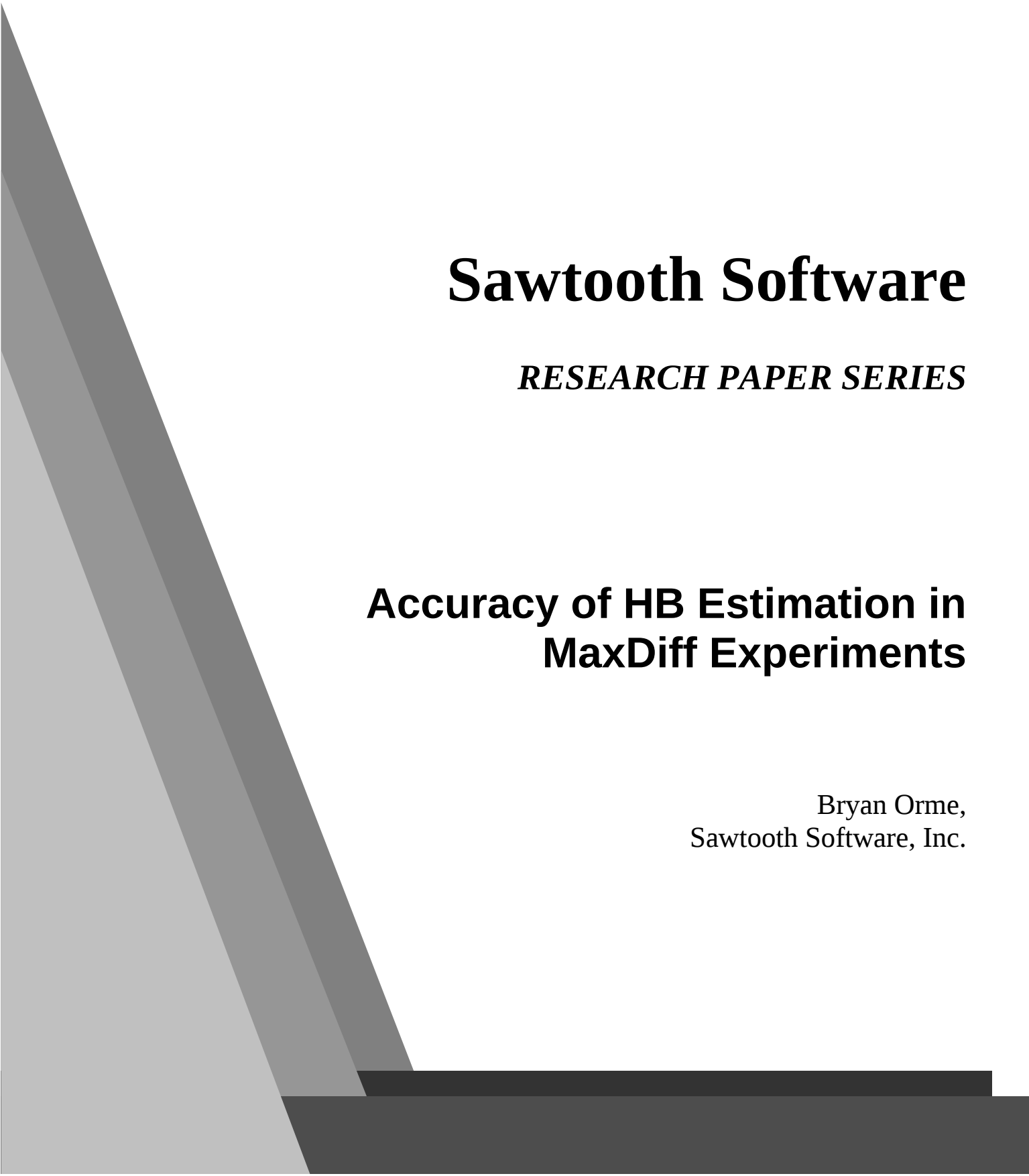




Sawtooth Software

RESEARCH PAPER SERIES

Accuracy of HB Estimation in MaxDiff Experiments

Bryan Orme,
Sawtooth Software, Inc.

