

## Statistical Innovations Online course:

# Latent Class Discrete Choice Modeling with Scale Factors and Tree Models

### Session 2:

#### Putting it All Together – Scale Adjusted Latent Class (SALC) Models in Practice

##### Outline:

- A. Individual-Level Coefficients
- B. General Issues in SALC Modeling
- C. Models for MaxDiff Data
- D. Analyzing MaxDiff Data with Scale Factors
- E. Latent Class Tree Models in Discrete Choice Experiments

## A. Individual-level Coefficients

The standard LC Choice model allows different part-worth parameters to be estimated for each latent class  $k=1,2,\dots,K$ . At the individual respondent level, the posterior membership probabilities associated with a given estimated model assigns respondent  $i$  to latent class  $k$  with probability  $P_{k,i}$ . As such, it is possible to compute individual-specific part-worth utilities for the  $i^{\text{th}}$  respondent as a weighted average of the class-specific part-worths, the weights for respondent  $i$  corresponding to their posterior membership probabilities. For more details, see Equation 20 on (printed) page 58 of the [LG Choice User's Guide](#).

For concreteness, let us return to [LG Choice Tutorial 10A](#), where choice data were generated for each of  $K=3$  latent classes, with the population part-worth utility values for each segment shown in Table 1 (see below). These values are *scale-adjusted* in the sense that they are associated with respondents having scale factor = 1.

Recall that for this application there are 2 nominal attributes, each having 3 levels, and notice that the part-worths are effect coded, since that they sum to 0 for each attribute; the sum of the part-worths is  $(1)+(0)+(-1) = 0$  for LC Segment 3 for attribute 1, and  $(2)+(0)+(-2) = 0$  for attribute 2.

	Segment 1	Segment 2	Segment 3
Attribute 1, Level 1 (b1)	-1	0	1
Attribute 1, Level 2 (b2)	0	-3	0
Attribute 1, Level 3 (b3)	1	3	-1
Attribute 2, Level 1 (b4)	2	-1.5	2
Attribute 2, Level 2 (b5)	0	0	0
Attribute 2, Level 3 (b6)	-2	1.5	-2

Table 1: Scale-adjusted part-worth utilities (Scale Factor =1)

In [LG Choice Tutorial 10A](#) we estimated several models for these data, Models 1-6 being standard (unadjusted) LC choice models (with 1-6 latent preference classes), and Models 7-12 being SALC models (with 1-6 latent preference classes). Individual-level coefficients for any of these models could be obtained using LG Choice by checking the box ‘Individual Coefficients’ in the ClassPred tab prior to estimating the model.

Figure 1 below shows a plot of two individual coefficients based on Model 4 (the unadjusted 4-class model). The individual coefficients plotted are the utility pair (b1, b4), b1 being associated with attribute 1 and b4 associated with attribute 4. **Note that Figure 1 plots the coefficients only for those N=200 respondents that truly belong to population Segment 3.** The true population coefficients for these respondents are highlighted in column 3 of Table 1.

