

## Tutorial 8C: Using Latent GOLD Choice to Estimate Discrete Choice Models Using Latent Class Trees

In this tutorial, we re-examine the MaxDiff data (same data used in [Tutorial 8A](#)) to introduce latent class tree models (LCT) models and interpret the resulting output. For a general introduction to LCT modeling see van den Bergh, van Kollenberg, and Vermunt ([PDF](#)).

In this tutorial, you will use a special experimental version of Latent GOLD to:

- Re-estimate the standard latent class “best-worst” choice model from Tutorial 8A
- Generate a latent class tree (LCT) model for these data
- Compare the resulting classes obtained from both approaches
- Explore the new graphical and tabular output generated for latent class choice tree models

### The Goal

Our goal is to illustrate the LCT alternative to the standard LC discrete choice modeling approach by comparing both approaches in this application.

### Introduction to Latent Class Choice Tree Modeling

The first step in latent class tree modeling is to explore the results obtained from a standard latent class approach (see [Choice Tutorial 8A](#)). In Choice Tutorial 8A, it was determined that the best solution was an 8-class model. The results for various standard latent class models are summarized in Figure 1.<sup>1</sup>

---

<sup>1</sup> Note: in Choice Tutorial 8A, the best solution was determined to be the eight-class solution with two scale factors. For simplicity, the scale factors are not included in the example used in this tutorial. The latent class models with and without scale factors indicate that the eight-class model. For an example that includes both latent class trees and scale factors, see Exercise E1.2 at the end of this tutorial.

Figure 1. Fit statistics for standard latent class models (1-, 2-, 3-, 4-, 5-, 6-, 7-, 8-, and 9-class).

File name:											
File size:		109832 bytes	4800 records								
File date:		2017-Nov-24	07:45:47								
Definition:		3cl2scl.lgf									
		LL	BIC(LL)	Npar	L <sup>2</sup>	df	p-value	Class.Err.	R <sup>2</sup> (0)	R <sup>2</sup>	
1-class	1-Class BestWorst	-3704.4130	7451.2125	8	5289.4990	192	3.4e-972	0.0000	0.1997	0.1056	
2-class	2-Class BestWorst	-3576.4621	7242.9955	17	5033.5972	183	4.1e-925	0.0622	0.2621	0.1760	
3-class	3-Class BestWorst	-3479.1023	7095.9608	26	4838.8776	174	7.6e-891	0.0761	0.3461	0.2694	
4-class	4-Class BestWorst	-3419.4811	7024.4034	35	4719.6353	165	2.1e-872	0.0787	0.3765	0.3038	
5-class	5-Class BestWorst	-3366.0531	6965.2322	44	4612.7793	156	1.4e-856	0.0687	0.3879	0.3163	
6-class	6-Class BestWorst	-3320.4852	6921.7812	53	4521.6434	147	4.2e-844	0.0785	0.4087	0.3394	
7-class	7-Class BestWorst	-3289.8976	6908.2909	62	4460.4683	138	5.4e-838	0.0765	0.4176	0.3493	
8-class	8-Class BestWorst	-3252.5109	6881.2024	71	4385.6949	129	4.2e-829	0.0799	0.4201	0.3521	
9-class	9-Class BestWorst	-3235.2497	6894.3647	80	4351.1724	120	8.7e-829	0.0747	0.4252	0.3578	
Model10	0-Class BestWorst										

