- For a test measuring *A* attributes, 2^A profiles are possible
  - The structural model provides the probability for each profile

DCMs are Constrained Latent Class Models

- Previously we've learned how different DCMs provide different parameterizations of the measurement component of the model
  - The LCDM – and which attributes are specified in the q-matrix
- In this session we'll learn about the parameterization of the structural component of DCMs
  - Choice of structural model not dependent on the measurement component

Observed Data: Probability of observing examinee *r*'s vector of item responses to all *I* items

Measurement Component: Product of Conditional Item Response Probabilities (Item Responses are Independent)

$$P(\mathbf{X}_r = \mathbf{x}_r) = \sum_{c=1}^{C} \nu_c \prod_{i=1}^{I} {\pi_{ic}}^{x_{ri}} (1 - \pi_{ic})^{1-x_{ri}}$$

Structural component: Proportion of examinees in each class

THE UNIVERSITY OF KANSAS

DCM Structural Models – Defined

- The parameter for the structural model is ν_c
- Each attribute profile $\boldsymbol{\alpha}_c$ has one
- ν_c is the base-rate probability of attribute profile c:

$$\nu_c = P(\boldsymbol{\alpha}_c)$$

- The DTMR estimates of ν_c are shown on the next slide

c	ν_c	α_1	α_2	α_3	α_4
1	.212	0	0	0	0
2	.070	0	0	0	1
3	.056	0	0	1	0
4	.084	0	0	1	1
5	.038	0	1	0	0
6	.026	0	1	0	1
7	.050	0	1	1	0
8	.153	0	1	1	1
9	.002	1	0	0	0
10	.003	1	0	0	1
11	.002	1	0	1	0
12	.017	1	0	1	1
13	.003	1	1	0	0
14	.011	1	1	0	1
15	.017	1	1	1	0
16	.255	1	1	1	1

Interpreting the Structural Model

- Because there are numerous ν_c parameters, interpretation is difficult
  - Useful for detecting attribute hierarchies
- Often, the ν_c parameters are re-expressed as:
  - The marginal probability an attribute is mastered in the population
  - The correlation between any two attributes
- Both can be computed using a frequency analysis weighted by ν_c

SAS Structural Model Summary

- SAS can be used to compute summaries of the structural model parameters

```
DATA structural;
INPUT class eta alpha1-alpha4;
DATALINES;
1  .21247 0 0 0 0
2  .06991 0 0 0 1
3  .05575 0 0 1 0
4  .08358 0 0 1 1
5  .03801 0 1 0 0
6  .02582 0 1 0 1
7  .04954 0 1 1 0
8  .15337 0 1 1 1
9  .00226 1 0 0 0
10 .00354 1 0 0 1
11 .00234 1 0 1 0
12 .01671 1 0 1 1
13 .00337 1 1 0 0
14 .01089 1 1 0 1
15 .01733 1 1 1 0
16 .25512 1 1 1 1
;
RUN;

PROC FREQ DATA=structural;
WEIGHT eta;
TABLE alpha1-alpha4;
TABLE alpha1*alpha2 alpha1*alpha3 alpha1*alpha4
      alpha2*alpha3 alpha2*alpha4 alpha3*alpha4 / PLCORR;
RUN;
```

SAS Structural Model Summary

- For each attribute, marginally:

Proportion of Masters

alpha1	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	0.68845	68.84	0.68845	68.84
1	0.31156	31.16	1.00001	100.00

alpha2	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	0.44656	44.66	0.44656	44.66
1	0.55345	55.34	1.00001	100.00

alpha3	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	0.36627	36.63	0.36627	36.63
1	0.63374	63.37	1.00001	100.00

alpha4	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	0.38107	38.11	0.38107	38.11
1	0.61894	61.89	1.00001	100.00

SAS Structural Model Summary

- For each pair of attributes:

Tetrachoric Correlation

Statistics for Table of alpha1 by alpha2

Statistic	Value	ASE
Gamma	0.8961	0.6958
Kendall's Tau-b	0.4963	0.7247
Stuart's Tau-c	0.4571	0.7520
Somers' D C|R	0.5328	0.7618
Somers' D R|C	0.4624	0.7542
Pearson Correlation	0.4963	0.7247
Spearman Correlation	0.4963	0.7247
Tetrachoric Correlation	0.7809	0.8990
Lambda Asymmetric C|R	0.3470	1.5014
Lambda Asymmetric R|C	0.0641	2.3100
Lambda Symmetric	0.2308	1.6109
Uncertainty Coefficient C|R	0.2053	0.6539
Uncertainty Coefficient R|C	0.2275	0.7020
Uncertainty Coefficient Symmetric	0.2159	0.6733

Attribute Summary

- For the DTMR data, we have the following summary of attribute summary information

Attribute	Prop. Master		Tetrachoric Correlations			
1. Referent Unit	.312					
2. Partitioning/Iterating	.553		.781			
3. Appropriateness	.634		.723	.740		
4. Multiplicative Comparison	.619		.703	.626	.711	

- Such information is helpful in determining nature of attributes in a population of interest
  - Analogous to information about latent variables in CFA/MIRT

Differing Structural Models

- The structural model of a DCM has the potential to have an overwhelming number of parameters
  - For A attributes, total estimated: 2^A-1
    - All must sum to 1
    - ***Saturated model***

- Multiple structural models exist
  - All reduce the number of parameters
  - All use categorical data analysis techniques to model ν_c

- Analogous to latent variable covariance structure in structural equation modeling
  - Distribution of attributes is categorical, not continuous
  - Can help to determine nature of attribute relationships

Types of Structural Models

- **Log-linear model**
  - Models the natural logarithm of ν_c by the attributes in each profile
  - Allows for varying levels of complexity
    - Most: Saturated Model – full set of parameters
    - Least: Independent Attributes Model – no parameters
  - Implemented in Mplus and main focus of discussion today

- **Tetrachoric correlation model**
  - Provides an item factor model for latent attributes
  - Uses only bivariate information for pairs of attributes
  - Allows for covariance structures to be estimated
  - Not available in any software packages (see Templin & Henson, 2006)

- **Hierarchical factors model**
  - Special case of tetrachoric correlation model (see de la Torre & Douglas, 2004)

- **Mixture models**
  - Henson and Templin (2005): used to evaluate types of pathological gamblers
  - Also given by von Davier (2008)

- All of these are described in Chapter 8 of Rupp et al. (2010)

LOG-LINEAR STRUCTURAL MODELS

The Logic Behind Log-Linear Models

- Log-linear models take the set of probabilities from the structural model and re-express them on the log scale

$$\mu_c = \log \nu_c$$

- Re-expression on the log scale is convenient as these terms can now be modeled (predicted) by other features in the model
  - The attributes themselves
  - Covariates (if any)

- Because of the re-expression, redundant terms can be removed from the model
  - Simplifying estimation, improving parsimony

- In a structural model, there are 2^A probabilities…but they all add up to 1.0
  - Therefore there can only be at most $2^A - 1$ log-linear model parameters

Log-Linear Structural Models

- The log-linear structural model is the easiest to implement with Mplus
  - Due to its availability in Mplus (called a latent variable mean)

- μ_c is the natural logarithm of ν_c:

$$\mu_c = \log \nu_c$$

- We can convert from μ_c back to probabilities:

$$\nu_c = \frac{\exp(\mu_c)}{\sum_{i=1}^{2^A} \exp(\mu_i)}$$

DTMR Latent Variable Means

- Mplus fixes the value of the last class "mean" to zero…
- So the rest are in reference to this last class

```
Categorical Latent Variables

 Means
    C#1                -0.183      0.142     -1.288      0.198
    C#2                -1.295      0.259     -5.006      0.000
    C#3                -1.521      0.341     -4.455      0.000
    C#4                -1.116      0.307     -3.637      0.000
    C#5                -1.904      0.314     -6.058      0.000
    C#6                -2.290      0.439     -5.213      0.000
    C#7                -1.639      0.307     -5.347      0.000
    C#8                -0.509      0.226     -2.257      0.024
    C#9                -4.725      0.783     -6.036      0.000
    C#10               -4.277      0.680     -6.288      0.000
    C#11               -4.690      0.700     -6.704      0.000
    C#12               -2.726      0.537     -5.076      0.000
    C#13               -4.327      0.674     -6.422      0.000
    C#14               -3.154      0.550     -5.737      0.000
    C#15               -2.689      0.381     -7.049      0.000
```