Accuracy of HB Estimation in MaxDiff Experiments ¹

Bryan Orme, Sawtooth Software
Copyright 2005, Sawtooth Software, Inc.

Background

MaxDiff (Best/Worst) scaling has received a great deal of interest lately. For this article, I'll assume that the reader is already familiar with MaxDiff. For a review of MaxDiff, I'd suggest an excellent introductory paper written by Steve Cohen, entitled "Maximum Difference Scaling: Improved Measures of Importance and Preference for Segmentation," available in our Technical Papers library at www.sawtoothsoftware.com.

The "Server" Study

I was fortunate to have worked closely with Steve Cohen on the methodological study reported in his paper. That experience was my first introduction to MaxDiff scaling. In that study², we interviewed 116 respondents using MaxDiff regarding 20 performance attributes for servers (computing technology). We asked respondents to complete 15 best/worst tasks, each defined on 4 attributes. For each task, respondents indicated which of the 4 attributes (items) was the most important and the least important. Across all 15 tasks times 4 items per task, each of the 20 attributes was displayed 3 times to each respondent. Having each attribute shown 3 times per respondent seemed about right, but this paper more formally investigates that point.

We additionally asked respondents to rank sets of 3 three items, which we held out for internal validation. We repeated the holdout tasks, for an assessment of test-retest reliability within each respondent. The test-retest reliability for holdout tasks was 81%. We used HB estimation (using Sawtooth Software's CBC/HB system) to derive utility scores for each respondent, for all 20 items. The utilities derived using HB were able to predict 78% percent of the holdouts correctly.

How good was this result? As the minimum expectation for success, we should see HB perform at least as well (for predicting individual-level holdout judgments) as when using average utilities from the population. Assigning average population parameters to each individual yields a minimum expected hit rate of 58% for the server study. What about the maximum expectation for success? Dick Wittink and Rich Johnson demonstrated that the maximum expected hit rate for predicting a fallible criterion measure is equal to:

¹ I have borrowed both the title and inspiration for this paper from Rich Johnson's Monte Carlo simulation study done for ACA in 1987, entitled "Accuracy of Utility Estimation in ACA." That paper is available within the Technical Papers library at www.sawtoothsoftware.com.

² We are grateful to Michael Patterson (formerly of HP) for sponsoring the "server" study and allowing us to share the results.

$$\pi = \frac{1 + \sqrt{(2p - 1)}}{2}$$

where π is the maximum expected hit rate and p is the agreement between independent replications of the criterion measure. Given test-retest reliability of 81%, the maximum possible hit rate for predicting the holdout choices is 89%. The predictive rate we observed (78%) for HB utilities was 65% the distance from the lowest expectation (58%) obtained from aggregate parameters toward the best theoretical prediction rate (89%), given the observed test-retest reliability. We'll use this metric as the relative indication of accuracy throughout the remainder of this article.

Out of curiosity, I investigated how the quality of estimation for the server study degraded as I threw away data. Each respondent had completed 15 best/worst tasks. What would happen if we used fewer tasks, such as the respondents' first 12 tasks? The absolute and relative hit rates are displayed in Table 1.

Table 1
Predictive Accuracy for Server Study

	Hit Rate	*Relative Hit Rate
15 tasks (each item shown 3x)	78.3	65%
12 tasks (each item shown 2.4x)	77.3	62%
10 tasks (each item shown 2x)	76.9	61%
8 tasks (each item shown 1.6x)	74.2	52%
6 tasks (each item shown 1.2x)	70.8	41%
4 tasks (each item shown 0.8x)	69.3	36%

*Percent of distance between theoretical minimum and maximum hit rate.