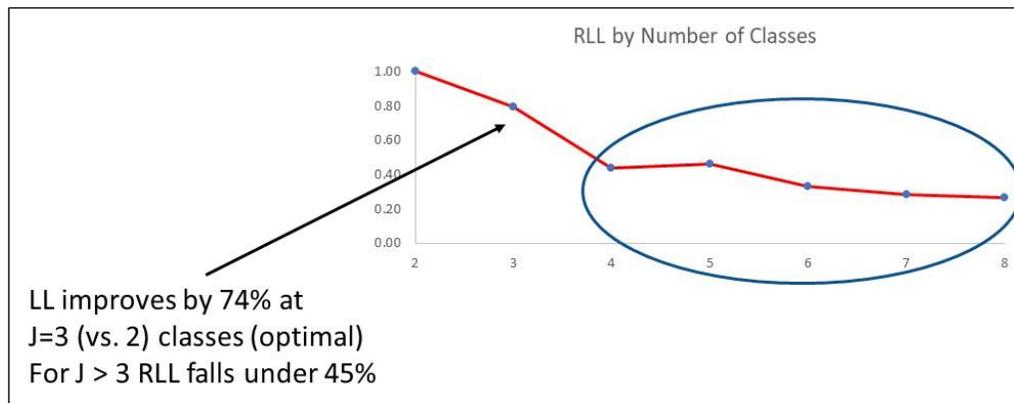
The results suggest that an 8-class model provides the best fit. The substantive interpretation of eight classes, however, can present a challenge because of the large number of classes. One solution to this challenge is to estimate a relatively small number of “theme classes” to be used as a starting point to define the “root” of the LC tree. These theme classes are then viewed as parent classes and each may be split into additional “child classes” as a more complete LC tree is grown.

### Identification and Enumeration of Substantively Meaningful “Parent Level” Classes

Traditional *exploratory* LC relies on information criteria such as BIC (Bayesian Information Criteria) to determine the number of latent classes (Schwarz, 1978). In *confirmatory* applications where the number of classes is determined by theory<sup>2</sup>, it is common to find that the number of classes recommended by BIC exceeds the number dictated by the original theory.

<sup>2</sup> While this is a confirmatory example, both the traditional LC and LCT approaches are also appropriate in *exploratory* applications, where statistics may be utilized instead of theory to determine the number of classes to use in the initial split (see e.g., van den Bergh et. al., 2017).

Figure 2. *The relative log-likelihood results for the class extractions.*



Formula for calculating change in the loglikelihood<sup>3</sup>:

$$RLL = \frac{(LL(J) - LL(J - 1))}{(LL(2) - LL(1))}$$

Based on the results above, it can be concluded that a 3-class solution provides the most basic set of classes upon which the relationship of classes can be determined. The three-parent level class can be described as (1) more off-peak trains, (2) integrated fares/integrated ticketing, and (3) trains use green power. The next step is to specify a model with three parent level classes and use that core set of classes and test whether additional pairs of classes can be extracted to further refine the model (see van den Bergh et al., in press and [Magidson & Madura, 2018](#), slide 25, for more detailed explanations).

### Step 1: Perform a latent class tree analysis

The first is to add tree commands to the LC choice model syntax. The current implementation of LCT in Latent GOLD utilizes the new keyword “tree” in the Syntax module to initiate the tree analysis process. The most straightforward approach to estimate a LCT model is to start by using the menu-driven graphical user interface (GUI), as we did above, to generate an initial set of syntax commands. This can be done as follows:

- Right click on “3 class model” (which you renamed earlier in this tutorial).
- Select *Generate Syntax* (to obtain the syntax for this 3-class model – see Figure 3).

<sup>3</sup> A more detailed explanation on using the change in loglikelihood for determining the appropriate number of theme classes can be found in van den Bergh, Schmittmann, and Vermunt (2017). For a more detailed explanation on using the change in loglikelihood for the Sydney Transport Inquiry study can be found in Magidson and Madura (Sawtooth, 2018).

