Last updated: 10/30/2014

Latent GOLD® Choice 5.0 tutorial #10B (1-file format)

How to Estimate Scale-Adjusted Latent Class (SALC) Models and Obtain Better Segments with Discrete Choice Data

Introduction and Goal of this tutorial

Researchers frequently use Latent Class (LC) choice models for strategic segmentation and targeting purposes to

1) find meaningful segments of respondents having different preferences, and
2) estimate part-worth utilities for these segments.

However, LC models can form spurious segments that mainly differ in terms of scale (response error) but don't differ much in terms of real preference patterns.

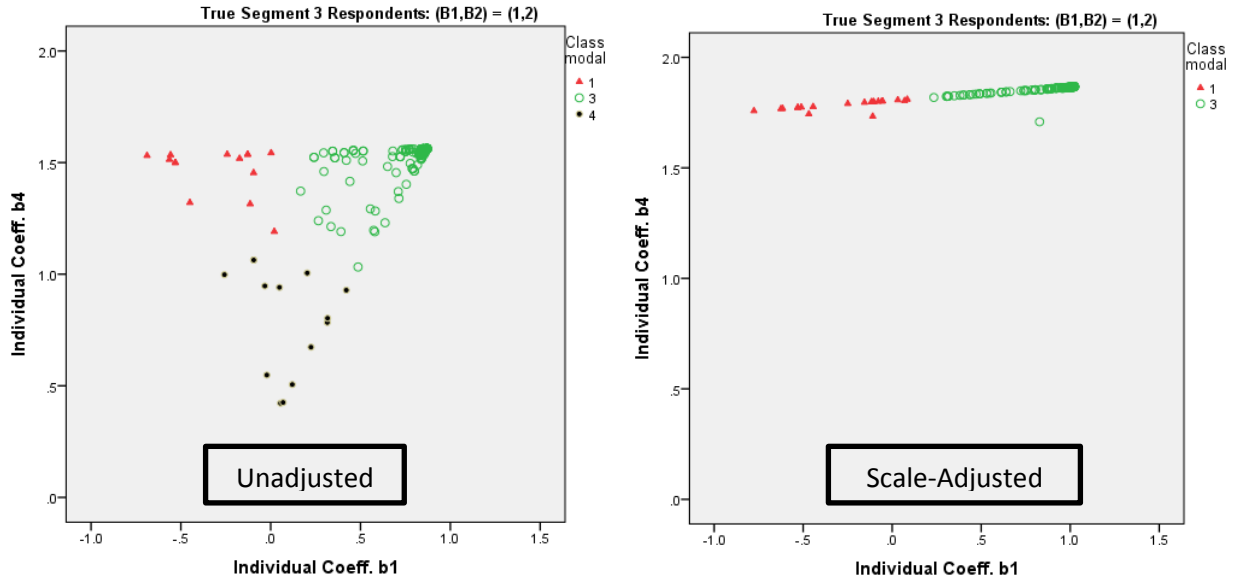


Figure 1. For a true homogeneous population (Segment 3) with preference part-worth utilities B1=1 and B4=2, the standard solution ("Unadjusted") shows much more error variation and greater misclassification than the "Scale-Adjusted" solution.

In this tutorial we illustrate a simple model-fitting strategy that can be used with Latent GOLD® Choice 5.0 Advanced to estimate Scale-Adjusted LC (SALC) models which adjust for differences in scale, resulting in more meaningful segments that differ only in preference.

The approach is illustrated using responses simulated based on 3 homogeneous segments (3 latent classes) that differ according to their preference part-worth utilities and also in their error variance (2 latent scale classes), assumed to be independent of their latent class.

SALC models can be estimated in LG Choice 5.0 with observed and/or unobserved scale heterogeneity (Magidson and Vermunt 2007). The former can be done by allowing the scale factor to depend on observed variables (e.g., time to complete task, stated vs. revealed preference), the latter by including latent scale classes or latent factors in the model. In this tutorial we illustrate the latter situation (unobservable scale heterogeneity).

At the end of this tutorial we show how the syntax module of LG Choice can be used to estimate more complex SALC models where the latent scale classes are correlated with the latent segments.

Background

As explained by Louviere and Eagle (2006), standard discrete choice models suffer from a potential confound between scale factors and utilities -- the larger the response error, the smaller the magnitude of the part-worth utilities; the smaller the response error, the larger the magnitude of the part-worth utilities. As a consequence, two segments of respondents could have essentially the same preferences, but appear to have different utilities due to a stretching or shrinking of the utilities (the scale factor).

SALC models were first introduced by Magidson and Vermunt (2007) and implemented in the LG Choice Syntax version 4.5 in a somewhat complex manner. Since that time, several applications have appeared in the literature (see e.g., Burke et al., 2010). Version 5.0 of LG Choice simplifies the earlier implementation, allowing basic SALC models to be setup and estimated with the point-and-click interface. Moreover, the current implementation allows greater user control via the new log-linear scale model. This model estimates the log-scale factor rather than the scale factor directly, thus guaranteeing that the scale factor is always non-negative.

For further details, see the LG Choice 5.0 Upgrade Manual and references cited there.

For the example used in this tutorial, we assume that there are three latent class segments differing in their part-worth utilities (see Table 1) but some individuals are more uncertain about their preferences than others. For the less certain or high error individuals, we assume a Scale Factor of .5, which means that their utilities are obtained by multiplying the part-worths in Table 1 by 0.5.

Table 1. Part-worth Utilities for Low Error Cases (Reference Scale Factor = 1).

	Reference (Scale Factor = 1)		
	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 2. Part-worth Utilities for High Error Cases (Scale Factor = .5).

	Scale Factor = .5		
	Segment 1	Segment 2	Segment 3
Attribute 1, Level 1	-0.5	0	0.5
Attribute 1, Level 2	0	-1.5	0
Attribute 1, Level 3	0.5	1.5	-0.5
Attribute 2, Level 1	1	-0.75	1
Attribute 2, Level 2	0	0	0
Attribute 2, Level 3	-1	0.75	-1

We begin by simulating data containing both preference heterogeneity (3 latent class segments) and scale heterogeneity. For simplicity, we generated an equal number of low error (scale class 1) and high error (scale class 2) cases (N=300 of each). Thus, the correct model will be one that scale heterogeneity is discrete, and in fact dichotomous. In a later tutorial, we will analyze data generated with continuous scale heterogeneity, each case having a different scale factor.

The simplified SALC approach implemented in release 5.0 of LG Choice consists of 2 steps:

Step 1 – estimate LC models with no scale classes (say, # latent classes = 1-6)
Step 2 – estimate LC models with 2 scale classes (can repeat for scale classes >2 if desired)
Select the model with the lowest BIC

As we will see in this tutorial, Scale-Adjusted LC (SALC) modeling determines the correct number of preference segments (3) here, provides estimates of the utilities (see Table 3) that are close to the assumed values provided in Table 1, and classifies over 92% of the cases into the correct segment[1]. In contrast, ignoring scale differences causes biased parameters and overestimates the number of latent preference segments (resulting in a reduced correct classification rate of 87.9%).

LG Choice 5.0 can be used to estimate SALC models consisting of either discrete or continuous scale heterogeneity. For the current data, both of these models yield very similar results. Latent GOLD® Choice 5.0 tutorial #11 describes similar analyses based on data simulated with continuous scale heterogeneity.

[1] For this sample of N=600 simulated respondents, Latent GOLD's ProbMeans output shows that 92.5% of the cases are classified correctly when using proportional assignment, whereas the modal assignment rule classifies 95.2% of the cases correctly.

