Sawtooth Software

RESEARCH PAPER SERIES

Bandit MaxDiff: When to Use It and Why It Can Be a Better Choice than Standard MaxDiff

Bryan Orme
Sawtooth Software, Inc.

3210 N. Canyon Rd., Provo, Utah
+1 801 477 4700
www.sawtoothsoftware.com

Bandit MaxDiff: When to Use It and Why It Can Be a Better Choice than Standard MaxDiff

Bryan Orme, Sawtooth Software

Executive Summary

Bandit MaxDiff (best-worst scaling) achieves greater measurement precision than standard MaxDiff for items that have the highest utility scores. When that is your focus, you can vastly improve the efficiency for large MaxDiff projects, reducing data collection costs by 50% to 80%. Bandit MaxDiff is a new type of adaptive MaxDiff available in Lighthouse Studio v9.6 and later. Whether studying relatively few items or an enormous number of items, Bandit MaxDiff can significantly improve your results.

Background

MaxDiff (best-worst scaling) is a survey research technique for measuring the importance or preference of an array of items. Respondents are shown subsets of items (for example, five items per screen) and are asked to identify the best and worst items among those five. MaxDiff experiments provide much greater discrimination than standard rating scales and are free from scale use bias (Cohen 2003).

What is the *bandit* in Bandit MaxDiff? *One-armed bandit* is a slang term for slot machines in casinos. They have one arm (the lever you pull) and they usually take your money (like bandits).

A *One-Armed Bandit*

For at least the last sixty years, statisticians have been interested in what they have called *multi-armed bandit* problems. For example, if you want to invest your resources over multiple time periods across multiple activities, each with uncertain outcomes (like pulling different arms across multiple slot machines), how should you allocate the resources (your bets) to maximize the long-term payoff? Bandit solutions maximize the expected return over multiple time periods by effectively exploring the possible

outcomes (advertising, clinical trials, product launches, website modifications) while exploiting the information learned along the way.

Explore and Exploit

With Bandit MaxDiff (Fairchild *et al.* 2015), we *explore* by asking respondents to evaluate multiple attributes (items) in MaxDiff tasks. Once respondents begin completing MaxDiff questionnaires, we can exploit that prior information for the next respondents by oversampling the items that previously tended to be more preferred across the sample (or tended to be most important, if you're phrasing the question in terms of importance). After just a few respondents have completed the survey, we aren't very certain regarding respondent preferences. So, the prudent strategy is to allocate our resources relatively evenly across the full array of items until we gain more information. But, after collecting more data the certainty increases and it becomes more efficient to ask the next respondents to evaluate the most preferred items more often than the least preferred items, leading to much greater precision regarding what the sample believes the best items are in your study. The way we've implemented Bandit MaxDiff within Lighthouse Studio, the entire process is continuous and seamless. You do not need to start and stop the interviewing process.

Technical Details

Bandit MaxDiff employs Thompson Sampling to select the items to oversample for each new respondent. Thompson Sampling leverages prior estimates of each item's mean and variance which can be estimated via aggregate logit or adequately enough via much quicker ways such as counting analysis.

We recognize that our prior estimates of item preferences are subject to uncertainty, so we pick the items for the next respondent probabilistically via Thompson sampling by perturbing the prior means by normal draws with standard deviation equal to the population estimates from logit or counting analysis. For each new respondent, the perturbed item score draws are sorted from best to worst and the top *t* items are selected for inclusion in the respondent's MaxDiff questionnaire, where *t* is specified by the researcher (for studies involving a large number of items, we recommend *t* of around 30). To make the approach even more robust (see the section below regarding misinformed first respondents), we recommend that at least some of the items be selected with a very diffuse prior (very large variance) or even purely randomly.