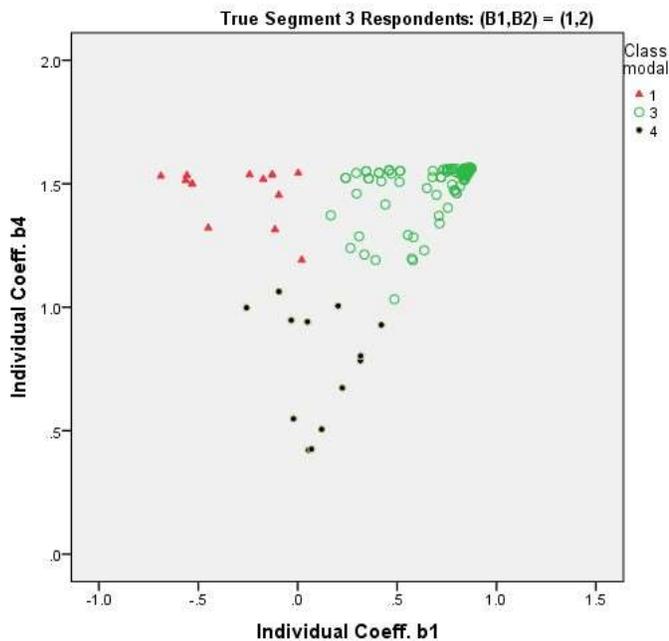


Figure 1. For a true homogeneous population (Segment 3) with preference part-worth utilities B1=1 and B4=2, the standard solution (“Unadjusted 4-class model”).

Note that the model on which Figure 1 is based makes the faulty assumption that the scale factor is identical for all respondents. The result of this faulty assumption is that classes differ not only with respect to preferences but also scale, and use of the BIC to results in too many classes (4 rather than 3 in this example). Thus, the resulting classes confound differences in preference with differences in scale. For more insight see material in Session 1.



## Exercise A1

Use SPSS to reproduce the plot shown in Figure 1.

Hint: You can re-estimate 'Model 4' from Tutorial 10A after using the ClassPred tab to request that the Individual Coefficients as well as Posterior Classification information be output to a SPSS (.sav) file. Please also include the variable 'sclss' (which contains the true population scale class) on the .sav outfile by moving this variable to the 'keep' box in the ClassPred tab. Be sure that you only plot the respondents for which the true population segment is 3.

In case you don't have access to SPSS you can use the Model4out.xls to produce the plot.

**Question A1.1: Which of the classes (the ones plotted in red, green and/or black) are correctly classified?**

**Question A1.2: Would you say that these respondents who are correctly classified tend to be *more* or *less* consistent in their responses than the respondents who are misclassified?** Hint: You can use the variable 'sclss' to check your answer.

**Question A1.3: How many of the 200 respondents that truly belong to segment 3 are misclassified?** Hint: Use the file you created during this exercise.

**Question A1.4: In Figure 1, why do many green points appear at  $(b_1, b_2) = (.87, 1.56)$  rather than at the true population value  $(b_1, b_2) = (1, 2)$ ?**

## B. General Issues in SALC Modeling

*“Essentially, all models are wrong, but some are useful”*

George Box

While it is widely accepted in choice modeling that standard choice models that account for heterogeneity confound preference and scale heterogeneity, there is no agreement as to how to model scale in such models. For example, Hess and Rose (2012) criticize the G-MNL approach:

*“this paper argues that attempts in the literature to disentangle scale heterogeneity from heterogeneity in individual coefficients in discrete choice models are misguided”*

showing how such scale heterogeneity in such models can be interpreted as preference heterogeneity using an alternative model specification. We agree that you never can be sure whether the heterogeneity that a model captures is heterogeneity in scale or heterogeneity in preference.

Our approach differs from G-MNL in that our scale classes are highly structured, meaning that our assumption that they pick up primarily scale difference is quite likely to hold in practice. For example, Figure 2 shows that respondents denoted by the black symbol (class 4) differ from those denoted by the green symbol (class 3) only in scale, the former showing up as being further out on a straight line.

Now that we have gained some experience estimating some SALC models, in this section we will summarize ways that SALC models can differ. All SALC models relax the assumption that the scale factor is homogeneous (i.e., constant for all respondents). However, some SALC models may account for observed scale heterogeneity by including observed predictors in the log-scale model. Other SALC models account for *unobserved* (latent) scale heterogeneity by including discrete or continuous latent variables in the log-scale model, and hybrid SALC models account for both observed and unobserved (latent) scale heterogeneity.