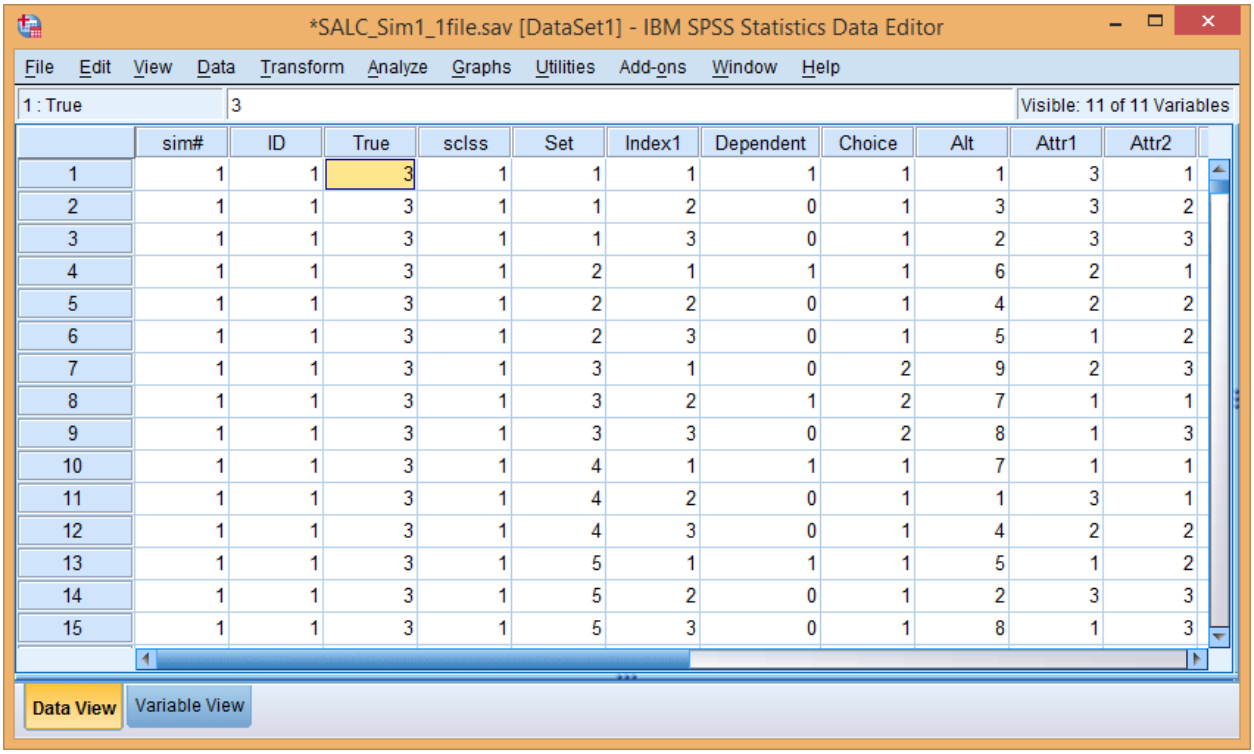
Table 3. Estimated solution with N=600 simulated cases.

	Segment 1	Segment 2	Segment 3
size=	0.36	0.33	0.31
Attr1			
1	-0.8	0.0	1.0
2	-0.1	-2.7	0.0
3	0.9	2.6	-1.0
Attr2			
1	1.8	-1.3	1.9
2	0.0	0.1	-0.1
3	-1.8	1.2	-1.8

Simulated Data

Simulated Response Data: 'SALC_Sim1_1file.sav' (seeFigure 2).

Note: The design consists of 9 alternatives which differ on 2 attributes, each having 3 categories. Using these 9 alternatives, 12 choice sets are created, each consisting of 3 alternatives.



	sim#	ID	True	sclss	Set	Index1	Dependent	Choice	Alt	Attr1	Attr2
1	1	1	3	1	1	1	1	1	1	3	1
2	1	1	3	1	1	2	0	1	3	3	2
3	1	1	3	1	1	3	0	1	2	3	3
4	1	1	3	1	2	1	1	1	6	2	1
5	1	1	3	1	2	2	0	1	4	2	2
6	1	1	3	1	2	3	0	1	5	1	2
7	1	1	3	1	3	1	0	2	9	2	3
8	1	1	3	1	3	2	1	2	7	1	1
9	1	1	3	1	3	3	0	2	8	1	3
10	1	1	3	1	4	1	1	1	7	1	1
11	1	1	3	1	4	2	0	1	1	3	1
12	1	1	3	1	4	3	0	1	4	2	2
13	1	1	3	1	5	1	1	1	5	1	2
14	1	1	3	1	5	2	0	1	2	3	3
15	1	1	3	1	5	3	0	1	8	1	3

Figure 2. SALC Sim1 data file in 1-file format.

This simulation ('sim#'=1), was performed using the parameters shown in Table 1 and Table 2. The response file below contains the responses of 600 individuals ('Choice') for each set, as well as their true segment ('True') and scale class ('sclss').

Estimating Models with LG Choice Point-and-Click Interface

We will begin by opening previously saved SALC models.[2]

In LG Choice:

- Click on File→Open.
- In the Open window, click on the 'Files of type' drop down menu and select 'LatentGOLD files(*.lgf)' (see Figure 3).
- Click on the file 'SALC_Sim1.lgf' and then click 'Open'.

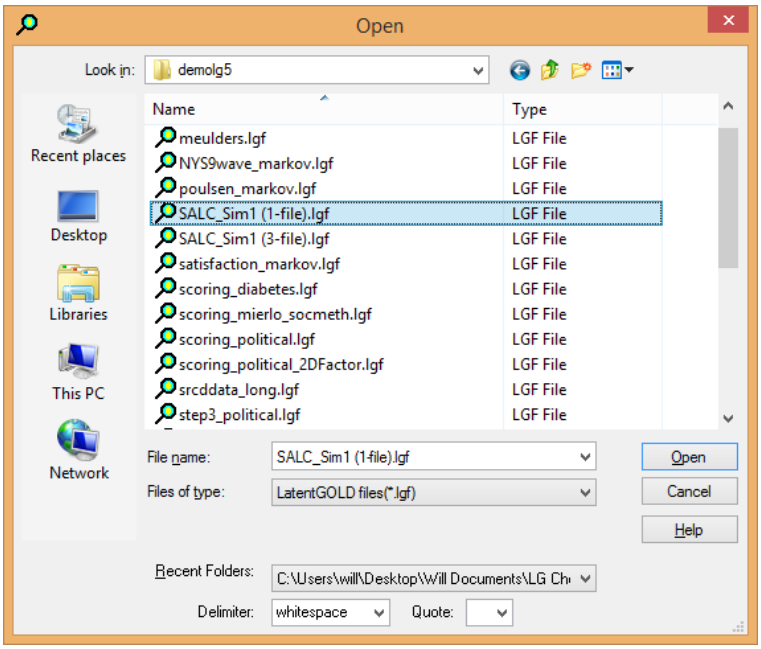


Figure 3. Opening previously saved models.

[2] For a step-by-step tutorial on how to 1) set up a model in LG Choice using the 3-file format, 2) determine the number of classes, and 3) save model definitions in a .lgf file, please see Tutorial 1.

Our 18 previously saved models appear in the left pane (see Figure 4).

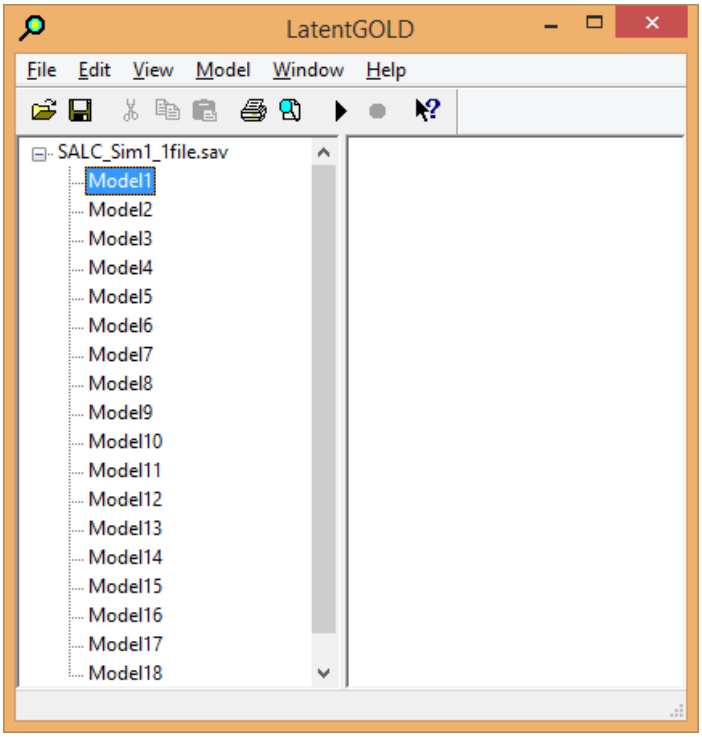


Figure 4. Previously saved models.