- We can overcome this by subtracting off what the last class would have been from every cell

Log-Linear Model Set Up

- It is important to remember that the structural model is a re-expression of the probability of any examinee having a given attribute pattern
  - The “saturated” log-linear model has as many parameters as possible ($2^A - 1$ for a test measuring A attributes)

- In our example, we have 4 attributes (16 probabilities, 15 of which must be estimated)
  - Mplus fixes the value of the last class to zero

- Our parameterization (and the Mplus implementation) will reflect this constraint
  - We will therefore omit what we will learn to be an intercept

Log-Linear Structural Model Notation

- Like the LCDM, the log-linear structural model parameters have several subscripts:

$$\gamma_{\boxed{e,}\boxed{(a_1,\dots)}}$$

- Subscript #1 – e: the level of the effect
  - 0 would be the intercept – but we won't have one
  - 1 is the main effect
  - 2 is the two-way interaction
  - 3 is the three-way interaction
- Subscript #2 – $(a_1,\dots)$: the attributes the effect applies to
  - Same number of attributes listed as number in Subscript #2

Log-Linear Model for μ_c

- The structural model then uses an ANOVA-like model to predict the value of μ_c as a function of the attributes that are defined in attribute profile *c*
  - Shown for 4-attribute model (used in the DTMR)
  - Includes main effects, 2-way, 3-way, and 4-way interactions

- The general model is given by:

Main effects

$$\mu_c = \gamma_{1,(1)}(\alpha_{c1}) + \gamma_{1,(2)}(\alpha_{c2}) + \gamma_{1,(3)}(\alpha_{c3}) + \gamma_{1,(4)}(\alpha_{c4})$$
$$+\gamma_{2,(1,2)}(\alpha_{c1})(\alpha_{c2}) + \gamma_{2,(1,3)}(\alpha_{c1})(\alpha_{c3}) + \gamma_{2,(1,4)}(\alpha_{c1})(\alpha_{c4})$$
$$+\gamma_{2,(2,3)}(\alpha_{c2})(\alpha_{c3}) + \gamma_{2,(2,4)}(\alpha_{c2})(\alpha_{c4}) + \gamma_{2,(3,4)}(\alpha_{c3})(\alpha_{c4})$$
$$+\gamma_{3,(1,2,3)}(\alpha_{c1})(\alpha_{c2})(\alpha_{c3}) + \gamma_{3,(1,2,4)}(\alpha_{c1})(\alpha_{c2})(\alpha_{c4})$$
$$+\gamma_{3,(2,3,4)}(\alpha_{c2})(\alpha_{c3})(\alpha_{c4}) + \gamma_{4,(1,2,3,4)}(\alpha_{c1})(\alpha_{c2})(\alpha_{c3})(\alpha_{c4})$$

Log-Linear Model Explained

- Because not all attribute profiles include all attributes, only some terms get used to predict each value of μ_c

- For profile 1: $\boldsymbol{\alpha}_1 = [\alpha_{11} = 0; \alpha_{12} = 0; \alpha_{13} = 0; \alpha_{14} = 0]$:

$$\begin{gathered}
\mu_1 = \gamma_{1,(1)}(0) + \gamma_{1,(2)}(0) + \gamma_{1,(3)}(0) + \gamma_{1,(4)}(0) \\
+\gamma_{2,(1,2)}(0)(0) + \gamma_{2,(1,3)}(0)(0) + \gamma_{2,(1,4)}(0)(0) \\
+\gamma_{2,(2,3)}(0)(0) + \gamma_{2,(2,4)}(0)(0) + \gamma_{2,(3,4)}(0)(0) \\
+\gamma_{3,(1,2,3)}(0)(0)(0) + \gamma_{3,(1,2,4)}(0)(0)(0) \\
+\gamma_{3,(2,3,4)}(0)(0)(0) + \gamma_{4,(1,2,3,4)}(0)(0)(0)(0)
\end{gathered}$$