- 1) **Observed Scale Heterogeneity.** This occurs when a respondent’s scale factor can be expressed as a function of one or more observed predictor variables. For example, with the TV data example in Session 1 the log-scale factor was expressed as a linear function of the observed completion time variable **TimeR**.
- 2) **Latent Discrete Scale Heterogeneity.** When respondents’ scale factors differ in unknown ways, the log-scale factor can be modeled using a nominal latent variable with two or more ‘scale classes’ (sClasses) using the LG Choice point-and-click interface. This type of model accounts for latent *discrete* scale heterogeneity, and was illustrated in Session 1 on pages 1-18 (through Fig. 21) of [LG Choice Tutorial 10A](#) with data simulated from a population consisting of two

latent scale classes – one sClass responding more consistently than the other across the choice sets.

- 3) **Latent Continuous Scale Heterogeneity.** While SALC models accounting for latent heterogeneity using a *discrete* latent variable assume that the population contains only a small number of scale factors (one for each latent scale class), SALC models may instead account for latent scale heterogeneity using a *continuous* latent variable, which assumes that each population respondent has their own unique scale factor. This type of model was illustrated on pages 18 - 19 of [LG Choice Tutorial 10A](#).
- 4) **Both Latent and Observed Scale Heterogeneity.** The general log-scale model can contain both a latent variable (e.g., sClasses) and observed variables as predictors. This type of SALC model can be estimated with the point-and click interface of LG Choice 5.0 (see Session 1). In addition, more general SALC models can be estimated with the Syntax version of the program (see Session 1).

For SALC models that account for latent scale heterogeneity, the latent variable used to model scale heterogeneity can be assumed to be independent of the latent preference classes (the default in LG Choice) or be related to the latent preference classes. In addition, scale factors associated with each sClass can be allowed to differ for the preference classes. The independence form of the class/sClass relationship is specified automatically by the point-and-click interface, while the more general model forms require the Syntax version of LG Choice. [LG Choice Tutorial 10A](#) illustrated how to estimate these kinds of SALC model forms:

- a. **Independence between Latent Scale Factor and Preference Classes.**  
Pages 1-17 of [LG Choice Tutorial 10A](#) showed how to estimate a SALC model where the two scale classes were independent of the three latent preference classes (segments). Independence here means that each segment has the same percentage of respondents who belong to the more consistent (less consistent) scale class.
- b. **Association between Latent Scale Factor and Latent Preference Classes.**  
Pages 17-18 of [LG Choice Tutorial 10A](#) show how to relax the independence assumption to allow some segments to have a higher (or lower) percentage of respondents belonging to the more consistent sClass.
- c. **Differential Scale Effects for Latent Preference Classes.** Pages 17-18 of [LG Choice Tutorial 10A](#) also shows how to use to update the scale model equation to allow the scale factor associated with a given sClass to differ for each latent preference class.



## Exercise B1

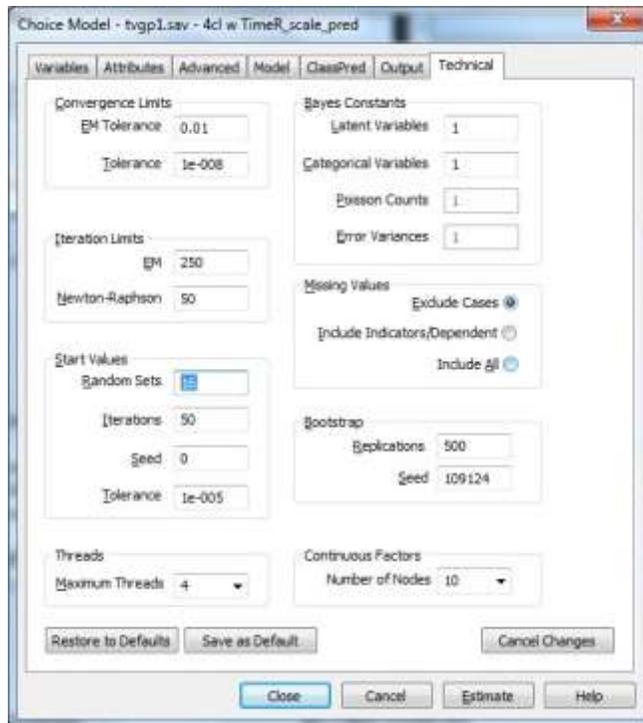
Using the TV data example from Session 1, modify the SALC models estimated earlier in Session 1 Exercise B1 so that they include scale classes. To do this, you can proceed as follows:

Re-open ‘TV\_4cl.lgf’ and re-estimate the three saved models:

- 1) ‘**4-cl w/o cov**’: This is a traditional 4-class choice model without any active covariates.
- 2) ‘**4-cl w TimeR\_ cov**’: This is the 4-class choice model with the active covariate ‘**TimeR**’.
- 3) ‘**4-cl w TimeR\_ scale\_pred**’: This is the same model as the traditional choice model in 1), except that a scale model is also estimated with ‘**TimeR**’ included as a numeric predictor.

Now double click on the SALC Model ‘**4-cl w TimeR\_ scale\_pred**’ to re-open the Variables tab, click on ‘**Advanced**’ to open the Advanced tab, **and set sClasses to ‘2’ in the section labeled ‘Scale Classes’**.

Since we formulated this hybrid SALC model by modifying a previously saved model, the best seed for estimating that model was saved. Before estimating this model we need to restore the default settings to avoid getting a local solution. To do this, click on ‘**Technical**’ to open the Technical tab and in the Start Values section change the Seed to ‘0’ and change the Random Sets from ‘0’ to ‘16’ to allow multiple starting values. Following these changes, the Technical tab should look like this:



Now estimate this model.

**Question B1.1: Which of the 4 estimated models fits best?**

**Question B1.2 For the newly estimated SALC model:**

- Which of the 2 sClasses is associated with the higher scale factor?
- What is the p-value associated with the effect of the sClasses in the scale model?
- Which of the 2 sClasses is larger?



## Exercise B2

Perform the following:

To generate the syntax for the new SALC model estimated in Exercise B1:

- Right-click on the new SALC model
- Select '**Generate Syntax**' from the menu

Now, modify this model to allow the effect of '**TimeR**' in the scale model to depend on class:

- Scroll down to the '**Equations section**' of this model and replace the line

Choice <<- TimeR + (-) sClass ;

with

Choice <<- (s) TimeR |class + (-) sClass;

That is, insert '(s)' to name the scale model effect parameters 's', and add the conditional operator '|class' to allow separate estimates for each class.

Finally, we will again make the random responder class the reference class for the scale factors:

- At the bottom of the syntax file, add the line '**s[4,1]=0;**' to set the scale effect for the random responder class 4 to 0.
- Estimate this new model.

Compare the BIC of this model to the others.

**Question B2.1: Does this updated model improve the fit to the data?**

**Question B2.2: Which scale model effect is more important – the 'TimeR | class' effect or the 'sClass' effect?**

Hint: Examine the parameters output to see which is more significant.

## C. Analyzing MaxDiff Data

Rather than just obtaining one choice (the most preferred) for each choice set, so-called “MaxDiff” designs elicit two choices for each set -- the worst (least preferred) as well as best (most preferred) options. MaxDiff data are thus a kind of “partial ranking” data, which fall short of eliciting a complete ranking of all the alternatives posed in a choice set but provide more information than choosing the most preferred option only. By eliciting both the best and worst, MaxDiff data are considered the “gold standard” for estimating preference in an efficient manner.



### Reading Material

Assigned Reading Material 2:

Pages 1-10 of [LG Choice Tutorial 8A](#).



### Exercise C1

Open ‘**SydneyTransport (3-file).lgf**’ and estimate the second model (the 8-class model) and answer the following questions based on the obtained output.