- Left click on 'SALC_Sim1_1file.sav' in the Outline pane.
- From the Toolbar, click 'Model' and then click 'Estimate All' (Figure 5).

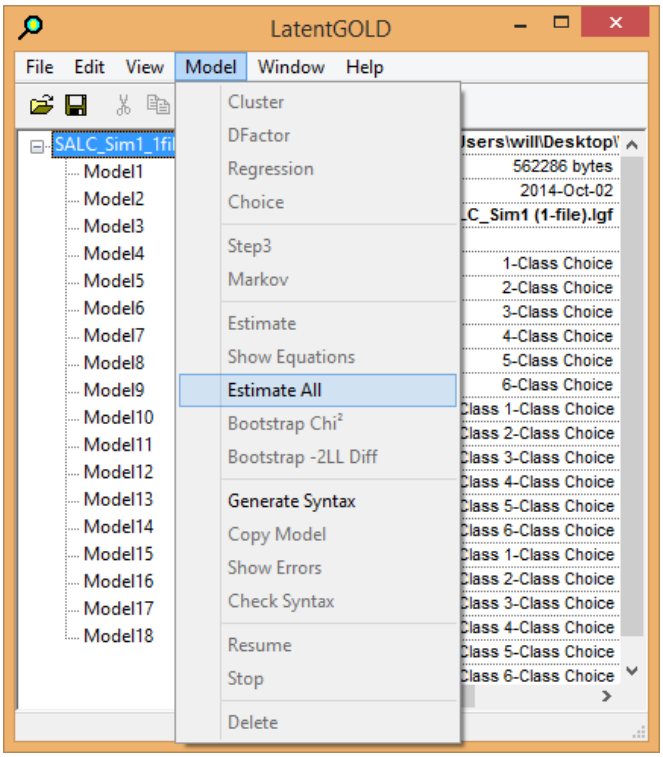


Figure 5. Estimating all saved models.

After estimation has completed, results for all estimated models appear in the Outline pane.

From the model labels (Figure 6) it can be seen that we estimated SALC models with 1-6 classes that have 2 scale classes (Models 7-12), 3 scale classes (Models 13-18), and 6 non-SALC models that are homogeneous with respect to scale (Models 1-6).

- Click on the data file name 'SALC_Sim1_1file.sav' in the Outline (left hand) pane to compare the fit of these models by viewing the Data File Summary output (see Figure 6).

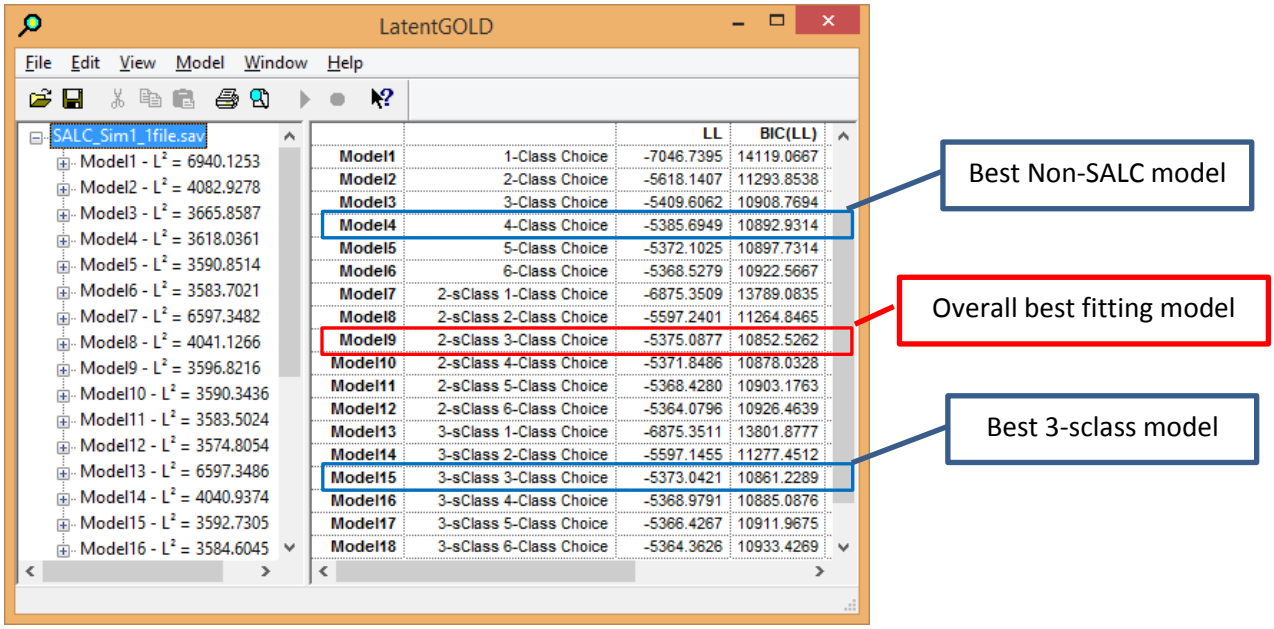


Figure 6. Data File Summary Output.

From this summary, we obtain the following results:

- Overall, the 2-sClass, 3-class SALC model ('Model9') fits best (lowest BIC).
- With 3 (or more) sClasses, BIC still selects the 3-class model as best[3]. Since the correct classification rate and parameter estimates from the 3-Class models with 2 or 3 sClasses are virtually identical, including too many sClasses does not cause a problem.
- Not accounting for scale heterogeneity (non-SALC Models 1-6), results in the 4-class model as best[4] (Model 4). Viewing the output for this model shows that class 4 consists of 7% of cases and the part-worth parameters for this class are all close to zero[5].

[3] For the 3-sClass model, the scale factors are 1, 0.57, and 0.22, and the third sClass (associated with scale factor of 0.22) consists of only 5% of the cases.
[4] If the sample was large enough we would get 6 classes (3 true classes x 2 true sClasses = 6 joint classes).
[5] About 93% of the cases in class 4 are those that gave less consistent responses (i.e., sClass=2 cases).

➢ Double click on Model9 to view the setup for this model (Figure 7).

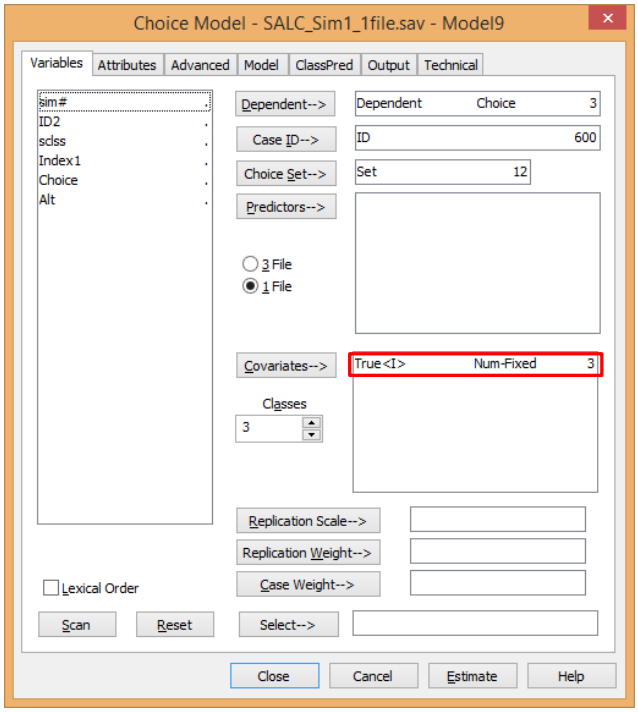


Figure 7. Model Setup for the 2-sClass, 3-class model.