  - As all attributes are zero, the predicted value of $\mu_1 = 0$

- Although this may seem counter-intuitive, this is our constraint
  - We only get 15 parameters, not 16
  - The value of μ_1 is relative – the probability ν_1 depends on the other terms in the model

Log-Linear Model Explained

- For profile 2: $\boldsymbol{\alpha}_2 = [\alpha_{21} = 0; \alpha_{22} = 0; \alpha_{23} = 0; \alpha_{24} = 1]$;

$$\begin{aligned}\mu_2 = \gamma_{1,(1)}(0) + \gamma_{1,(2)}(0) + \gamma_{1,(3)}(0) + \boxed{\gamma_{1,(4)}(1)} \\ +\gamma_{2,(1,2)}(0)(0) + \gamma_{2,(1,3)}(0)(0) + \gamma_{2,(1,4)}(0)(1) \\ +\gamma_{2,(2,3)}(0)(0) + \gamma_{2,(2,4)}(0)(1) + \gamma_{2,(3,4)}(0)(1) \\ +\gamma_{3,(1,2,3)}(0)(0)(0) + \gamma_{3,(1,2,4)}(0)(0)(1) \\ +\gamma_{3,(2,3,4)}(0)(0)(1) + \gamma_{4,(1,2,3,4)}(0)(0)(0)(1)\end{aligned}$$

  - The main effect of attribute 4 only applies

$$\mu_4 = \gamma_{1,(4)}$$

Log-Linear Model Explained

- For profile 6: $\boldsymbol{\alpha}_6 = [\alpha_{61} = 0; \alpha_{62} = 1; \alpha_{63} = 0;\ \alpha_{64} = 1]$:

$$\mu_6 = \gamma_{1,(1)}(0) + \boxed{\gamma_{1,(2)}(1)} + \gamma_{1,(3)}(0) + \boxed{\gamma_{1,(4)}(1)}$$
$$+\gamma_{2,(1,2)}(0)(1) + \gamma_{2,(1,3)}(0)(0) + \gamma_{2,(1,4)}(0)(1)$$
$$+\gamma_{2,(2,3)}(1)(0) + \boxed{\gamma_{2,(2,4)}(1)(1)} + \gamma_{2,(3,4)}(0)(1)$$
$$+\gamma_{3,(1,2,3)}(0)(1)(0) + \gamma_{3,(1,2,4)}(0)(1)(1)$$
$$+\gamma_{3,(2,3,4)}(1)(0)(1) + \gamma_{4,(1,2,3,4)}(0)(1)(0)(1)$$

  - The main effects of attribute 2 and attribute 4, and interaction between attributes 2 and 4 apply

$$\mu_6 = \gamma_{1,(2)} + \gamma_{1,(4)} + \gamma_{2,(2,4)}$$

THE UNIVERSITY OF KANSAS

Log-Linear Model Explained

- For profile 16: $\boldsymbol{\alpha}_{16} = [\alpha_{16,1} = 1; \alpha_{16,2} = 1; \alpha_{16,3} = 1; \alpha_{16,4} = 1]$:

$$\begin{gathered}\mu_{16} = \gamma_{1,(1)}(1) + \gamma_{1,(2)}(1) + \gamma_{1,(3)}(1) + \gamma_{1,(4)}(1) \\ +\gamma_{2,(1,2)}(1)(1) + \gamma_{2,(1,3)}(1)(1) + \gamma_{2,(1,4)}(1)(1) \\ +\gamma_{2,(2,3)}(1)(1) + \gamma_{2,(2,4)}(1)(1) + \gamma_{2,(3,4)}(1)(1) \\ +\gamma_{3,(1,2,3)}(1)(1)(1) + \gamma_{3,(1,2,4)}(1)(1)(1) \\ +\gamma_{3,(2,3,4)}(1)(1)(1) + \gamma_{4,(1,2,3,4)}(1)(1)(1)(1)\end{gathered}$$