Because logit is computationally intensive, quicker methods for estimating population means and variances can be used: for example counting analysis and either Gamma or Gaussian (normal) distributions. Lighthouse Studio's Bandit MaxDiff capability utilizes counting analysis to estimate each item's preference and variance, where the variance is estimated via the well-known *pq/n* formula for the variance of proportions. Our strategy for counting analysis involves exploding the MaxDiff task into all inferred pairwise comparisons and simply counting the percent of "wins" (*p*) for each item across all inferred pairs (across all tasks and respondents). We compute *n* as the average between the sample size and the number of times each item appears across all MaxDiff tasks for all respondents.

Is Bandit MaxDiff Just for Studies with a Large Number of Items?

No. Whether you are conducting a MaxDiff study with few (20 or fewer) items or an extremely large number of items (300 or more), Bandit MaxDiff will increase measurement precision for your sample's most preferred items. When studying few items, we recommend that each item be shown to every respondent: but the most preferred items for the population should be made to appear more times across each respondent's MaxDiff questions than the less preferred items (see Appendix A for more details regarding experimental design). For studies involving a large number of items, only a subset *t* of the items should be shown to any one respondent (such as t=30)—again oversampling the most preferred items.

Why Focus on Increasing Precision for Best Items?

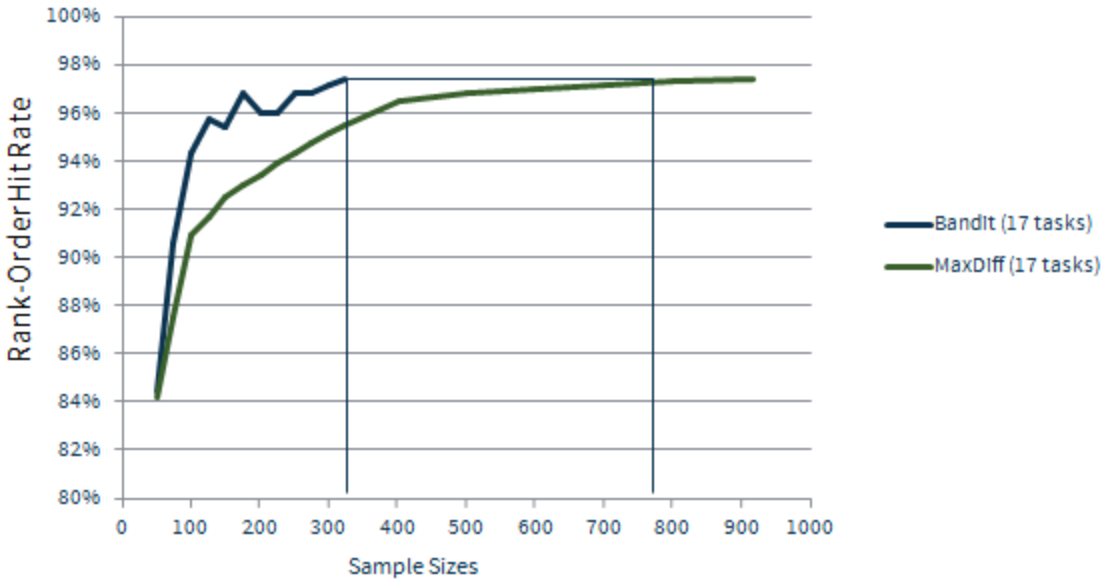
It's valuable to identify those items that are most preferred since they often represent levers that are strong indicators of choice or purchase. Increasing measurement precision for these highest-preference items generally will lead to better insights and better identification of successful marketing strategies. It often is wasteful to spend much respondent effort continuing to evaluate items that we are extremely confident are relatively undesirable.

How Much Better Is Bandit MaxDiff?

Bandit MaxDiff can save a great deal of money on data collection costs. Based on extensive simulations, we've found that MaxDiff studies involving 120 or more items can achieve predictive hit rates[1] with 200 to 250 respondents what would require 1,000 respondents to do with standard MaxDiff (a 75% to 80% reduction in data collection costs). With 300 or more items, you will likely experience 80% or more reduction in data collection costs (Fairchild *et al.* 2013). The greater the number of items, the more efficient Bandit MaxDiff becomes relative to standard MaxDiff design methods that sample equally across the items for all respondents (again, assuming the goal is to increase precision for the most preferred items).