The remarkable thing about HB is that it can estimate a full set of parameters, even if the data are very sparse for each person. For example, even when some of the items were *never* shown to individuals (4 tasks allowed at maximum only up to 16 of the 20 items to be displayed), HB borrows information from other respondents to estimate reasonable parameters. The estimates using just 4 tasks are one-third higher (relative to lowest and highest theoretical bounds) than if using aggregate means (estimated from all 15 tasks) and all respondents to predict each individual's holdout choices.

I will not discuss the use of aggregate logit or latent class to estimate parameters for MaxDiff experiments in this article. However, these techniques are commonly used in MaxDiff estimation, and might be particularly useful when the data are especially sparse.

Simulation Study

Data Generation

The previous section investigated how the number of tasks affects the accuracy of prediction when using HB for MaxDiff studies. There are two other decisions to make when designing MaxDiff projects: how many items to display within each task, and how the number of total items in the study affects the data requirements. We cannot learn about these questions from the data at hand—but we can use simulated data.

Two major questions when constructing simulated data are the size of the parameters (an indication of the certainty of respondent judgments) and the heterogeneity across respondents. Rather than making naïve assumptions, I referred to the actual data from the server study to obtain reasonable estimates. The average population parameters (HB, with a logit model at the “lower level”) ranged from about -3.54 to +3.64. The standard deviations of the point estimates across individuals ranged from a minimum of 1.2 to a maximum 3.1. On average, the standard deviations for the estimated parameters were 2.05. Based on this information, I generated 300 simulated respondents with average means ranging from -3.5 to +3.5 (where the parameters were distributed evenly across the range), with a standard deviation for each parameter of 2.0. I then gave these computerized “respondents” various MaxDiff questionnaires (described further below), and projected that respondents would make judgments of “best” and “worst” according to their true utilities, after adding Gumbel distributed error to the utility of each item. (Gumbel error is the error assumed by Multinomial Logit models to develop utilities with maximum likelihood fit to observed choices.)

For each questionnaire condition, I used CBC/HB software (version 3) to estimate utilities for each of the 300 simulated respondents. I employed dummy coding and Peter Lenk’s suggestions (as documented in CBC/HB software) regarding the prior covariance matrix. To check my work, I verified that the posterior means and standard deviations very closely matched the characteristics I assumed when constructing the computerized respondents.

I generated a single set of holdout tasks (three items at a time, fully-ranked) for validation of all the questionnaires studied. Each simulated respondent completed 30 such holdout tasks. As with the server study, hit rates were computed with respect to predicting all implied paired comparisons from the full ranking.

Study Design

I varied the following questionnaire design elements:

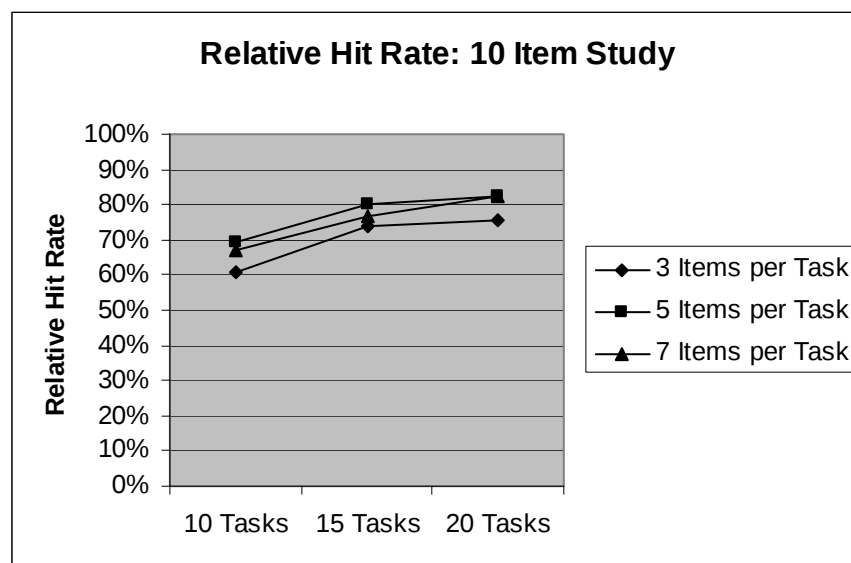
- Number of items in the study: 10, 20, 30
- Number of items displayed per task: 3, 5, 7
- Number of tasks in the questionnaire: 10, 20, 30