In the Covariates box, the variable 'True' appears as an *inactive* covariate -- the symbol 'I' within the brackets '<I>' denotes *inactive*. 'Inactive' means that information regarding the 'True' segments is not used at all during the parameter estimation process. We include it as an inactive covariate so that we can see (in Figure 9) how well the model recovers the true segments.

Note that this recovery information is possible to obtain in our simulated data example because the true class membership is known. In most real world applications, the true segment membership is unknown. In this case, we can rely on the classification table from which we can obtain the expected correct classification rates under the assumption that our model is correct.

- Click on the Advanced tab to verify that 2 sClasses were specified for this model (Figure 8).

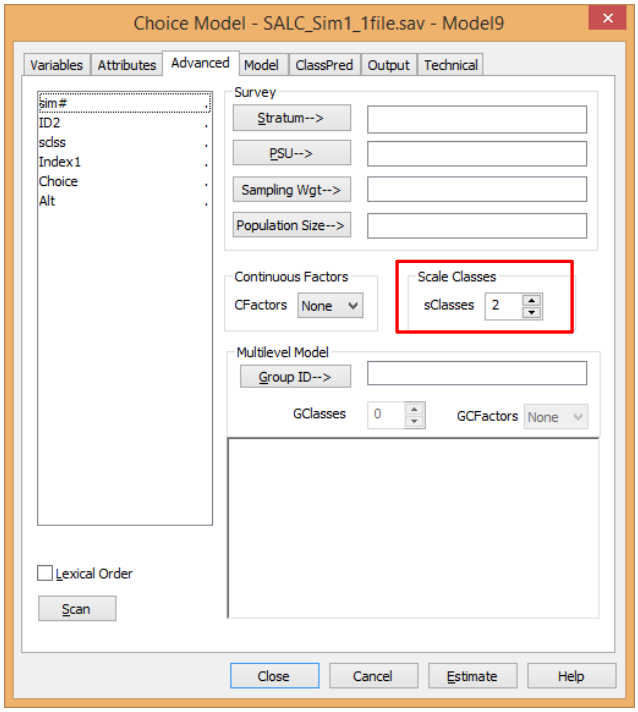


Figure 8. 2 sClasses specified in the Advanced tab.

Next we will see how well this model recovers the true segments.

- Click on the '+' symbol next to Model9 in the Outline pane to expand the model output menus.
- Click on 'ProbMeans' to display this output (see Figure 9).

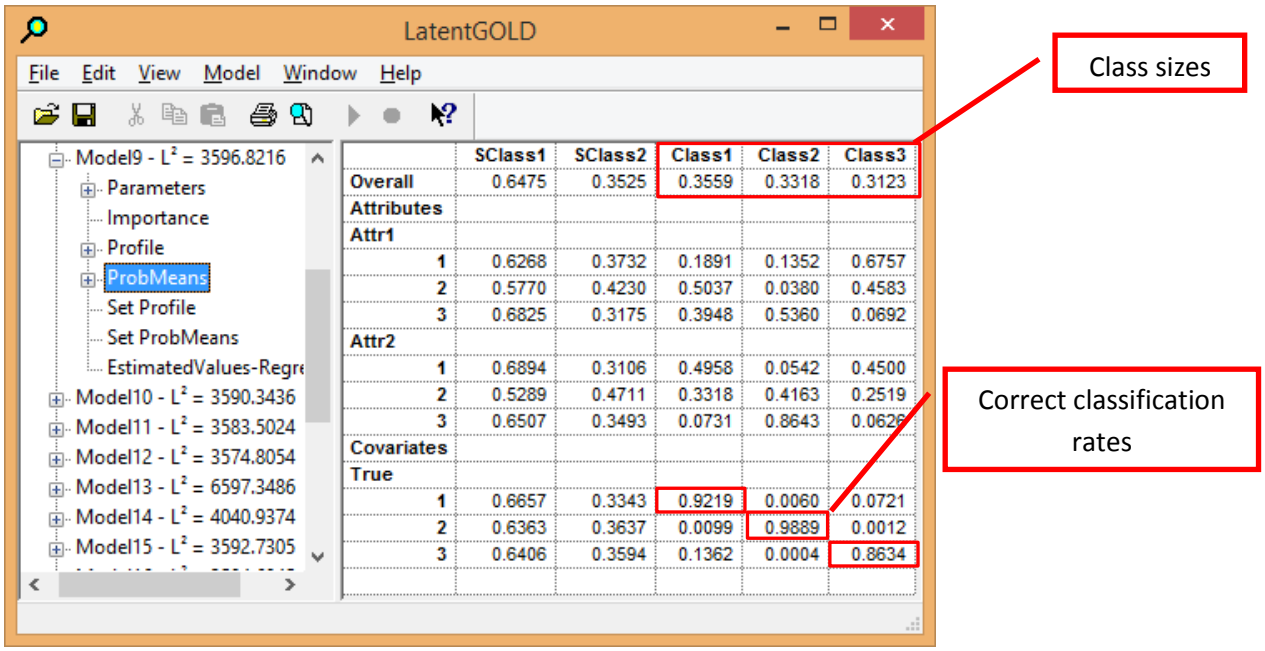


Figure 9. ProbMeans output for the 2-sClass, 3-class model.

As highlighted in Figure 9 (Class sizes), the classes are ordered from largest to smallest. Although all classes are estimated to be about the same size, 'Class1' is largest, containing about 36% of the cases.

In the Covariates section of the output, we can see how well each of the true segments is recovered. For example, for true Segment 3 (True = 3), 86.34% are classified into the same class (Class3), while 13.6% are mistakenly assigned to a different class (Class 1). Since the simulated data consist of equal numbers of cases in each of the three true classes, we can obtain the overall correct classification rate by simply averaging the three rates highlighted in Figure 9:

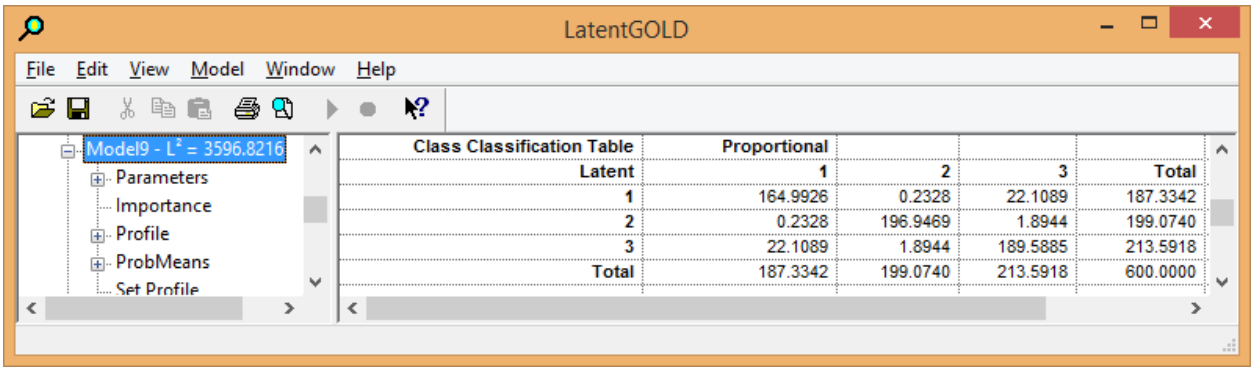
(.8634 + .9219 + .9889)/3 = .925.

Thus, we see that overall, 92.5% are correctly classified according to the variable 'True' when using proportional class assignment. This compares to 87.9% if scale heterogeneity is ignored[6].