  - All parameters apply

$$\begin{gathered}\mu_{16} = \\ \gamma_{1,(1)} + \gamma_{1,(2)} + \gamma_{1,(3)} + \gamma_{1,(4)} + \gamma_{2,(1,2)} + \gamma_{2,(1,3)} + \gamma_{2,(1,4)} \\ +\gamma_{2,(2,3)} + \gamma_{2,(2,4)} + \gamma_{2,(3,4)} + \gamma_{3,(1,2,3)} + \gamma_{3,(1,2,4)} + \gamma_{3,(2,3,4)} + \gamma_{4,(1,2,3,4)}\end{gathered}$$

Interpretations of Model Parameters

- The log-linear model with ALL main effects and interactions is statistically equivalent to the saturated structural model

- Two-way interactions are analogous to bivariate correlations in categorical models
  - Higher-level interactions represent higher level of characteristics of attribute distribution (i.e., skewness, kurtosis, etc…)

- Models without interactions imply uncorrelated attributes
  - Main effects are essentially attribute base-rates

- Models without main effects or interactions assume all attribute profiles are equally likely

- Higher order interactions can be removed if not significantly different from zero

LOG-LINEAR STRUCTURAL MODELS IN MPLUS

Implementing Log-Linear Structural Models with Mplus

- Implementation of log-linear structural models in Mplus is much like implementation of the LCDM
  - Labeling parameters
  - Creating new parameters
  - Expressing the labeled parameters as a function of the new parameters through model constraints

- Note: if no structural model is specified, the Mplus default is the saturated structural model

Our Previous Analysis

- From our DTMR analysis, we initially get the output for our structural model:

```
FINAL CLASS COUNTS AND PROPORTIONS FOR THE LATENT CLASSE
BASED ON THE ESTIMATED MODEL

    Latent
   Classes

      1        210.34540          0.21247
      2         69.20994          0.06991
      3         55.19197          0.05575
      4         82.74175          0.08358
      5         37.63398          0.03801
      6         25.56661          0.02582
      7         49.04489          0.04954
      8        151.80987          0.15334
      9          2.24094          0.00226
     10          3.50645          0.00354
     11          2.32019          0.00234
     12         16.54147          0.01671
     13          3.33667          0.00337
     14         10.77973          0.01089
     15         17.15841          0.01733
     16        252.57174          0.25512
```

Categorical Latent Variable Means Output

- From our DTMR analysis, a there was a section of output that we overlooked:

```
Categorical Latent Variables

 Means
    C#1                -0.183      0.142     -1.288      0.198
    C#2                -1.295      0.259     -5.006      0.000
    C#3                -1.521      0.341     -4.455      0.000
    C#4                -1.116      0.307     -3.637      0.000
    C#5                -1.904      0.314     -6.058      0.000
    C#6                -2.290      0.439     -5.213      0.000
    C#7                -1.639      0.307     -5.347      0.000
    C#8                -0.509      0.226     -2.257      0.024
    C#9                -4.725      0.783     -6.036      0.000
    C#10               -4.277      0.680     -6.288      0.000
    C#11               -4.690      0.700     -6.704      0.000
    C#12               -2.726      0.537     -5.076      0.000
    C#13               -4.327      0.674     -6.422      0.000
    C#14               -3.154      0.550     -5.737      0.000
    C#15               -2.689      0.381     -7.049      0.000
```

- You will note, Mplus only gives 15 of these

- The last class “mean” is fixed to zero in Mplus
  - We will have to build code to get around this

The Interplay Between Latent Variable Means and Class Probabilities

- The latent variable means are directly related to the class probabilities using the conversion formula:

$$\nu_c = \frac{\exp(\mu_c)}{\sum_{i=1}^{2^A} \exp(\mu_i)}$$

- Under the spreadsheet for this session, the conversion from μ_c to ν_c is shown

- Our next step is to implement the log-linear structural model as a series of main effects and interactions

