



Sawtooth Software

RESEARCH PAPER SERIES

Real-Time Detection of Random Respondents in MaxDiff

Keith Chrzan and Bryan Orme
Sawtooth Software, Inc.

Real-Time Detection of Random Respondents in MaxDiff

Keith Chrzan and Bryan Orme, Sawtooth Software, Inc.

February 2022

1.0 Random Respondents and Their Identification

Respondents who answer survey questions randomly create obvious problems for researchers. Respondents who answer questions in choice experiments randomly arguably cause more trouble, because the random noise they add biases the experiment's results: their utilities tend to be smaller (in absolute magnitude) than the utilities of respondents who answer conscientiously. As a result, random respondents cause predictable problems:

- Their muted utilities can reduce the sensitivity of tests for differences in utility across subgroups of respondents
- In simulations they reduce sensitivity to changes in attribute levels, which can cause odd results in share of preference and inflated WTP estimates
- They can masquerade as a unique segment of respondents who lack strong preferences (a segment that can also be difficult to identify in a typing tool)

Indices of fit from utility-generating models have been used to identify random respondents for decades, dating back at least to the use of R^2 to identify random respondents in ratings-based conjoint (Chrzan 1991). More recently, Orme (2019) suggested a way to use the root likelihood (RLH) fit statistic from hierarchical Bayesian multinomial logit (HB-MNL) to identify random respondents:

- For a large number (say 1,000) cases, generate a set of random respondents to your choice (CBC, MaxDiff, ACBC) experiment
- Run HB-MNL on these random responses and identify the RLH value below which a target proportion (e.g. 95%) of random respondents fall
- Use this value as a cutoff to identify random human respondents in your empirical data set (see Section 5.0 below for an important caveat about this method)

Compared to two methods based on latent class MNL (Hoogerbrugge and de Jong 2019; Magidson and Vermunt 2007) the RLH-based method performs better at targeted detection of random respondents (Chrzan and Halversen 2020).

It turns out that we can extend the RLH-based method to allow for real-time detection of random respondents, at least for MaxDiff experiments, using the on-the-fly utility estimation¹ available in Sawtooth Software's Lighthouse Studio and Discover interviewing platforms. This allows us to identify likely random respondents *before* they complete their surveys and before they qualify for compensation.

Using a fit statistic to identify misbehaving respondents is even more incisive for MaxDiff experiments than for CBC or ACBC experiments. While the fit statistic for either MaxDiff or conjoint experiments can reliably identify random responders, a simplifying/speeding respondent can easily fool a CBC or ACBC experiment and achieve a good fit statistic merely by choosing None a lot, or always picking (say) the lowest price alternative on the screen. In contrast, it's extremely hard for speeding/simplifying responders to fool the MaxDiff fit statistic (as long as each item appears multiple times; see our results below).

After demonstrating how to effect real time detection of random respondents in MaxDiff experiments using both Lighthouse Studio and Discover, we will validate the real-time detection method using two empirical data sets. Then, we will apply the method to see how well it would have performed on a convenience sample of several recent commercial studies.

In the Appendix, we report the RLH cutoff values for sample MaxDiff experiments to identify random responders, at 80%, 90% and 95% detection rates. If your MaxDiff study reasonably resembles these MaxDiff experiment examples, you can use these RLH cutoff values.

2.0 Software Configuration Steps

Detecting random and near-random responders in MaxDiff surveys may be done in real time (at the moment the respondent answers the survey) using either Sawtooth Software's Lighthouse Studio or Discover platforms. In either case, we recommend each item be seen 3x or preferably 4x by each respondent to discriminate accurately between random responders and respondents who are answering with the usual tendencies of human error (see details in Section 3).

Lighthouse Studio

Let's assume you've programmed a MaxDiff exercise called MXD.

¹ The on-the-fly utility estimation procedure uses purely individual-level MNL estimation via a simplified gradient search procedure as described in "Becoming an Expert in Conjoint Analysis" 2nd Edition (Orme, Chrzan 2021).