During 2017, in cooperation with P&G we conducted an empirical test for their MaxDiff study involving 80 items. P&G interviewed approximately 2800 respondents (online consumer panel sample). 324 respondents received a Bandit MaxDiff questionnaire (17 tasks, 6 items per task) programmed in Lighthouse Studio, using essentially the same algorithm released in Lighthouse v9.6. 923 respondents received a standard MaxDiff questionnaire (17 tasks, 6 items per task). 1519 respondents served as holdout respondents for validation and received a standard MaxDiff questionnaire (27 tasks, 6 items per task). In terms of ability to predict correctly the rank-order of the top 10 items out of 80 as determined by the holdout respondents, the Bandit MaxDiff respondents achieved with 324 respondents what took 775 standard MaxDiff respondents to do (see graphic below). Bandit MaxDiff was 2.4x more efficient than standard MaxDiff for predicting the top 10 items for P&G's 80-item study.

[1] Of the top 3 globally preferred items (hit rate for top 3).



Holdout Rank-Order Prediction Accuracy (Top 10) for P&G Study with 80 Items

If the goal is to identify the top few items, even 1,000-item studies or greater may be possible with Bandit MaxDiff, given a large enough sample size (see Appendix C). Such studies would be entirely impractical and prohibitively expensive using standard MaxDiff. The reader might wonder how it is possible that researchers could devise a MaxDiff experiment involving 1,000 or more items. This is quite possible if the items are actually profiles (or pictures) made up of combinations of multiple attributes, where strong interactions are concerned (say, for aesthetic package design research).

Analysis Options for Bandit MaxDiff

If you are studying relatively few items (say 30 or fewer), Bandit MaxDiff can show every item at least once for every respondent and the most preferred items could be shown four or more times. In such cases, HB, latent class, or aggregate logit will all work quite well.

If you are studying a large number of items (say, 60 or more), Bandit MaxDiff typically will not carry all items into each respondent's MaxDiff questionnaire. A subset of, say, 30 items could be chosen for each respondent. Due to how sparse the data could become for any one respondent, pooled analysis such as aggregate logit would be effective.