All combinations of these treatments resulted in 27 separate questionnaires, with accompanying data sets, and HB runs. The experimental designs for the questionnaires were generated using Sawtooth Software's Best/Worst Experiment Designer. This software generates MaxDiff plans with level balance, orthogonality, and positional balance. I maintained the number of versions (blocks) of the questionnaire constant at 4, and distributed questionnaire versions evenly across simulated respondents.

Results

Relative hit rate accuracy for 10, 20, and 30 items in the study is displayed in exhibits 1, 2, and 3, respectively.

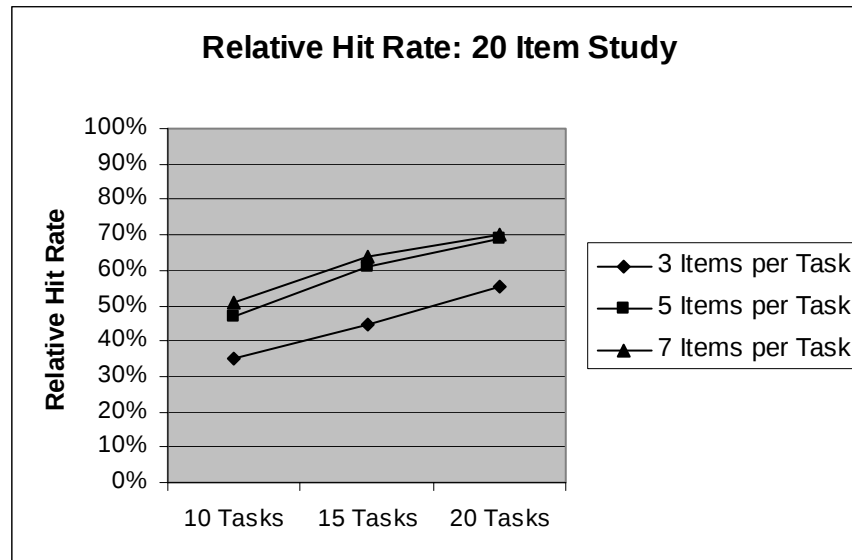
Exhibit 1



Relative hit rates were quite good for all 9 questionnaires used to study 10 items. In every case, results were closer to the theoretical maximum than to the theoretical minimum. In the least informative case (10 tasks, displaying 3 items per task), each item is shown 3 times per respondent. In the most informative case (20 tasks, displaying 7 items per task) each item is shown 14 times per respondent.