First, determine the threshold fit for your study that will identify most bad respondents, while throwing out very few reasonably good respondents:

- After programming your MaxDiff exercise (to have each item display a suggested 3x or preferably 4x), generate 1000 random respondents using Lighthouse Studio's random respondent generator (click *Test + Generate Data* and specify at least 1000 respondents). After the data are generated, click the *Get Data* option that appears which will copy the test respondent data into your respondent database.
- Calculate on-the-fly utility scores for your random respondents by clicking *Analysis + Analysis Manager* and choosing *Add* and selecting *Logit* as the analysis type. Using the cog/gear icon at the right of the *Logit Analysis Type* field, click the *Settings* link and then click the box under *Advanced Settings to Recreate on-the-fly scores and rankings*.

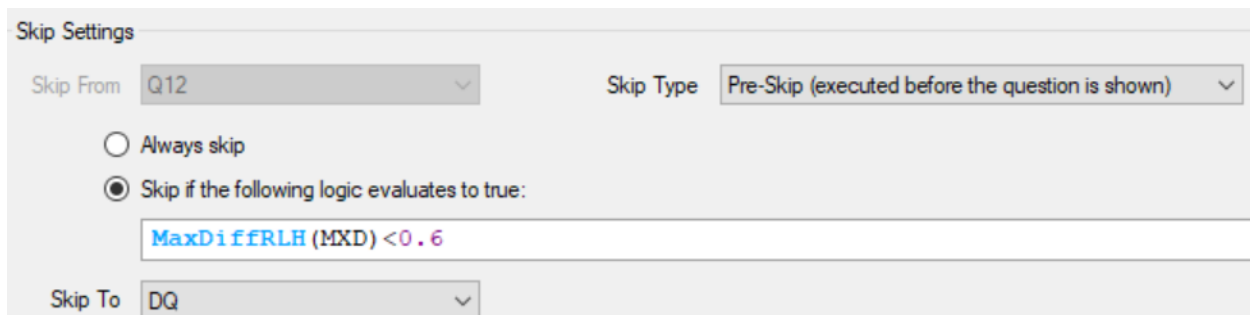
The screenshot shows the 'MaxDiff Logit Analysis' settings window. On the left is a sidebar with a tree view containing: 'Respondent Filter' (with sub-item 'Include All Respondents'), 'Weights' (with sub-item 'Equal'), 'Sets to Include' (with sub-item 'All Sets Included'), 'Constraints' (with sub-item 'Default'), 'Anchored Scaling' (with sub-item 'None'), 'Settings' (with sub-item 'Default'), and 'Default'. The 'Settings' item is selected. The main area is divided into two sections: 'General Settings' and 'Advanced Settings'. Under 'General Settings', there are three input fields: 'Step size' with the value '1', 'Maximum number of iterations' with the value '100', and 'Convergence limit' with the value '0.00001'. Under 'Advanced Settings', there is a dropdown menu for 'Tasks to include for best/worst data' set to 'Best and Worst (recommended)', a checkbox for 'Display covariance matrix' which is unchecked, and a checkbox for 'Recreate on-the-fly scores and rankings' which is checked.

Close this dialog and then click *Run* to estimate on-the-fly scores using the individual-level logit approach. A report appears with multiple tabs (along the bottom). Click the *On-the-fly Scores* tab to see the RLH scores for each of your random respondents.

Copy these scores into a program like Excel and sort them from lowest to highest RLH score. Use this sorted array of RLH scores to find the cutoff value below which 80%, 90%, 95% etc. of the random respondents fall (see Section 3 for more guidance; we'd generally recommend using an 80% or 90% cutoff to avoid throwing away very many reasonably consistent human respondents). And, of course, delete these test respondents after you've used them for their purpose so they don't get combined locally on your hard drive with your eventual real respondents (access that data table using *Field + Data Management* to delete these respondents).

In the page directly following the last MaxDiff question, add skip logic (typically “pre-skip” logic, so that the skip logic is evaluated before the page is displayed) that immediately sends a “bad” respondent to a disqualified (incomplete) terminate question that you have added at the end of your survey.

The skip logic invokes the MaxDiffRLH instruction, which has one argument (the name of the MaxDiff exercise, which in our case is MXD):



Skip Settings

Skip From: Q12

Skip Type: Pre-Skip (executed before the question is shown)

☐ Always skip

☒ Skip if the following logic evaluates to true:

MaxDiffRLH(MXD) < 0.6

Skip To: DQ

In the example above, Q12 is the question directly following the last MaxDiff question. The value 0.6 in this example represents the fit threshold you’ve determined for your MaxDiff experiment (***note: you’ll need to specify the right threshold for your experiment following the previous instructions or by referring to RLH values in the Appendix; 0.6 is just for illustration***). A respondent with a fit statistic lower than the threshold is immediately skipped (prior to displaying Q12) to a terminating question called DQ. DQ is a terminating question where the respondent is marked as not complete (which will not count this respondent toward any completed quotas).