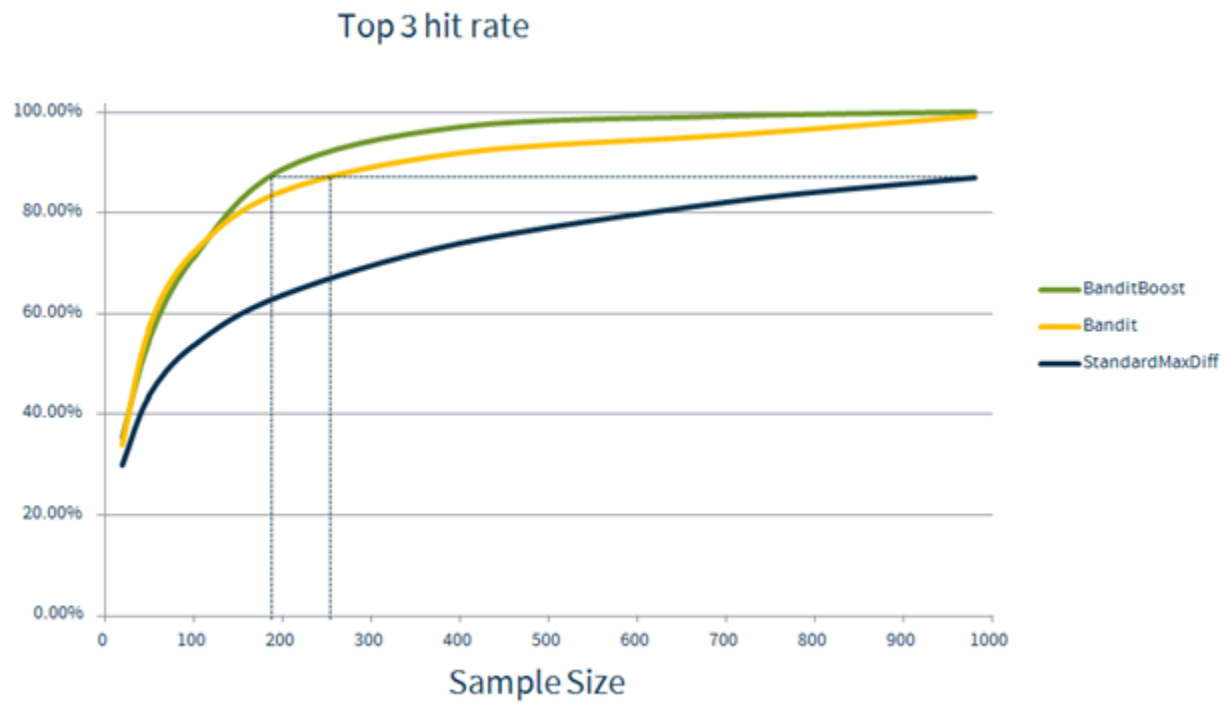
Can Initial Non-Representative Respondents Foil Bandit MaxDiff?

What if the first respondents are not very representative of the sample? After all, early responders are sometimes different from later responders. Could they throw Bandit MaxDiff off the scent such that the adaptive learning actually does more harm than good? We conducted hundreds of simulations based on patterns of real respondent preferences to examine this issue (Fairchild et al. 2015). For each of the simulations, we purposefully rearranged the preferences for the first 50 respondents so that it appeared that they believed *nearly the opposite* of their true preferences for the most preferred items. This is

much worse than we'd ever expect to see in practice and indeed for the first 200 respondents the overall results (pooled estimation) were worse than if standard MaxDiff were employed. But, after the first 50 misinformed respondents plus another 150 to 200 rationally-behaved respondents, the results for Bandit MaxDiff were equally good as standard MaxDiff without the misinformed early responders. So that Bandit MaxDiff performs well even with non-representative first responders, some of the items selected for each respondent should use a more diffuse variance (or be selected purely randomly). Our Lighthouse Studio implementation of Bandit MaxDiff does this automatically and allows you to adjust the aggressiveness of the exploitation strategy to be less aggressive or more aggressive.

Boosted Bandit MaxDiff: A Power Trick to Improve Performance

Using a power trick that is not very difficult to implement, you can improve results even more for Bandit MaxDiff. An oversampling boost to the topmost items relative to the other items achieves even greater efficiency than reported earlier (Fairchild *et al.* 2015) for identifying the most preferred items across respondents. Simulation results for 120 items using Lighthouse Studio's built-in Bandit MaxDiff procedure are shown below, using the same robotic respondents as reported by Fairchild et al. 2015:



With about 200 respondents, Boosted Bandit MaxDiff can achieve the same hit rate (for predicting the top three items out of 120 for the population) as standard MaxDiff can with about 1,000 respondents. With about 250 respondents, Bandit MaxDiff matches what standard MaxDiff can do with about 1,000 respondents.

The idea behind Boosted Bandit MaxDiff is quite simple: we modify the experimental design such that the top few (Thompson-sampled) items appear more times *within each* respondent than the next few. See Appendix B for more details.

Segment-Based Bandit MaxDiff

The Bandit MaxDiff strategy assumes your main interest is to identify the most preferred items for the overall population. Rather, if your goal was to identify the most preferred items for targetable segments of the population, then you should use a segmented and tailored Bandit MaxDiff approach.