**Question C1.1: From the 8-class model, which classes have “Improved Peak Rail Capacity” as their highest priority?**

**Question C1.2: What percentage of the respondents are predicted by this model to have “Improved Peak Rail Capacity” as their highest priority?**

Many users familiar with MaxDiff designs and MaxDiff data may be surprised to learn that there are different models that have been proposed to analyze this type of data. Models used primarily by marketing researchers and economists assume one of two different behavioral processes for analyzing this type of data. Within each process there are also two alternative specifications, giving FOUR primary models (1a, 1b, 2a, and 2b) for analyzing MaxDiff data:

1. THE SEQUENTIAL BEST/WORST MODELS. Under this process it is usually assumed that respondents (a) initially choose their Best option from the available K alternatives and then choose their Worst option from the remaining K-1 alternatives.

Alternatively, respondents may (b) initially choose their Worst option followed by their Best option. These alternative specifications – BestWorst (BW) and WorstBest (WB) - - may yield different predictions and needs segments; Since the option selected best (worst) is sequentially removed from the choice set, the possibility of selecting the same alternative as both Best and Worst is excluded.

2. THE MAXDIFF MODELS. Unlike the sequential approach, the MaxDiff model is symmetric in that the utilities do not depend upon the order of selection. Two versions of the MaxDiff model are most frequently used:
  - a. the MaxDiff Quasi-Independence (MD-QI) model assumes that a respondent evaluates all possible (Best, Worst) pairs of options within the displayed set and chooses that pair that reflects the maximum difference in preference or importance. This model explicitly excludes the possibility that an option is selected as both Best and Worst.
  - b. The MaxDiff Independence (MD-I) model assumes best and worst options are evaluated independently (not simultaneously) and “relaxes the constraint that the most and least preferred options have to be different” (Bacon, et. al., 2007).

In this Session we focused on the Sequential Best/Worst model (“BestWorst”) where respondents are assumed to first select the Best and then select the Worst option from the remaining options. This model was implemented in earlier versions of LG Choice (along with the “WorstBest”), but Version 5.0 simplifies the setup of this model using the new ‘BestWorst’ scale type. In Session 3 of this course, we will show how to estimate the MaxDiff Models, which has been newly implemented in version 5.0 of LG Choice, as part of the syntax module.

## D. Analyzing MaxDiff Data with Scale Factors

The BestWorst models can also be estimated with scale factors. This is described on pages 11-16 of [LG Choice Tutorial 8A](#).



### Reading Material

Assigned Reading Material 2:

Pages 11-16 of [LG Choice Tutorial 8A](#). Analyzing MaxDiff Data with Scale Factors: Applications of Latent GOLD Choice 5.0.



### Exercise D1

We continue with the example from Exercise C1.

Estimate the third model from the example ('8-class+scale\_worst') to allow separate scale factors for the best and worst choices.

**Question D1.1: Does this model improve model fit compared to the 8-class model without the scale factors?**

**Question D1.2: Is the Worst response associated with a higher or lower scale factor than Best response?**

Now also estimate the fourth model that also includes the main effects for two scale classes.

**Question D1.3: Does including these scale classes improve the fit of the model?**

**Question D1.4: Is the Worst response associated with a higher or lower scale factor than Best response?**

**Question D1.5: Which scale model effect is more important – the ‘Worst’ effect or the ‘sClass’ effect?**

Based on the output from the model summary tab, the hit rate of a model can be computed in two ways:

- 1) Compute 1 minus the prediction error
- 2) Divide the sum of the off the diagonals of the prediction table by the overall total

**Question D1.6: Does including scale effects improve the hit rate of the Best and Worst response?**

### E. Latent Class Tree Models in Discrete Choice Experiments

A recurring problem in standard exploratory latent class analysis is issue of class extraction. Likelihood-based criteria such as the Bayesian Information Criterion (BIC) or the Akaike Information Criterion (AIC) often result in a large number of classes that are difficult to interpret. An emerging modeling approach that addresses this problem is latent class trees (van den Bergh, Schmittmann, & Vermunt, 2017; van den Bergh, van Kollenburg, & Vermunt, in press).