Multiple terminate questions can exist in your survey and you mark a terminating question as incomplete using the *Settings* tab of the terminate question.

Discover

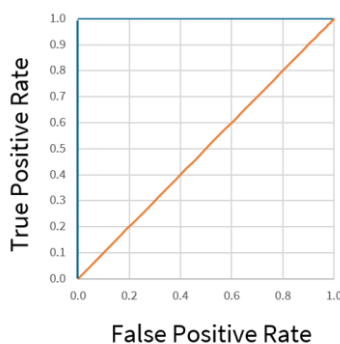
Discover’s platform uses the same utility estimation algorithm and the same MaxDiffRLH instruction as Lighthouse Studio. It also has skip pattern capabilities, so the process is very similar as implementation in our web-based platform (as described above).

The catch with using Discover is there is no random respondent generator in this platform, which is needed for calculating the right RLH fit cutoff to distinguish random responders from the non-random responders. Moreover, Discover doesn’t have an option to estimate the on-the-fly scores for respondents after the fact that have already been generated/collected. So, our recommendation at this point is to use Lighthouse Studio to estimate the RLH cutoff, mirroring your Discover project (# of items, # sets, #items/set) within Lighthouse Studio, and following the instructions above in the Lighthouse Studio section for calculating the RLH cutoff.

Or, if your study specifications are similar enough to those covered in the Appendix of this article, you can refer to the RLH cutoffs listed there.

3.0 Empirical Studies

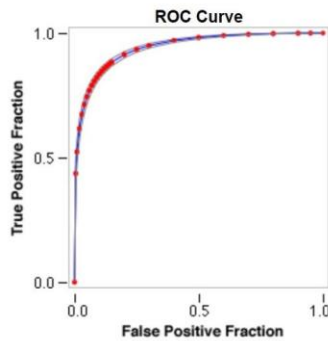
One way to evaluate a diagnostic test involves plotting a receiver operating characteristic (ROC) curve and measuring the area under the curve (AUC). Plotted here are two extreme results:



The orange line represents a perfectly worthless diagnostic test – at every value of the test statistic true positive rates equal false positive rates and the AUC is 0.50. The blue line represents a test capable of achieving a 0% false positive rate and a 100% true positive rate (i.e. it includes the point in the upper left-hand corner) and it has an AUC of 1.0. Just for comparison, a cursory internet search found AUCs of 0.934 and 0.983 reported for laboratory-run Covid tests.

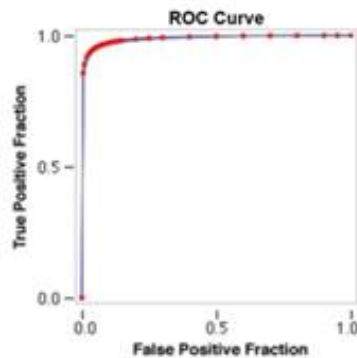
In our empirical studies we use MaxDiff utilities estimated from human respondents in a pair of R&D studies and we program robotic respondents to choose in a manner consistent with Random Utility Theory and the multinomial logistic regression model. We have the robots choose MaxDiff responses according to the human respondents' utilities plus randomly drawn standard Gumbel errors – call these our RUM Respondents. In addition, we program a second set of respondents to make entirely random choices to the MaxDiff questionnaire – call these our Random Respondents.