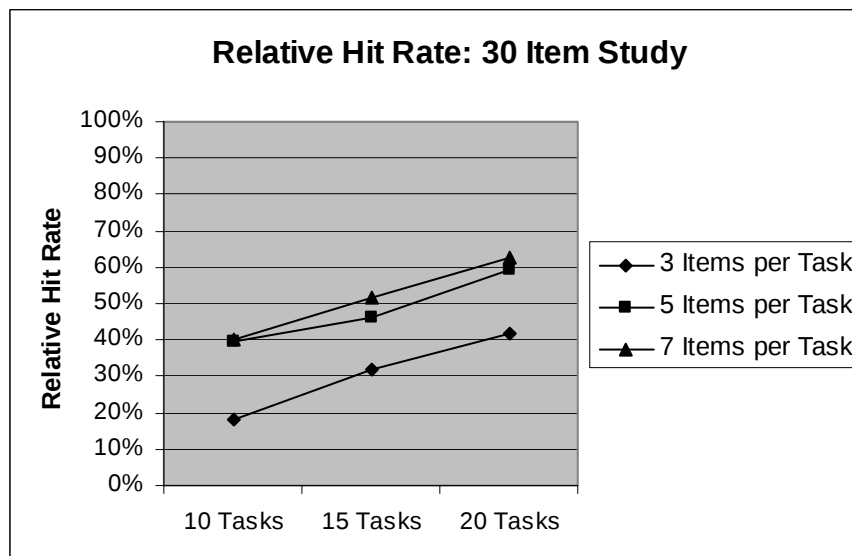
As expected, increasing the number of tasks increases the relative hit rate, with noticeably diminishing returns. However, the potential surprise in Exhibit 1 is that the relative hit rate actually *decreases* when displaying 7 items instead of 5 at a time within each task. To explain this result, consider a MaxDiff study of 10 items where we display all 10 items in each task. For each respondent, we'd certainly learn which item was best and which was worst, but we'd learn little else about the items of middle importance for each individual. Thus, increasing the number of items per set eventually results in lower precision for items of middle importance or preference. This leads to the suggestion that one include no more than about half as many items per task as being studied.

Exhibit 2



With a 20 item study, the gains in predictive accuracy are essentially linear out to 20 tasks. There is virtually no difference between using 5 items per task instead of 7 items. Given that computerized respondents are not fatigued or confused by more items, this certainly calls into question whether we should use more than 5 items at a time with real respondents. For the 20 item study, relative hit rates at least half-way between the lowest and highest theoretical bounds can be achieved when each item is shown at least 3 times to each respondent.

Exhibit 3



As expected, the lowest relative hit rates are achieved in the 30 item study (there are more parameters to estimate). As with the 20 item study, the gains in predictive accuracy are essentially linear out to 20 tasks, and there seems to be little value in displaying more

than 5 items per task. In the least informative case (10 tasks, 3 items per task), each item is shown just once per respondent. With 20 tasks, displaying 5 items per task, each item is shown 3.33 times per respondent, and the relative hit rate is 60%. When studying 30 items, there would seem to be clear benefit in asking 20 tasks or more. In fact, if respondents do not fatigue, we would probably want to ask them to complete additional tasks to obtain better estimates for the 30 items.

Overall Effect of MaxDiff Design Decisions

Because the experimental plan reflects a full-factorial ($3 \times 3 \times 3 = 27$ runs), we can summarize and separate the effect of each element of the design on the quality of the results. To do so, we simply average the relative hit rates across the 9 runs that involved each treatment. Exhibit 4 displays the results:

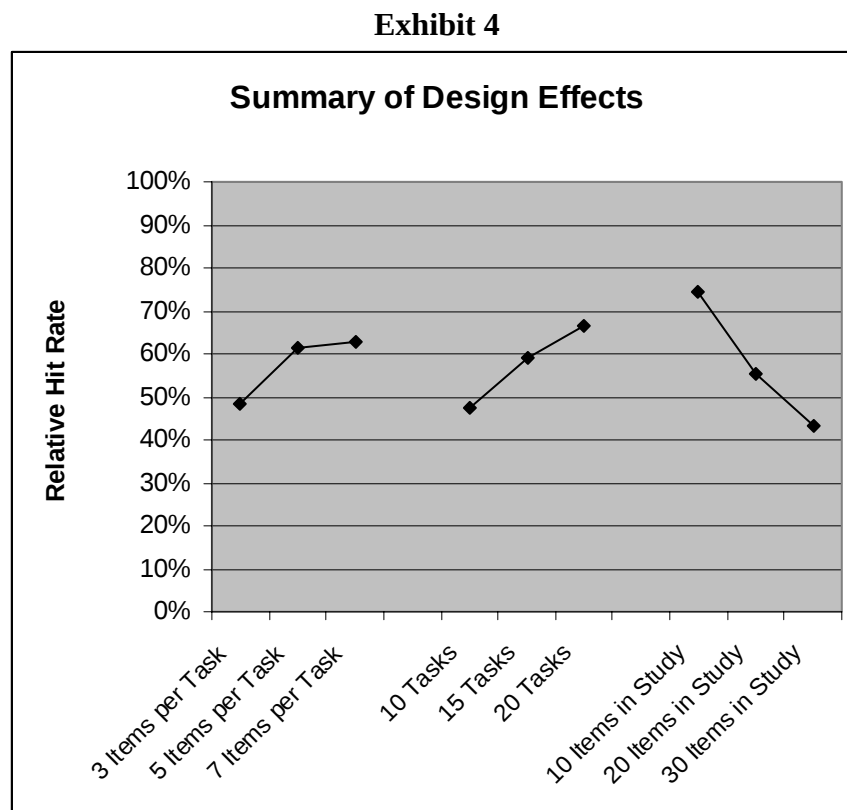


Exhibit four leads to the following conclusions:

- Over the ranges studied, the number of items in the study has the greatest effect on accuracy of results
- The smallest effect is the number of items displayed per task
- One gains very little by showing more than five items per task
- Gains from increasing the number of tasks are nearly linear, at least to 20 tasks.

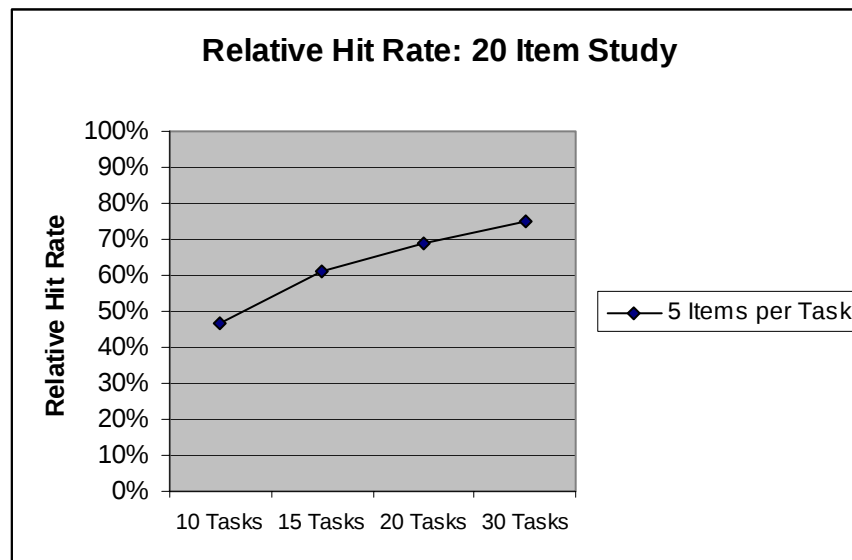
General Recommendations

Based on this simulation study, I'd recommend the following:

- Display either four or five items per MaxDiff task. More than five provides little incremental gain.
- For relatively accurate individual-level HB estimates, make sure each item is displayed three or more times for each respondent.
- Accuracy exceeding the aggregate solution may be obtained by showing each item to each respondent just once (or even fewer times, with a blocked plan!).
- For studies involving a dozen or more items, if respondents have the energy to complete 20 tasks or more, the incremental gains in accuracy seem worth having them do so. If the number of items is relatively small, such as 10 or less, then it may make sense to stop at around 15 tasks.

One may wonder what to do when studying a moderately large number of items, such as 20. The range of questionnaire length in this experimental plan did not reveal a point of much diminishing return. Out of curiosity, I preformed an additional run: 20 items, 5 items per task, for *30 tasks*. I've added that result to the previous curve (as reported in Exhibit 2) in Exhibit 5.

Exhibit 5



For computerized respondents, we see only slight diminishing returns out to 30 tasks when studying 20 items. Of course, actual respondents will fatigue, so the findings from simulated respondents should be interpreted with caution. I'd also caution against generalizing any of the other results much beyond the ranges studied here.