In latent class tree (LCT) modeling, a hierarchical structure – similar to hierarchical cluster analysis – is assumed for the latent classes. In current applications of this technique, the hierarchical structuring results from divisive procedures (procedures that split classes) that are based on predefined criteria such as the change in loglikelihood. This procedure, based on an algorithm developed by van der Palm, van der Ark, and Vermunt (2016) for density estimation, starts with a parent node consisting of the entire dataset and involves estimating 1- and 2-class model for the subsample at each node in the tree. If a 2-class model is preferred, based on fit criteria, the node is split and two additional nodes are created. This procedure is repeated at the next level of the hierarchy until no additional splits are warranted. This procedure can be done manually within Latent GOLD®, however, the latent version of the software has the ability to automate this process.

Figure 1. Annotated latent class tree output from LatentGOLD® 5.1 for the Sydney Transport Inquiry data (three theme classes and dLL split criteria).

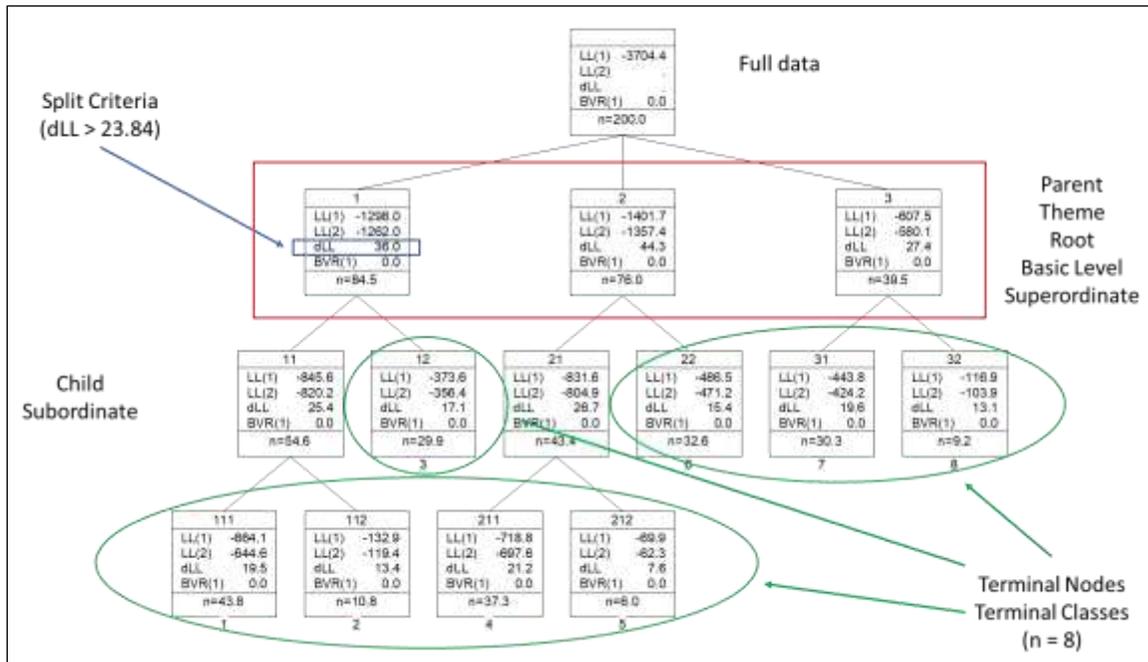


Figure 1 provide a graphical summary the results for the same Sydney Transport Inquiry data used in Tutorial 10A. The simplest way to specify a latent class tree model is to first generate a standard latent class model using the GUI. Users identify the number of theme classes for use in the tree model (this case three) and then copy of the model syntax (right clicking and then *copy model*). A new model will appear in the model summary section on the left. Figure 2 is a snapshot of the proper standard model syntax.