In our first empirical study we have utilities from 2,400 human respondents and we generate robotic 2,400 RUM respondents. We also created 1,800 Random Respondents. The MaxDiff experiment featured 36 items shown in 18, 27 or 36 quads (to allow each respondent to see each item 2, 3 or 4 times, respectively). For two item-views per respondent the ROC below produces an AUC of 0.943 and the table below shows that we can identify 80% of Random Respondents at a cost nearly 8% of RUM respondents:



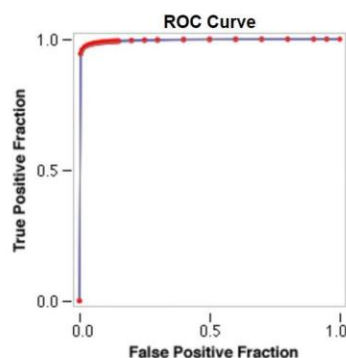
RLH cutoff (cut below)	Randoms Detected (%)	RUMs Mis-classified (%)
0.8286	95	26.83
0.7890	90	16.17
0.7606	85	11.21
0.7374	80	7.92

At three item-views per respondent the AUC improves to 0.989 and we can detect 80% of Randoms at a cost of barely 1% of valid RUM Respondents:



RLH cutoff	Randoms Detected (%)	RUMs Mis-classified (%)
0.53059	95	4.67
0.50225	90	3.20
0.48117	85	2.00
0.46908	80	1.13

As you would expect, with 36 questions (each item shown four times per respondent) the AUC improves even more (to a near-perfect 0.996) and we can detect 80% of Random Respondents at a cost of only half a percent of RUM Respondents:



RLH cutoff	Randoms Detected (%)	RUMs Mis-classified (%)
0.42032	95	1.67
0.40498	90	1.21
0.39630	85	0.79
0.38820	80	0.50

The second empirical study uses utilities from human respondents for 10 items, this time shown in 4, 6 or 8 quintiles (to allow for 2, 3 or 4 views per item per respondent). Using 1,000 RUM Respondents and 1,000 Random Respondents, AUCs and false positive rates for several levels of Random Respondent detection appear in the table below.

	2x	3x	4x
AUC	0.916	0.980	0.994
% RUMs lost at Randoms cutoff of			
95%	44.6	10.5	3.3
90%	23.0	4.7	2.3
85%	16.2	3.1	1.6
80%	12.6	2.5	1.2

Our empirical studies show that with enough questions to allow each respondent to see each item three or four times, we can do an excellent job of distinguishing Random from RUM Respondents in our R&D studies.

4.0 Application to Commercial Studies

We applied the on-the-fly utility and RLH estimation to data collected in a convenience sample of eight recent commercial studies. These surveys included consumer and B2B studies, with topics ranging from brand image to messaging to psychometrics and importance measurement, for industries ranging from automobile to IT, from gaming to pharmaceuticals. These results apply to surveys whose respondents were already subjected to ordinary QC procedures (e.g. detecting straightliners and speeders). We found a range of results:

Study	n	Views/ item	80% RLH cutoff	% Randoms
1	100	3.1	0.46015	0.0
2	2750	3.3	0.38426	33.4
3	164	3.0	0.46389	2.4
4	85	4.0	0.48626	10.6
5	118	3.0	0.61470	16.9
6	9857	3.0	0.39292	22.7
7	203	2.3	0.40853	19.7
8	760	3.0	0.48074	19.5
9	861	3.2	0.45886	1.6

The large range in sample quality (with anywhere from 0% to 33.4% of respondents appearing to answer randomly) may owe to different cleaning procedures, different panel quality, different levels of interest respondents may have had to the topic of the survey or different levels of fatigue due to the MaxDiffs' locations in the various surveys. For example, study #1 (0% random respondents) was a survey of employee benefits among a sample of employees, whereas study #2 surveyed consumers, many of them lapsed users, about aspects of an entertainment product.

5.0 Caveat

Before concluding, we should note a caveat about the logic of the RLH-based random respondent detection method: it allows us to find the probability that respondents have low RLHs given that they are random. Unfortunately, it does NOT allow us to infer the probability that a respondent with a low RLH has answered randomly. Respondents who fail the RLH hurdle may be random respondents or they may be "low signal" respondents with weak preferences (i.e. respondents with utilities too small to overcome the Gumbel errors, with their mean of 0.5772 and their standard deviation of 1.2825 that the logit model assumes to perturb their choices).