Mplus Syntax for Log-Linear Structural Model

- The Mplus syntax must be modified to add the log-linear structural model
  - Labeling parameters
  - Creating new parameters
  - Expressing the labeled parameters as a function of the new parameters through model constraints

Labeling Parameters

- Under the MODEL: section, the latent class means are accessed and labeled using the following syntax:

```
%OVERALL%

[c#1] (m1);  ! Latent variable mean for class 1
[c#2] (m2);  ! Latent variable mean for class 2
[c#3] (m3);  ! Latent variable mean for class 3
[c#4] (m4);  ! Latent variable mean for class 4
[c#5] (m5);  ! Latent variable mean for class 5
[c#6] (m6);  ! Latent variable mean for class 6
[c#7] (m7);  ! Latent variable mean for class 7
```

- Goes under %OVERALL% command
- The parameter goes in brackets (Mplus syntax for means)
- The name of the class (here C) is the name used under the CLASSES = C(16); line, in the VARIABLE: section
- Our labels are M1 – M15 (arbitrary names, must be unique)
  - There are only 15 – the 16^{th} cannot be specified as it is fixed to zero

Creating New Parameters

- The next step is to create the set of new log-linear structural model parameters

```
NEW(G_0 G_11 G_12 G_13 G_14 G_212 G_213 G_214 G_223 G_224 G_234);
```

- These go under MODEL CONSTRAINT
- The labels are arbitrary (our follow LCDM labels)

- NOTE: The DTMR analysis used only up to two-way interaction parameters

Expressing Labels as Functions of New Parameters

- The syntax for expressing the M1-M15 labeled parameters as our log-linear parameters can be complicated
  - The issue is the last class being 0 by default
  - We must subtract our model's last class from all other classes
  - This makes it so $\mu_1 = 0$ – as specified in our model

```
m1=  - (G_11+G_12+G_13+G_14+G_212+G_213+G_214+G_223+G_224+G_234);
m2= G_14 - (G_11+G_12+G_13+G_14+G_212+G_213+G_214+G_223+G_224+G_234);
m3= G_13 - (G_11+G_12+G_13+G_14+G_212+G_213+G_214+G_223+G_224+G_234);
```


THE UNIVERSITY OF KANSAS

Mplus Syntax from SAS Macro

- The SAS macro can assist somewhat in the building of syntax:
- Set structon = 1
- ```
  * Use structural model(0=N,1=Y);              %LET structon=      0;
  * Order of interaction in structural model;   %LET structorder=   2;
  ```
- Note: the syntax created by the macro will exceed 90 characters per line (the Mplus limit)
  - So you will have to go through the input file by hand to fix this before running Mplus
  ```

Log-Linear Model for DTMR

- To demonstrate the log-linear model, we again present our DTMR data
  - Two-way interaction model

```
New/Additional Parameters
   G_0                -0.183      0.142     -1.288      0.198
   G_11               -4.542      0.758     -5.989      0.000
   G_12               -1.721      0.319     -5.390      0.000
   G_13               -1.338      0.358     -3.742      0.000
   G_14               -1.112      0.241     -4.618      0.000
   G_212               2.119      0.515      4.117      0.000
   G_213               1.373      0.557      2.465      0.014
   G_214               1.559      0.488      3.196      0.001
   G_223               1.603      0.374      4.285      0.000
   G_224               0.725      0.362      2.005      0.045
   G_234               1.517      0.348      4.354      0.000
```

Investigating Log-Linear Structural Model Parameters

- To further investigate these parameters, we return to the Excel worksheet for this section of the workshop

- Go to the tab labeled “Saturated Log-linear Model”

Session 4: Diagnostic Classification Structural Models

CONCLUDING REMARKS

Session 4 – Take-home Points

- DCM Structural Models describe the distribution of attributes
  - Means
  - Correlations
  - Overall structure

- Log-linear structural models are implemented in Mplus
  - Provide great flexibility in terms of number of parameters
  - Allow for ability to detect higher order structures
    - Attribute hierarchies
  - Allow for potential to model attributes using covariates