Figure 2. *Annotated syntax for a standard three-class latent class model for use in the tree model specification.*

```

options
  maxthreads=6;
  Algorithm
    tolerance=1e-008 smtolerance=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=ul standarderrors profile probmeans=posterior
    bivariateresiduals classification estimatedvalues predictionstatistic
    setprofile setprobmeans;
choice = 3
  alternatives 'SydneyTransport_alt.sav'
  id=AltID
  choicesets 'C:\Users\SydneyTransport_set_allTrip@Quad.sav'
  id=sets;
variables
  caseid ResponseID;
  choicestid Set ;
  dependent choice bestworst;
  attribute Alt nominal;
  latent
  Class nominal 3;
equations
  Class <- 1;
  choice <- Alt | Class;

```

The most important step at this stage is identifying the appropriate number of theme class for construction of the tree model. Theme class represent a simpler subset of classes upon which additional classes are constructed. In this way, additional subordinate classes represent refinements of the basic set of class. Theme level class can be identified from theory or they can be constructed in an exploratory fashion using fit criteria. How this is done is explained in greater detail in Tutorial 8C: Analyzing MaxDiff Data with Latent Class Trees (3-file). Figure 3 is an example of the proper syntax for estimation of a tree model with three theme classes.

Figure 3. Annotated syntax for a latent class tree model.

```

options
maxthreads=6;
algorithm
tolerance=1e-006 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-006;
quadrature nodes=10;
missing excludeall;
output
parameters=effect betaopts=wl standarderrors profile probmeans=posterior
bivariatereiduals classification estimatedvalues predictionstatistics
setprofile setprobmeans tree;
choice = 3
alternatives 'SydneyTransport_alt.sav'
id=AltID
choicesets 'C:\Users\SydneyTransport_set_allTrip@Quad.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;

```

Once the model is estimated, users can review the *Iteration Detail* to examine the results of every split attempt. In addition, users can select the *Parameter* tab to examine the utilities for every class within the tree – whether it is a terminal class or not. Results for the *Set Profile* are available for all classes as well. Additional material on

It is important to note that currently there is no established statistical approach for evaluating the overall fit of latent class tree model. Users can only evaluate each individual split. Instead, using the results from the standard latent class model, users can qualitatively evaluate results from the tree model and compare them to the standard latent class model results. At this stage, users can decide whether the tree results represent an improved set of meaningful classes over the standard latent class results, or not.



## Reading Material

Assigned Reading Material:

(Post-) Materialism discrete choice model example in van den Bergh et al. (in press)  
Tutorial 8C: Analyzing MaxDiff Data with Latent Class Tree (3-file)

Optional Reading:

SALC Tree Models ([Sawtooth Conference Presentation, 2018](#))



## Exercise E1

**Question E1.1: Read Tutorial 8C: Analyzing MaxDiff Data with Latent Class Tree (3-file). Can you reproduce the tree model in the tutorial?**

**Question E1.2: If you are interested in seeing a case where techniques of scale adjusted latent class models are combined with those for latent class tree modeling, read the presentation of SALC Trees ([SALC Tree Model Presentation](#)).**

## References

Bacon, Lenk, Seryakova, and Veccia (2007). Making MaxDiff more informative: Statistical data fusion by way of latent variable modeling. *Sawtooth Software Conference Proceedings*.

Hess, S., and J.M. Rose (2012). Can scale and coefficient heterogeneity be separated in random coefficients models? *Transportation*. DOI 10.1007/s11116-012-9394-9.

van den Bergh, M., Schmittmann, V.D., and Vermunt, J.K. (2017). Building latent class trees, with an application to social capital. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences*, 13(Suppl. 1), 12-22.

van den Bergh, M., van Kollenburg, G. H., and Vermunt, J. K. (in press). Deciding on the starting number of classes of a latent class tree model. *Sociological Methodology*.