Let's imagine that previous research had identified six known and targetable segments (such as by geography, income, age, risk tolerance, brand preference, or some combination of such traits). Upfront, we could ask a few questions needed to assign respondents into one of the six segments. Six identical Bandit MaxDiff exercises could be programmed within the Lighthouse Studio questionnaire and respondents would be skipped into the exercise prepared for their segment. With Lighthouse Studio's implementation of Bandit MaxDiff, means and variances are estimated only using the data within each MaxDiff exercise, so the Thompson sampling procedure would be customized for each respondent group. The data later could be combined for analysis, for example, using HB analysis with the six-group variable as a covariate (assuming the data were not very sparse). Or, if the data were particularly sparse, aggregate logit could be run separately within each respondent group.

How many respondents per group would be needed for segment-based Bandit MaxDiff to become a viable approach? Based on our simulation results when isolating small sample sizes, our opinion is that if a) preferences indeed were meaningfully different between the groups, b) the researcher intended to analyze the results separately by the groups, and c) early respondents were not very different from later respondents, then about 90 respondents per group would be sufficient to make segment-based Bandit MaxDiff more effective than non-segmented Bandit MaxDiff.

Can I Confidently Embrace Bandit MaxDiff?

Even though the theory and statistics for solving multi-armed bandit problems have existed for decades, applying them to MaxDiff is new. We've conducted a large number of simulations based on realistic patterns from real respondent data involving 120 items, 300 items, and 1,000 items. We've also conducted a real study involving 80 items in cooperation with P&G. The results have been consistent and impressive. Our colleagues at SKIM Group also have programmed their own customized version of Bandit MaxDiff (using counting analysis and Gamma draws) and have conducted seven Bandit MaxDiff studies for clients with success. These studies have featured as many as 153 items.

References

Cohen, Steve (2003), "Maximum Difference Scaling: Improved Measures of Importance and Preference for Segmentation," 2003 Sawtooth Software Conference Proceedings, Sequim, WA.

Fairchild, Kenneth, Bryan Orme, and Eric Schwartz (2015), "Bandit Adaptive MaxDiff Designs for Huge Number of Items," 2015 Sawtooth Software Conference Proceedings, pp. 105-118.

Appendix A:

Boosted Bandit MaxDiff
When All Items Are Shown to Every Respondent

Bandit MaxDiff may be used advantageously even for MaxDiff problems involving very few items. For example, the experiment could involve just twelve items where all respondents see all twelve items in their MaxDiff questionnaires and the prior most preferred items (as judged by previous respondents) are oversampled in the design. We describe how this could be done using Lighthouse Studio below.

Recall that for each respondent, Thompson sampling is used to select the items to show the respondent based on the prior means and variances. For the purposes of studies involving very few items, all items are selected for each respondent, but Bandit MaxDiff ensures that the most preferred items are oversampled. The key to making this happen hinges on the fact that the constructed list command that implements Thompson sampling in Lighthouse Studio sorts the draws for the items from best to worst.

First, decide to what degree you would like to oversample the most preferred items. A reasonable scheme involving 12 items is to show the top three (Thompson-drawn) items for each respondent 2x as many times as the 9^{th} through 12^{th} most preferred (Thompson-drawn) items.

Below, we describe the steps in Lighthouse Studio for implementing a within-respondent Boosted Bandit MaxDiff study involving just 12 items:

1. Create a predefined list that includes a few replicated items to accomplish the oversampling scheme above. Specify a predefined list to use in the MaxDiff exercise that includes 15 total items in the list (that we'll recode back to the original 12 items in a later step). The 15 elements in that list are as follows (one row per element):

   1
   1 <replicate>
   2
   2 <replicate>
   3
   3 <replicate>
   4
   5
   6
   7
   8
   9
   10