Latent GOLD also reports various Classification Statistics as part of its summary output. In the absence of the variable True which identifies the true population segments, we can examine the Classification Table which provides the expected correct classification information under the assumption that the model is true.

To view the Classification Table under proportional assignment

- Click on 'Model 9' and scroll down to the Classification Statistics section (see Figure 11).



Class Classification Table	Proportional			
Latent	1	2	3	Total
1	164.9926	0.2328	22.1089	187.3342
2	0.2328	196.9469	1.8944	199.0740
3	22.1089	1.8944	189.5885	213.5918
Total	187.3342	199.0740	213.5918	600.0000

Figure 10. Classification Table under proportional assignment for the 2-sClass, 3-class model.

To calculate the *expected* correct classification rate of the true Latent Class 3 under this model:

164.9926/187.3342 = 88.1% which is close to the actual 86.3% obtained above.

To calculate the overall *expected* correct classification rate:

(189.5885 + 196.9469 + 164.9926)/600 = 91.9% which is close to the actual 92.5% rate computed above.

Next, we will examine the part-worth parameter estimates obtained from Model9.

[6] Ignoring scale heterogeneity results in a 4-class model (Model4). Averaging the three highest probabilities in the ProbMeans output for that model results in a reduced overall correct classification rate of 87.9%. For a graphical comparison of these 2 models with respect to correct classification of true Segment 3, see Figure 1.

➢ Click on 'Parameters' (see Figure 11).

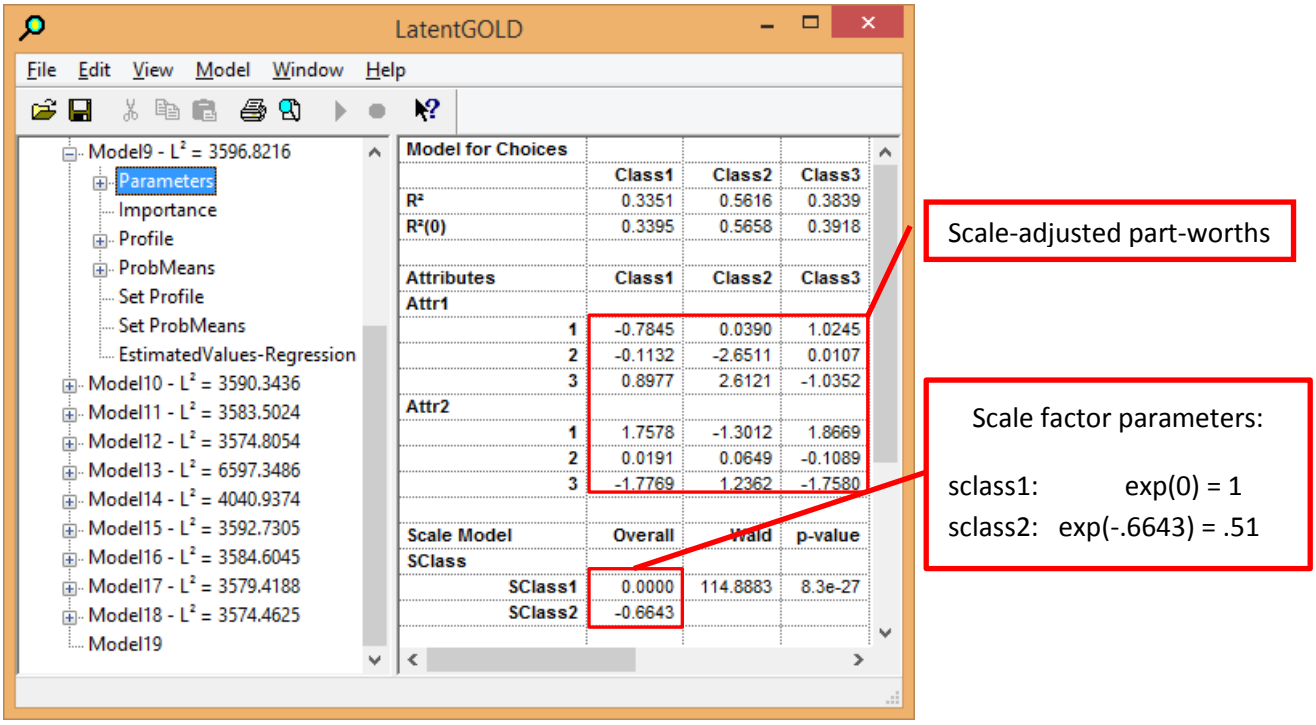


Figure 11. Parameters output for the 2-sClass, 3-class model.

These 'scale-adjusted' part-worth estimates are seen to be close to the population parameters shown in Table 1. (Standard errors for these estimates can be obtained by right clicking in the Output window.)

Since the scale factor parameters are log-linear, we need to exponentiate them to obtain estimates for the scale factors. The estimated scale factor for sclass2 is .51 (see Figure 12), near the true value of .5.

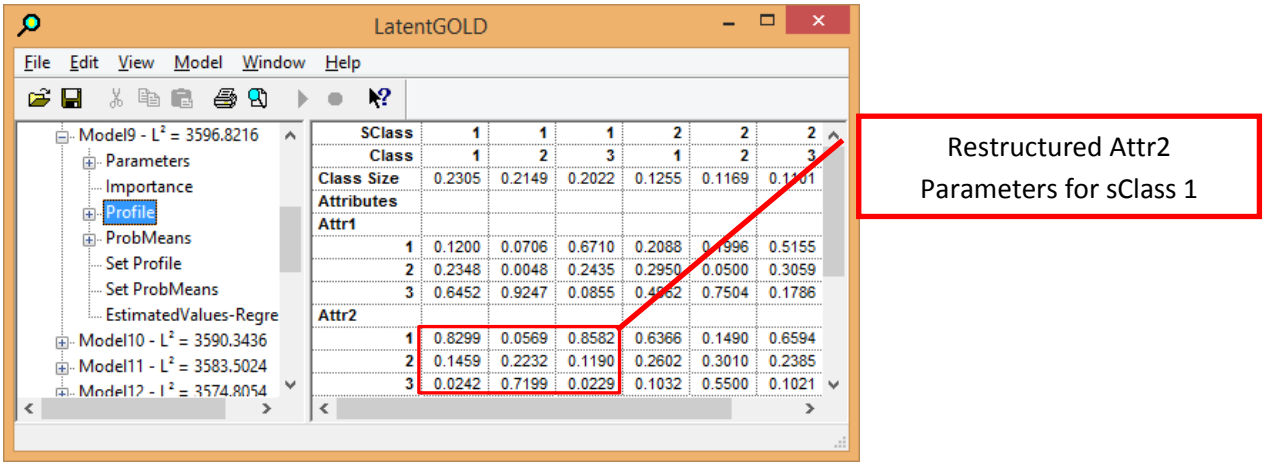
Figure 1 provides a plot of 2 individual part-worth coefficients (b1 and b4)[7] for true Segment 3 respondents. As indicated on Table 1, the *population* parameters for these true Segment 3 respondents (identified by 'True'=3), are B1=1 and B4=2. Thus, if the models plotted in Figure 1 were perfect, all of these Segment 3 cases would be plotted at the single point (b1,b4) = 1,2.

As can be seen, for the unadjusted 4-class model (Model 4), the points form a triangle, with most points in the upper right vertex. The other points have shrunken values for b1 and/or b4, many of which are misclassified (into latent class 1 or 4). It turns out that these are mostly respondents in true sClass = 2, the higher error sClasses. Thus, these less consistent respondents are much less likely to be classified correctly.

The Profile output restructures the parameters to probabilities. To view this output:

[7] Individual part-worth coefficients are weighted averages of the class-specific part-worths, where the individual posterior membership probabilities for each respondent serve as the weights for that respondent. They can be requested to be output to a file using the ClassPred tab. For further information about individual coefficients, see Latent GOLD Choice 5.0 tutorial #11.

- Click 'Profile' (Figure 13).