Figure 3. Latent class choice tree model syntax.

```

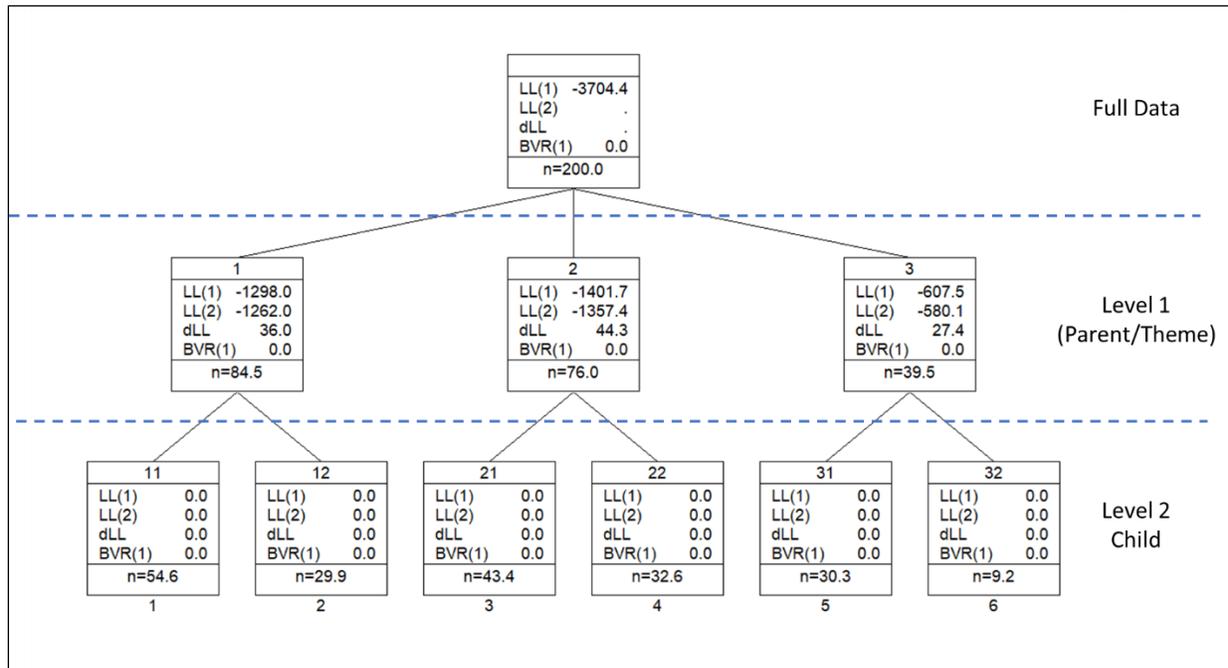
options
maxthreads=6;
algorithm
  tolerance=1e-008 emtolerance=0.01 emiterations=250 niterations=50 ;
startvalues
  seed=0 sets=60 tolerance=1e-005 iterations=50;
bayes
  categorical=1 variances=1 latent=1 poisson=1;
montecarlo
  seed=93841 sets=0 replicates=500 tolerance=1e-008;
quadrature nodes=10;
missing excludeall;
output
  parameters=effect betaopts=wl standarderrors profile probmeans=posterior
  bivariateresiduals classification estimatedvalues predictionstatistics
  setprofile setprobmeans; tree maxlevels=2;
choice = 3
alternatives 'SydneyTransport_alt.sav'
id=AltID
choicesets 'C:\SydneyTransport_set_allTrip8Quad.sav'
id=set;
variables
  caseid ResponseID;
  choicesetid Set ;
  dependent choice bestworst;
  attribute Alt nominal;
  latent

  Class nominal 3 tree;
equations
  Class <- 1;
  choice <- Alt | Class;
  
```

Users have the option to let the iterative class splitting process continue until the termination criteria is met.<sup>4</sup> In this example, the syntax “maxlevels = 2” is used. This command restricts the class extraction process to three levels (or two split levels beneath the ‘root level’).

<sup>4</sup> The approach described above is the simplest way to estimate a latent class tree model. In approach described above, a LC Tree is developed that attempts to perform binary splits of each class using the BIC criteria to determine whether the log-likelihood difference (dLL) for the 2-class model represents a significant improvement over the 1-class model. The splitting process proceeds until this criterion is no longer met.

Figure 4. Latent class tree model results with a three-class parent level.



Since the goal of the tree analysis is identify a “manageable” set of terminal classes, this restriction makes some substantive sense. By selecting “maxlevels = 2” with three theme (or basic level classes) the splitting is restricted to a total of six terminal classes (see Figure 4). In contrast, if “maxlevels = 3” was selected, then twelve classes would result (each of the 6 terminal classes shown in Fig. 4 would split into 2 subclasses)– far exceeding the eight classes that emerged from the “best” standard latent class solution.