11
12

2. Generate a standard MaxDiff experimental design with multiple versions where the total number of items is 15. Following typical practice, you might decide to use 10 sets per respondent showing 4 items per set such that each item appears on average 2.67x per respondent. Prohibit the replicated items from showing with each other within the same sets (for example, items 1-2, items 3-4, and items 5-6 above). The default is 300 versions, but just 10 would work extremely well (hardly any loss of precision due to having 10 versions rather than 300).
3. Export the 15-item design to a .CSV file using the **Export…** button on the *Design* tab. Open that .CSV file with Excel and modify it to recode levels 1 and 2 to 1; recode levels 3 and 4 to 2, etc. Now you have recoded all item indices to the original 12 items, but items 1, 2 and 3 are now represented 2x as many times in the design as before. In the new design, the top 3 Thompson-drawn items will now appear on average 5.33 times per respondent across the 10 tasks. The bottom 9 Thompson-drawn items will now appear on average 2.67 times per respondent across the 10 tasks.
4. Modify the pre-defined list specified in the MaxDiff exercise to have only 12 items. (The software will complain that this will invalidate the previous design. This is OK, since you haven't fielded the study yet, and you can ignore the warning.)
5. Using the **Import…** button from the *Design* tab, import the modified .CSV design file. Run *Test Design* to make sure the level counts are as expected across all versions in your questionnaire (item 1 should appear 2x as often as level 12, etc.)
6. Create a constructed list using the Bandit MaxDiff constructed list instruction (as described in the Lighthouse Studio documentation). Specify that all 12 items should be selected for each respondent.
7. In the MaxDiff exercise, specify that the exercise should use the constructed list created in step 6 rather than the predefined list.

The reason this procedure works is that the Thompson Sampling constructed list instruction selects items to include on the constructed list in priority (best to worst order) according to the draws from the prior preferences. So, the probable best item is assigned to the first list element—and that first list element has been oversampled in your design.

Note: if an item appears in too many of a respondent's MaxDiff tasks, this could become distracting or annoying. For example, the same item appearing in 2/3 or more of the MaxDiff tasks should probably be avoided.

Appendix B:

Boosted Bandit MaxDiff
When Respondents See a Subset of All Items

Very similar to the strategy outlined in Appendix A, we can oversample most preferred items within a respondent to improve bandit MaxDiff results (achieving even better results than reported by Fairchild et al. 2015). For example, imagine a situation in which you are studying 100 total items. You decide to use Bandit MaxDiff with Thompson sampling to select only 30 items to show any one respondent. And, you decide to use 18 sets, with 5 items per set. With a standard MaxDiff design, each of those 30 items would appear 3 times across the planned 18 sets. However, we can do a power trick within the bandit sampling scheme to further boost and oversample the very top few items.

- 1st through 6th best Thompson-sampled items *<show 5x per respondent>*
- 7th through 30th best Thompson-sampled items *<show 2.5x per respondent>*

Steps for doing this in Lighthouse Studio are shown below:

1. Create a predefined list that includes a few replicated items to accomplish the oversampling scheme above. Specify a predefined list to use in the MaxDiff exercise that includes 36 total items in the list (that we'll recode back to 30 items in a later step). The 36 elements in that list are as follows (one row per element):

   1
   1 <replicate>
   2
   2 <replicate>
   3
   3 <replicate>
   4
   4 <replicate>
   5
   5 <replicate>
   6
   6 <replicate>
   7
   8
   9
   10
   11