SClass	1	1	1	2	2	2
Class	1	2	3	1	2	3
Class Size	0.2305	0.2149	0.2022	0.1255	0.1169	0.1101
Attributes						
Attr1						
1	0.1200	0.0706	0.6710	0.2088	0.1996	0.5155
2	0.2348	0.0048	0.2435	0.2950	0.0500	0.3059
3	0.6452	0.9247	0.0855	0.4962	0.7504	0.1786
Attr2						
1	0.8299	0.0569	0.8582	0.6366	0.1490	0.6594
2	0.1459	0.2232	0.1190	0.2602	0.3010	0.2385
3	0.0242	0.7199	0.0229	0.1032	0.5500	0.1021

Figure 13. Profile output for the 2-sClass, 3-class model.

Each restructured parameter (highlighted in Figure 13) is interpretable as the probability of choosing that attribute level in a choice between 3 alternatives, each consisting of different levels of the associated attribute but the same levels of all other attributes[8]. For example, for class 1 persons in sClass 1, the profile probabilities for Attribute 2, can be seen in the highlighted section associated with Column identified by sClass = 1 and Class = 1. These are .8299, .1459, and .0242.

Since the 3 alternatives associated with choice set 1 consist of different levels of Attribute 2 but the same level (level 3) of Attribute 1 (see Table 4), the 3 profile probabilities also correspond to choice probabilities for this choice set.

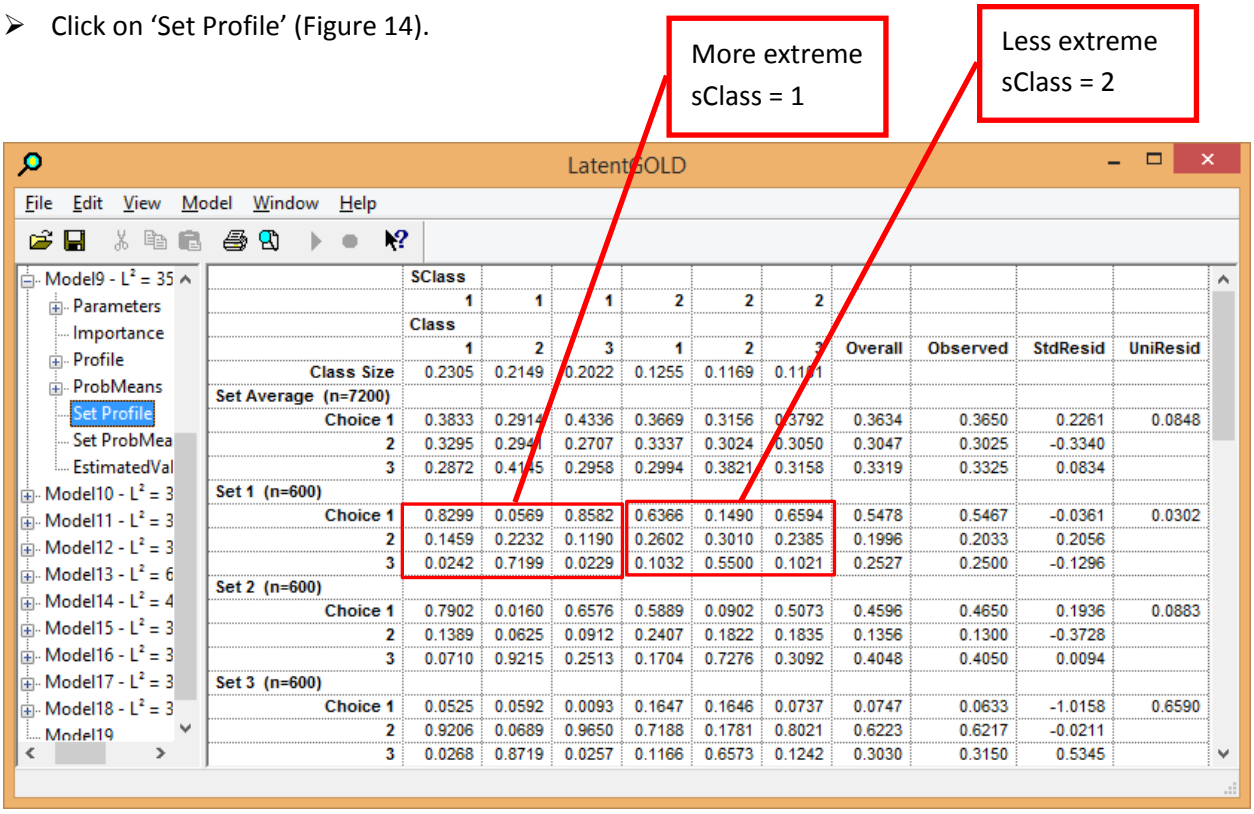
Alternative	Attribute 1	Attribute 2
Alt1	3	1
Alt2	3	2
Alt3	3	3

Table 4. Set 1 Alternatives.

[8] As pointed out by Magidson, Eagle, and Vermunt (2003), for attributes such as price, these restructured parameters indicate price sensitivities rather than choice probabilities. Choice sets such as set 1 would not be included in the design if the only attribute that varied across alternatives was price or a similar attribute.

Next we will examine the Set Profile output which shows the predicted choice probabilities for *all* choice sets and we will verify that the choice probabilities for choice set 1 match the corresponding profile probabilities.

➢ Click on 'Set Profile' (Figure 14).



	SClass									
	1	1	1	2	2	2				
	Class									
	1	2	3	1	2	3	Overall	Observed	StdResid	UniResid
Class Size	0.2305	0.2149	0.2022	0.1255	0.1169	0.1101				
Set Average (n=7200)										
Choice 1	0.3833	0.2914	0.4336	0.3669	0.3156	0.3792	0.3634	0.3650	0.2261	0.0848
2	0.3295	0.2941	0.2707	0.3337	0.3024	0.3050	0.3047	0.3025	-0.3340	
3	0.2872	0.4145	0.2958	0.2994	0.3821	0.3158	0.3319	0.3325	0.0834	
Set 1 (n=600)										
Choice 1	0.8299	0.0569	0.8582	0.6366	0.1490	0.6594	0.5478	0.5467	-0.0361	0.0302
2	0.1459	0.2232	0.1190	0.2602	0.3010	0.2385	0.1996	0.2033	0.2056	
3	0.0242	0.7199	0.0229	0.1032	0.5500	0.1021	0.2527	0.2500	-0.1296	
Set 2 (n=600)										
Choice 1	0.7902	0.0160	0.6576	0.5889	0.0902	0.5073	0.4596	0.4650	0.1936	0.0883
2	0.1389	0.0625	0.0912	0.2407	0.1822	0.1835	0.1356	0.1300	-0.3728	
3	0.0710	0.9215	0.2513	0.1704	0.7276	0.3092	0.4048	0.4050	0.0094	
Set 3 (n=600)										
Choice 1	0.0525	0.0592	0.0093	0.1647	0.1646	0.0737	0.0747	0.0633	-1.0158	0.6590
2	0.9206	0.0689	0.9650	0.7188	0.1781	0.8021	0.6223	0.6217	-0.0211	
3	0.0268	0.8719	0.0257	0.1166	0.6573	0.1242	0.3030	0.3150	0.5345	

Figure 14. Set profile output for the 2-sClass, 3-class model.

For each set we see that the choice probabilities differ not only across classes but also across scale classes.

The first three columns in Figure 14 provide the predicted choice probabilities for each class among the *lower* error respondents (sClass = 1). Note the different choice preferences across classes. The next three columns show these probabilities shrink (are less extreme) for the *higher* error respondents (sClass = 2). The column 'Overall' aggregates the six columns and the two right-most columns compare the overall choice probabilities with the corresponding observed proportions. The highlighted section above confirms that the choice probabilities for choice set 1 match the corresponding profile.

Note also that the standardized residuals (StdResid) and univariate residuals (UniResid) are all quite small (magnitude less than 2.0) showing that the predicted choice probabilities obtained from this model (reported in the column 'Overall') are not significantly different than the observed choice proportions.