Before deciding to exclude respondents who fail the RLH test we should weigh the relative harms and benefits of keeping them in our models. For example, if we think our subject matter is of little interest to respondents we might expect a lot of low signal RUM choosers whose (noisy) preferences reflect their true, weak, opinions. Or again, if we intend to base a needs-based segmentation on our MaxDiff experiment, we may not want a segment of random respondents who might be hard to identify in a segment typing tool

6.0 Summary/Recommendation

Using on-the-fly utility and RLH estimation enables researchers to detect random responders to MaxDiff exercises in real time and to exclude them from research samples without compensating them. Sawtooth Software's platforms for fielding MaxDiff experiments are especially valuable in this respect, as the skip pattern is very easily implemented that invokes the on-the-fly fit statistic. Using two data sets, we find that if each respondent sees each item 4x, one can exclude 80% of random responders while throwing away about 1% or fewer real respondents. Under these conditions, any real respondents that are falsely identified as random respondents are low-signal respondents who either have very little opinion on the subject matter and/or are paying a relatively low degree of attention in our survey.

References

- Chrzan, K. (1991) "Unreliable respondents in conjoint analysis: Their impact and identification," *Sawtooth Software Conference Proceedings*, 205-227.
- Chrzan, K. and C. Halversen (2020) "Diagnostics for random respondents in choice experiments," *Sawtooth Software Research Paper* downloaded from <https://sawtoothsoftware.com/resources/technical-papers/diagnostics-for-random-respondents-in-choice-experiments> on December 29, 2021.
- Hoogerbrugge, M and M. de Jong (2019) "Can we use RLH to assess respondent quality?" *Sawtooth Software Conference Proceedings*, 105-112.
- Magidson, J. & J.K. Vermunt (2007) "Removing the scale factor confound in multinomial logit choice models to obtain better estimates of preference," *Sawtooth Software Conference Proceedings*, 139-154.
- Orme, B.K. (2019) "Consistency cutoffs to identify 'bad' respondents in CBC, ACBC and MaxDiff," <https://www.linkedin.com/pulse/identifying-consistency-cutoffs-identify-bad-respondents-orme/?trackingId=g9yIG8GeasHXn79cZcC5JQ%3D%3D>

Appendix

RLH Cutoff Values to identify Random Responders for On-The-Fly Individual-Level Logit Estimation

The table below shows RLH cutoff values to identify 80%, 90%, or 95% of random responders for MaxDiff experiments involving 12, 18, and 24 items. We find that the number of items in the experiment (between 12 and 24) only modestly affects the RLH cutoff, but how many items are shown per set and especially how many times each respondent sees each item has a strong effect. The more times each item is displayed to each respondent, the harder it is for random respondents to achieve a higher RLH consistency score.

If your MaxDiff experiment doesn't conform very closely to these examples, you should follow the instructions in the body of this article for generating random respondents, computing the on-the-fly individual logit RLH fit values, and estimating your cutoff values.

We used 4000 random responders in each of the 12 simulated data sets below to compute these cutoff values. When the experiment doesn't allow the number of items to be shown exactly 3x or 4x, we use the closest multiple and note this in the table.

		Each item seen 3x:	Each item seen 4x:
12 items in experiment:	4 items shown per set:	80% cutoff: 0.483 90% cutoff: 0.545 95% cutoff: 0.603	80% cutoff: 0.396 90% cutoff: 0.431 95% cutoff: 0.463
	5 items shown per set:	80% cutoff: 0.441 90% cutoff: 0.515 95% cutoff: 0.580 (each item shown 2.92x)	80% cutoff: 0.330 90% cutoff: 0.361 95% cutoff: 0.392 (each item shown 4.17x)
18 items in experiment:	4 items shown per set:	80% cutoff: 0.466 90% cutoff: 0.510 95% cutoff: 0.551 (each item shown 3.11x)	80% cutoff: 0.395 90% cutoff: 0.419 95% cutoff: 0.444
	5 items shown per set:	80% cutoff: 0.418 90% cutoff: 0.467 95% cutoff: 0.513 (each item shown 3.06x)	80% cutoff: 0.343 90% cutoff: 0.371 95% cutoff: 0.395 (each item shown 3.89x)
24 items in experiment:	4 items shown per set:	80% cutoff: 0.475 90% cutoff: 0.513 95% cutoff: 0.551	80% cutoff: 0.390 90% cutoff: 0.412 95% cutoff: 0.430
	5 items shown per set:	80% cutoff: 0.432 90% cutoff: 0.475 95% cutoff: 0.516 (each item shown 2.92x)	80% cutoff: 0.327 90% cutoff: 0.348 95% cutoff: 0.366 (each item shown 4.17x)