12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

2. Generate a standard MaxDiff experimental design with multiple versions where the total number of items is 36. For this example, we've decided to use 18 sets per respondent showing 5 items per set such that each item appears 2.5x per respondent. Prohibit the replicated items from showing with each other within the same sets (for example, items 1-2, items 3-4, and items 5-6, etc. above). The default is 300 versions, but just 10 would work extremely well (hardly any loss of precision due to having 10 versions rather than 300).
3. Export the 36-item design to a .CSV file using the **Export…** button on the *Design* tab. Open that .CSV file with Excel and modify it to recode levels 1 and 2 to 1; recode levels 3 and 4 to 2, etc. Now you have recoded all item indices to 30 items, but items 1 through 6 are now represented 2x as many times in the design as were before. The original design included each of the items 2.5x, so the Thompson-sampled best item six items are now included 5x per respondent. (Note that due to the random nature of the Thompson sampling draws, this item can be different between respondents.)
4. Modify the pre-defined list specified in the MaxDiff exercise to have only 30 items. (The software will complain that this will invalidate the previous design. This is OK, since you haven't fielded the study yet, and you can ignore the warning.)
5. Using the **Import…** button from the *Design* tab, import the modified .CSV design file. Run *Test Design* to make sure the level counts are as expected across all versions in your questionnaire (item 1 should appear 2x as often as level 7, etc.)
6. Create a constructed list using the Bandit MaxDiff constructed list instruction (as described in the Lighthouse Studio documentation). Specify that 30 items should be selected for each respondent.

7. In the MaxDiff exercise, specify that the exercise should use the constructed list created in step 6 rather than the predefined list.

The reason this procedure works is that the Thompson Sampling constructed list instruction selects items to include on the constructed list in priority (best to worst order) according to the draws from the prior preferences. So, the probable best item is assigned to the first list element—and that first list element has been oversampled in your design.

Note: if your goal is to identify the top five items for the sample, we recommend a within-respondent oversampling boost on about the top seven items. If the goal is to identify the top ten items for the sample, we recommend an oversampling boost on about the top twelve items, etc.

Appendix C:

Sample Size for Bandit MaxDiff Studies

This section provides guidance regarding sample size needed to achieve 90% correct classification of the top few items out of many under Bandit MaxDiff. The first results below are based on a simulation study conducted by Fairchild, Orme, & Schwartz (2015 Sawtooth Software Conference) that generated simulated respondents based on actual HB utilities from a real MaxDiff study conducted by P&G.

In our 2015 paper, Fairchild et al. conducted simulations with bootstrap sampling to compute the sample size needed for 90% correct classification via pooled multinomial logit estimation of top-3 and top-10 items for studies with 120 and 300 items, where each robotic respondent completed 18 MaxDiff sets showing 5 items per set:

	Sample Size to Achieve Top-3 Hit Rate of 90%	Sample Size to Achieve Top-10 Hit Rate of 90%	Row Average
120 items	250	160	205
300 items	1,000	1,050	1,025

The 300-item study was generated based on the patterns of preferences and variances found in the original 120-item study, so the results should be fairly comparable.

Later, with the help of Zachary Anderson, we conducted simulations (13 separate replications, using different random seeds) for a 1,000-item study, based on the same variances and patterns of preference correlation seen in the original 120-item study collected by P&G. On average, it took 8,500 respondents to achieve a 90% hit rate (correct classification of the top-30 items) and 2,500 respondents to achieve an 80% hit rate. The true top item had a 3% higher likelihood of choice than the true second-place item, and we were very pleased at how well Bandit MaxDiff with robotic respondents was able to classify that top-ranked item correctly out of 1,000 items (almost like finding a needle in a haystack). With just 1,000 robotic respondents and again using pooled logit estimation, it identified the top true item out of 1,000 items in 7 out 13 of our simulations. With 2,000 robotic respondents Bandit MaxDiff identified the top true item in 10 of 13 simulations (and the three simulations that didn't correctly identify the top item ranked it in second place). With 5,000 robotic respondents, it found the top item in 12 out of 13 simulations (and the top item came in second place for that one miss).

Is 90% correct classification rate for the top few items really needed for studies involving hundreds of items? Perhaps for a 500-item study, obtaining an 80% classification rate of the top true few items would be sufficient. Even if an item was incorrectly classified among the top 10 out of 500, the Bandit MaxDiff methodology is robust enough that items misclassified into the top 10 should still be very high quality and very near the top 10.