For purposes of clarity we will now rename Model 9 to '(2sClass,3class)':

- Click twice on Model9 to enter the editing mode.
- Rename Model9 to '(2sClass, 3Class)' (Figure 15).

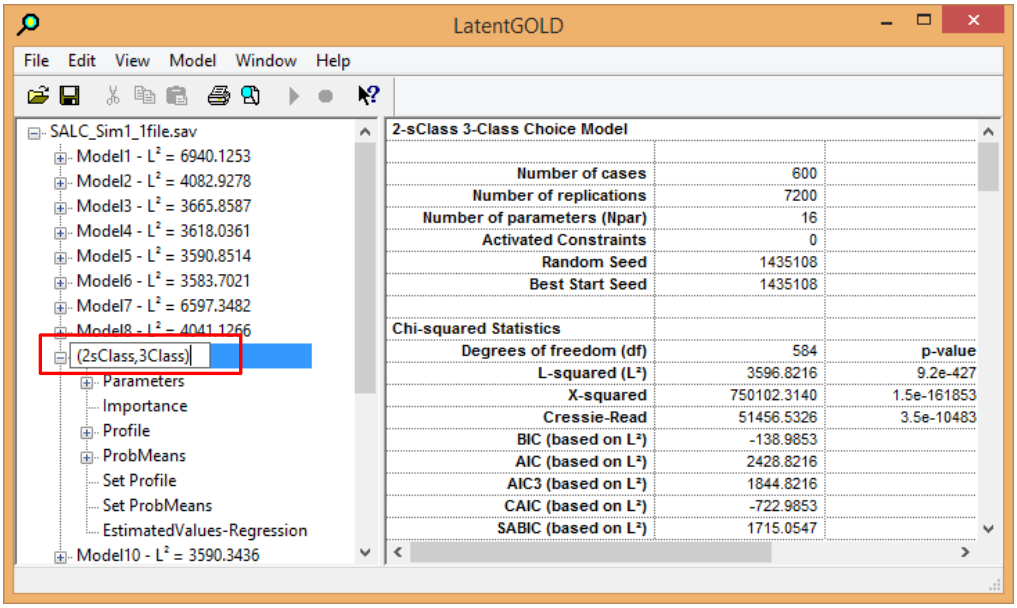


Figure 15. Renaming Model9 to '(2sClass,3Class)'.

Estimating Models with LG Choice Syntax

All SALC models above were setup and estimated using Latent GOLD's point-and-click user interface. These models assume that there are a distinct number of sClasses (discrete scale heterogeneity) and that the scale classes are independent of the classes. We simulated the data in accordance with these model specifications – 2 sclasses that are independent of the 3 classes.

More general SALC models that relax these assumptions can be estimated by LG Choice, but to do so requires the LG-Syntax module. We will now show how to setup some more general SALC models using the LG-Syntax, including models that assume continuous scale heterogeneity.

To copy the current model setup to the LG-Syntax editor:

- Right click on model '(2sClass,3Class)'.
- Select 'Generate Syntax' (Figure 16).

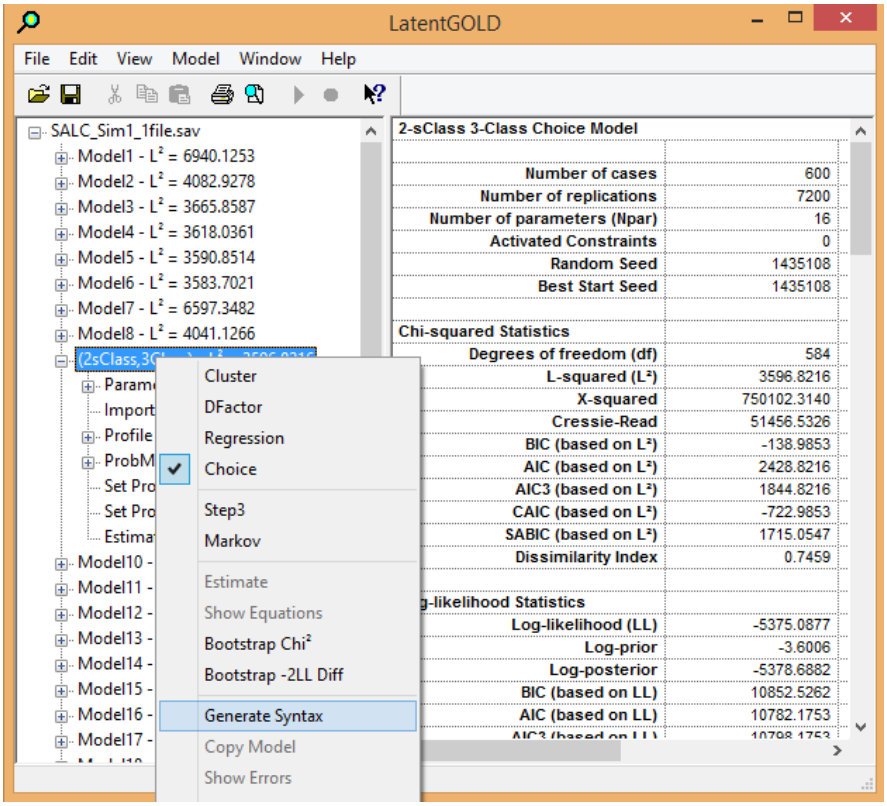


Figure 16. Converting the (2sClass-3class) model to a syntax model.

A separate (new) syntax tree appears above the other tree as shown in Figure 17, similar in appearance to a new dataset being opened. Note that the model name '(2sClass,3Class)' is preserved in the syntax.

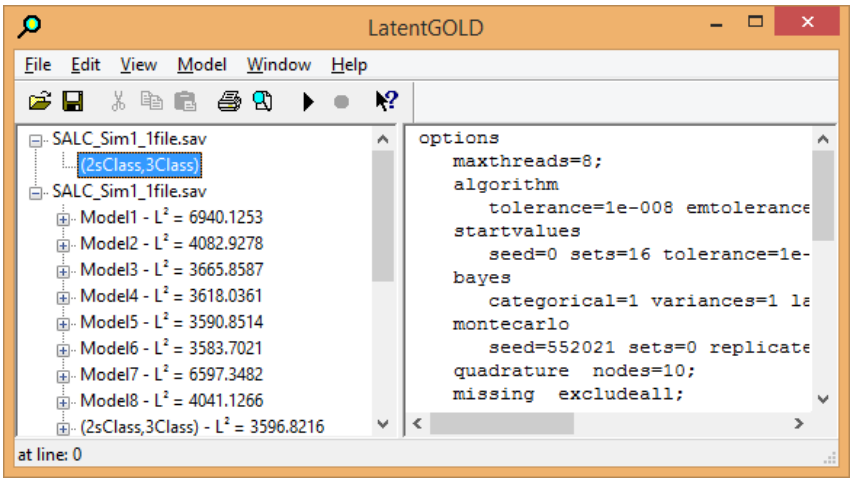


Figure 17. Syntax tree above the GUI tree.

- In the syntax editor, scroll down to the variables section (Figure 18).