Figure 5. *Parameters for the parent- (or theme-) level classes.*

		term	coef	Wald(0)	df	p-value	Wald(=)	df	p-value
Class(1)	←	1	0.2871	7.9599	2	0.019			
Class(2)	←	1	0.1811						
Class(3)	←	1	-0.4682						
choice	←	Alt(More frequent off peak trains between major centres)	Class(1) 0.8782	1202.4832	24	7.3e-239	587.8389	16	8.7e-115
choice	←	Alt(Improved peak rail capacity)	Class(1) 2.2183						
choice	←	Alt(More frequent bus services on major routes)	Class(1) 1.6289						
choice	←	Alt(Extensions of light rail services)	Class(1) -0.7518						
choice	←	Alt(Integrated fares)	Class(1) -0.2763						
choice	←	Alt(Integrated ticketing)	Class(1) -0.4371						
choice	←	Alt(Real-time arrival information)	Class(1) -0.5596						
choice	←	Alt(New cycleways; more bike and scooter parking)	Class(1) -1.3841						
choice	←	Alt(Trains use green power)	Class(1) -1.3165						
choice	←	Alt(More frequent off peak trains between major centres)	Class(2) -0.0750						
choice	←	Alt(Improved peak rail capacity)	Class(2) 1.0328						
choice	←	Alt(More frequent bus services on major routes)	Class(2) 0.8891						
choice	←	Alt(Extensions of light rail services)	Class(2) -0.5958						
choice	←	Alt(Integrated fares)	Class(2) -0.2759						
choice	←	Alt(Integrated ticketing)	Class(2) -0.3892						
choice	←	Alt(Real-time arrival information)	Class(2) -1.0702						
choice	←	Alt(New cycleways; more bike and scooter parking)	Class(2) -0.3714						
choice	←	Alt(Trains use green power)	Class(2) 0.8555						
choice	←	Alt(More frequent off peak trains between major centres)	Class(3) -0.0180						
choice	←	Alt(Improved peak rail capacity)	Class(3) 1.0382						
choice	←	Alt(More frequent bus services on major routes)	Class(3) 0.4410						
choice	←	Alt(Extensions of light rail services)	Class(3) -1.1548						
choice	←	Alt(Integrated fares)	Class(3) 1.3013						
choice	←	Alt(Integrated ticketing)	Class(3) 1.6324						
choice	←	Alt(Real-time arrival information)	Class(3) -0.2876						
choice	←	Alt(New cycleways; more bike and scooter parking)	Class(3) -1.9336						
choice	←	Alt(Trains use green power)	Class(3) -1.0189						

In this model specification, the three parent classes are split. In addition to meeting statistical criteria for splits, the split should also meet substantive criteria in order to justify the split. The parameter output in Figure 5 and Figure 6 illustrates two key concepts that are central to evaluating tree models. The first concept, illustrated by Figure 5, is that theme classes (and subordinate classes that split further down the model) if they are defined by value differences on choice options. In this example:

- Theme class #1 values having “more frequent off-peak trains between major routes;”
- Theme class #2 values “trains using green power;” and
- Theme class #3 values “integrated fares and ticketing

The differences on these choices in the experiment provide substantive evidence as to why a split occurred. The second concept, illustrated by Figure 6, is that subordinate classes (classes further “down the tree”) are invariant on the values that define the theme class they are connected to/belong to. In Figure 2, it is clear that respondents in the child or subordinate classes of theme class #3 (identified by the node identifiers “31” and “32”) agree on the value “integrated fares and ticketing,” (highlighted in black in Figure 6; values are 1.35 and 1.65 for class 31; and 1.71 and 2.17 for class 32). However, they differ on the value of “extensions of light rail services.” More specifically, class 32 values “extensions of light rail services” (value is 0.99) but class 31 does not (values is -1.84).

Figure 6. Parameters for the split of parent class #3 (integrated fares/integrated).

term		coef	Wald(0)	df	p-value
Class(1)	← 1		0.9489		
Class(2)	← 1		-0.2050		
choice	← Alt(More frequent off peak trains between major centres)	Class(1)	0.1057		
choice	← Alt(Improved peak rail capacity)	Class(1)	1.2358		
choice	← Alt(More frequent bus services on major routes)	Class(1)	0.6111		
choice	← Alt(Extensions of light rail services)	Class(1)	-1.8447		
choice	← Alt(Integrated fares)	Class(1)	1.3530		
choice	← Alt(Integrated ticketing)	Class(1)	1.6546		
choice	← Alt(Real-time arrival information)	Class(1)	-0.4429		
choice	← Alt(New cycleways; more bike and scooter parking)	Class(1)	-1.9773		
choice	← Alt(Trains use green power)	Class(1)	-0.6952		
choice	← Alt(More frequent off peak trains between major centres)	Class(2)	-0.1385		
choice	← Alt(Improved peak rail capacity)	Class(2)	0.9644		
choice	← Alt(More frequent bus services on major routes)	Class(2)	0.3260		
choice	← Alt(Extensions of light rail services)	Class(2)	0.9997		
choice	← Alt(Integrated fares)	Class(2)	1.7127		
choice	← Alt(Integrated ticketing)	Class(2)	2.1681		
choice	← Alt(Real-time arrival information)	Class(2)	0.4779		
choice	← Alt(New cycleways; more bike and scooter parking)	Class(2)	-3.2984		
choice	← Alt(Trains use green power)	Class(2)	-3.2119		

Again, observing consistency (symmetry or invariance) on the theme classes through splits and differences on choices where classes split provides key information to inform users about the quality (or substantive sense) of the model.<sup>5</sup>

## Step 2: Examine the LCT output

After the model is estimated, tabular and graphical output is available to view. For example,

- Click on Iteration Detail

In addition to the usual Iteration Detail documenting the iterations performed during estimation of the 3-class model, this output also contains summary results for the LCT analysis, indicating the results of tests for splitting these classes (see Figure 7).

<sup>5</sup> To examine a case where symmetry is not preserved and the classes are “pruned” appears in the Sawtooth 2018 Presentation, slide 36 through 41.

Figure 7. Iteration Detail output from the depression data latent class tree model.

Newton	1	-3481.3799078512		4838.8775711024	0.00003025	
Newton	2	-3481.3799078512		4838.8775711024	0.00000000	
Standard errors						
LC Tree: minN = 0.00, minBVR = 0.00, minLLdiff = 23.84						
Class	N	BVR1	LL1	LL2	LLdiff	split?
1	84.50	0.00	-1298.02	-1262.04	35.98	class is split
2	75.97	0.00	-1401.70	-1357.43	44.27	class is split
3	39.53	0.00	-607.51	-580.14	27.38	class is split
11	54.62	0.00	0.00	0.00	0.00	max. # of levels
12	29.88	0.00	0.00	0.00	0.00	max. # of levels
21	43.36	0.00	0.00	0.00	0.00	max. # of levels
22	32.61	0.00	0.00	0.00	0.00	max. # of levels
31	30.31	0.00	0.00	0.00	0.00	max. # of levels
32	9.22	0.00	0.00	0.00	0.00	max. # of levels
Total number of nodes/classes: 9						
Number of splits (incl. root): 4						
Number of end nodes/classes: 6						
Preparing output						

The first line of the LC Tree section of this output begins by documenting the split criteria. Based on the BIC, the critical value for the LL difference is computed by Latent GOLD to be 23.84.

## References

Magidson, J. & Vermunt, J. K. (2001). Latent class factor and cluster models, bi-plots, and related graphical displays. *Sociological Methodology, 31*, 223-264.

Pearlin, L. I., & Johnson, J. S. (1977). Marital status, life-strains, and depression. *American Sociological Review, 42*, 104-115.

Schaeffer, N. C. (1988). An application of item response theory to the measurement of depression. In *Sociological Methodology 1988*, C. Clogg (Ed.). Washington: American Sociological Society.

Schwarz, G. (1978). *Estimating the dimension of a model*. The Annals of Statistics, 6(2), 461-464.

van den Bergh, M., Kollenburg, G. H., & Vermunt, J. K. (under review). Deciding on the starting number of classes of a latent class tree.

van den Bergh, M., Schmittman, V. D., & Vermunt, J. K. (2017). Building latent class trees, with

an application to a study of social capital. *Methodology: European Journal of Research Methods for Behavioral and Social Sciences*, 13, 13-22.