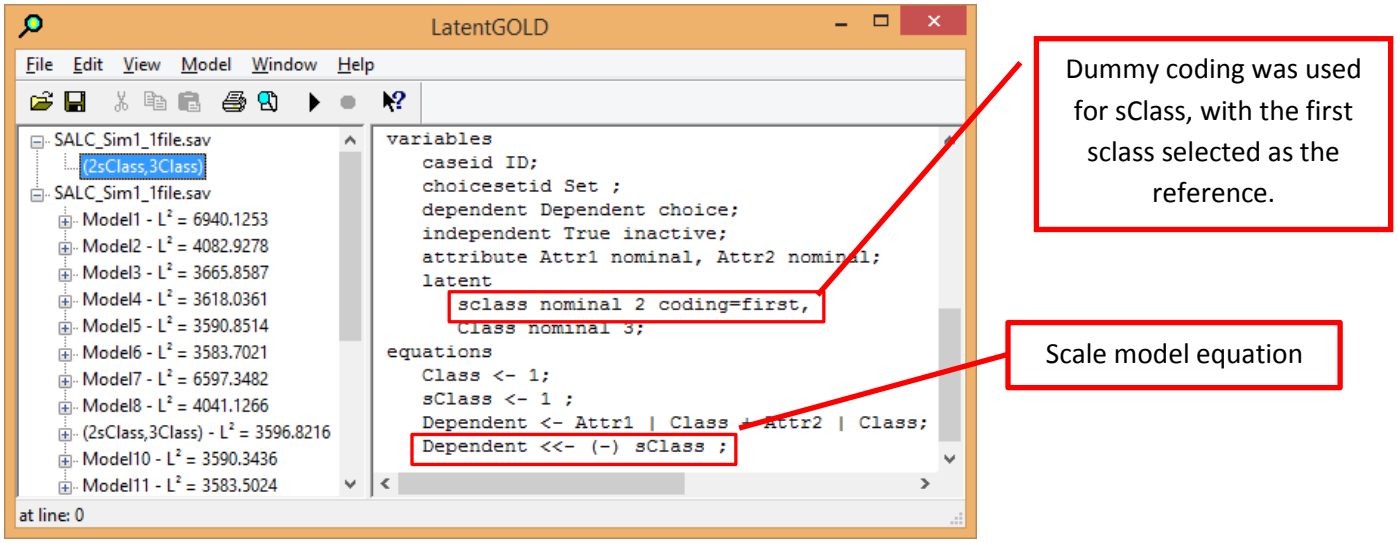


Figure 18. Variables and Equations Sections in the LG-Syntax.

The symbol '<<-' is used in the syntax to define the scale model. The scale model equation at the bottom of Figure 18 uses sClass as a nominal predictor which means that separate scale factor parameters are estimated for each sClass. (Recall these scale parameter estimates which were shown in Figure 11). The symbol '(-)' in the scale model orders the sClasses from high to low so that the first sClass corresponds to that sClass having the highest scale factor (lowest error), which serves as the reference sClass.

SALC models which relax the assumption that scale is independent of class

Next we will show how to modify this model to allow these two latent variables sClass and Class to be correlated, with different scale factors estimated for each class:

- Right click on the (2sClass,3Class) model and select 'Copy Model'
- Make the changes as shown in Figure 19.

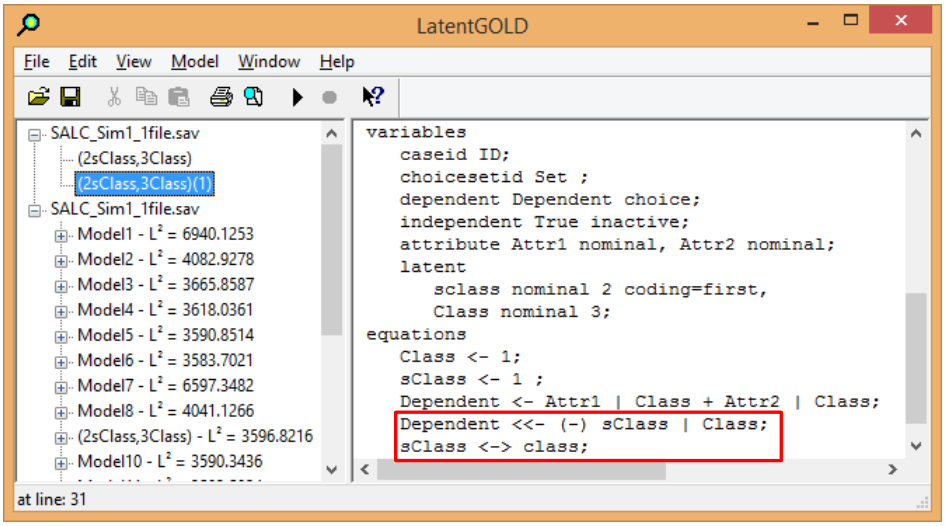


Figure 19. Edited syntax to allow correlation between SClass and Class with different scale factors for each class.

SALC models with a continuous scale factor

In this final section we will show how to estimate a SALC model with a *continuous* latent scale factor, which assumes that the log-scale factor is normally distributed.

Instead of estimating a 3-class SALC model with a *dichotomous* latent scale factor, suppose we wanted to estimate a 3-class SALC model with a *continuous* scale factor. That is, rather than assuming that each respondent has either low or high error, we assume that respondents vary continuously with respect to their error variance.

The following syntax shows the changes that would be made (Figure 20).

- Right click on the '(2sClass,3Class)' model and select 'Copy Model'
- Left click twice on the newly created model and rename the model to '(sCFactor,3Class)'
- Make the changes as shown in Figure 20.

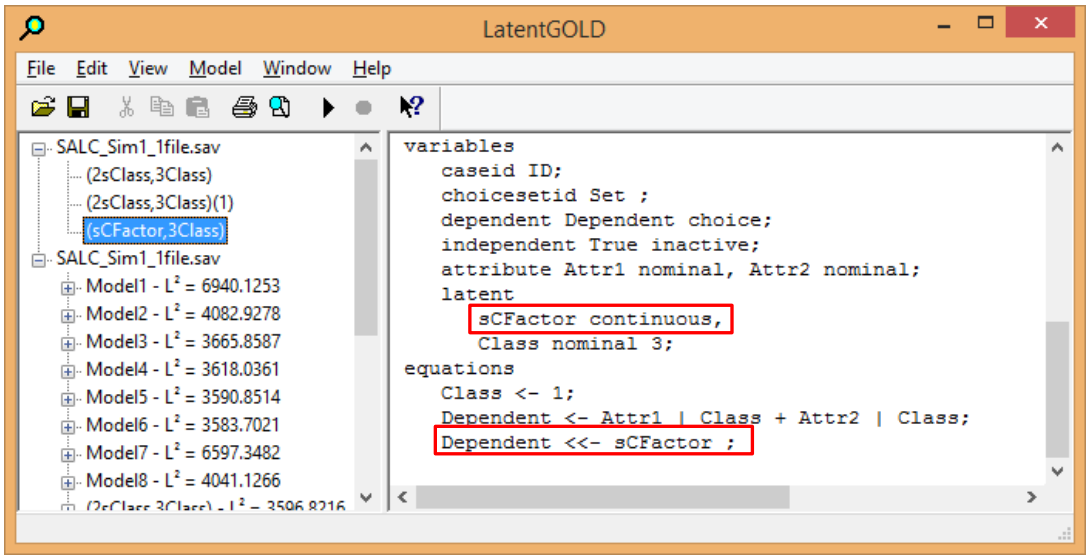


Figure 20. 3-class SALC model with a *continuous* latent scale factor.

If you estimate this model you will find that the dichotomous and continuous scale models result in very similar parameter estimates and very similar classification results for these data. It is noteworthy that if the data were simulated with a continuous rather than a dichotomous scale factor, the results again turn out to be very similar.

See the LG Choice 5.0 Upgrade Manual and the references in that document for more details.

References

Burke, P., C. Burton, T. Huybers, T. Islam, J. Louviere, and C.Wise (2010). The scale-adjusted latent class model: application to museum visitation. Tourism Analysis, Vol.15, pp.147-165.

Louviere, J. J., and T. Eagle (2006). Confound it! That pesky little scale constant messes up our convenient assumptions. Proceedings, 2006 Sawtooth Software Conference, Sawtooth Software, pp.211-228.

Magidson, J., and J.K. Vermunt (2007). Removing the scale factor confound in multinomial logit choice models to obtain better estimates of preference. Proceedings, 2007 Sawtooth Software Conference, Sawtooth Software, pp.139-154.

Magidson, J., T. Eagle, and J.K. Vermunt (2003). New Developments in Latent Class Choice Models. Proceedings, 2003 Sawtooth Software Conference, Sawtooth Software, pp. 89-112.