Submitted to
manuscript (Please, provide the manuscript number!)

Bandit MaxDiff: Large-scale adaptive MaxDiff with ϵ-Diffuse Thompson Sampling PRELIMINARY! DO NOT DISTRIBUTE

Eric Schwartz
University of Michigan, ericmsch@umich.edu,

Kenneth Fairchild
Sawtooth Software, ,

Bryan Orme
Sawtooth Software, ,

Alexander Zaitzeff
University of Michigan, azaitzef@umich.edu,

For large MaxDiff studies whose main purpose is identifying the top few items for the sample, a new adaptive approach called Bandit MaxDiff may increase efficiency fourfold over standard non-adaptive MaxDiff. Bandit MaxDiff leverages information from previous respondents via aggregate logit and Thompson Sampling so later respondents receive designs that oversample the topmost items that are most likely to turn out to be the overall winners. Our approach applies beyond MaxDiff problems to a more general set of bandit problems. We propose a flexible algorithm, ϵ-Diffuse Thompson Sampling (TS), which nests traditional TS. Blending ideas from ϵ-greedy in machine learning and Bayesian approaches, this is a more robust version of TS, which a manager can control with tuning parameters. For instance, being less risk averse, one may drawn more items from diffuse posteriors, making the algorithm robust to changing environments, even extreme non-stationarity. We implement the methods using MaxDiff survey from a large consumer packaged goods manufacturer. Beyond showing our approach outperforms better than current methods, we show under which conditions it performs even better than (larger problems) or just as well as existing methods (smaller problems).

Key words: Active learning, adaptive conjoint, multi-armed bandit, best-worst scaling, MaxDiff, Thompson Sampling

1. Introduction

Firms collect more customer data than ever. While much of that is behavioral data (e.g., purchases, clicks, visits, etc.), the online market research relies on surveys (e.g., preference measurement) at an increasingly large scale. And as firms push preference measurement methods into a larger scale, market researchers seek to collect data as efficiently.

One common managerial decision driving such data collection is to select which are most preferred among a large set of hundreds of possible items. For instance, consider these illustrative managerial decisions: select which features should we offer in a new product; select the most preferred levels of a product attribute; and select the consumer benefits we should emphasize in marketing communications. For each problem, a market researcher may have to screen out hundreds of “items” (e.g., features, benefits, messages).

But typical choice experiment methods, such as conjoint, are infeasible for cost or simply inefficient at this scale. Instead, researchers often use pretest or screening phases to select a smaller set, and use that for the main analysis. Yet there is no systematic method to smoothly transition out of that screening phase, even when the ultimate objective is learning precisely the set of most preferred items.

Choice experiment methods for preference measurement continue to be among the most widely adopted market research methods. Choice-based conjoint (CBC) analysis and maximum difference (MaxDiff), for instance, are common choice tasks that continue to be used in both academic and industry settings.

Methodological advances, such as adaptive conjoint, have long sought to improve efficiency, obtaining more information from fewer respondents. Those methods will use past responses to select the next question to improve precision of all parameters. For instance, adaptive choice-based conjoint (ACBC) methods present respondents with questions to reduce the uncertainty where it is greatest. It chooses the next question to best reduce overall uncertainty.

But the market researcher’s goal is typically not to obtain precise estimates of utility of every single alternative or attribute level. If their goal is to identify the most preferred items, why do they need to spend resources (questions, respondents, time) to learn how precisely poor the least-preferred item is compared to the second-least preferred item?

With that objective in mind, we propose a new adaptive choice experiment method. Our approach departs from traditional adaptive survey methods like adaptive conjoint analysis

(ACA). They aim to improve the precision of each of the parameters; we estimate the best items more precisely.

We propose a new adaptive method, combining MaxDiff and multi-armed bandit methods, as an example of an even broader framework for stated choice data collection. The focus of our research is a broader framework to align the data collection process with such managerial objectives. By aligning the two, we aim to improve efficiency and scalability for big data collection via choice experiments. We find we can "get more for less," by improving precision using a smaller sample size compared to existing methods.

We propose this framework more broadly but will demonstrate it with MaxDiff, as it is an increasingly popular and important choice experiment method. But we will consider the general choice-based conjoint case in our discussion section.

Our approach draws on state-of-the-art MAB algorithms in statistical machine learning called Follow The Perturbed Leader (FTPL) Kalai and Vempala (2005). Recent theoretical advances prove that adding perturbations of Gumbel distributed noise leads to an optimal MAB strategy Abernethy et al. (2015); Kujala and Elomaa (2005). The perturbation methods provide an intuitive randomization decision strategy, which is appropriate for our application, yet their connection to other approaches and applications are limited. Perturbation not only resolves the explore-exploit (learn-earn) tradeoff for the stochastic (iid) bandit setting, but it also does so in the adversarial setting, suggesting it is an MAB strategy more robust to changes in the non-random changes in the environment. We shed light on this methodological link as it has consequences beyond our method.

We contribute three ideas to the literature and practice. First, we frame choice task data collection as an earning-and-learning problem. During the process of data collection, we simultaneously balance the desire to learn preferences of all items and earn a reward towards achieving objective, by only serving questions the truly best subset of items. In this case, the objective is to identify the best set of items, for instance, the most desirable levels of attributes. The faster we identify the best items, the more we include them in questions, improving the precision of estimates. And if we need to reach a desired level of precision, we can do so faster, saving money in the form of number of respondents and time. Multi-armed bandit problems appear in marketing experiments for online advertising Schwartz et al. (2015), Urban et al. (2013) and website design Hauser et al. (2009). One particular MAB method amenable to multi-variable parametric models is

Thompson Sampling.

While we use bandit algorithms, our approach differs from extant bandit applications in a number of ways. For a bandit problem the reward is observed immediately (click, acquisition, purchase). While we have immediate observations (choice), that is not our reward. Instead, our goal is to correctly identify the truly best items as precisely as possible. These are not evaluable in real-time.

While our problem is not a bandit problem exactly, it is a bandit-inspired problem. However we find bandit algorithms form a heuristic to achieve our objective better than existing MaxDiff and conjoint methods.

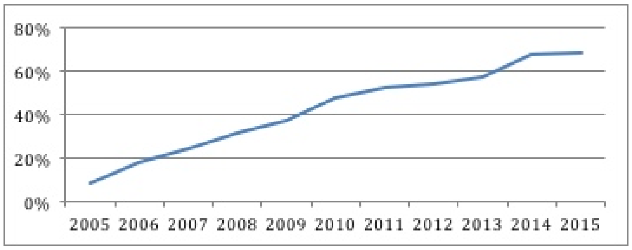
Consider what would occur if the reward was whether an item is selected as best alternative, the arm's reward would be completely dependent on all of the other items presented in the set (e.g., was it presented alongside poor items). Second, the need to optimize this process is greater for large-scale problems with extremely large number of possible attribute levels or items. In settings with hundreds of items, there is greater opportunity cost of focusing on items that are not important, calling for a need to improve efficiency.

Third, we propose an adaptive method for best-worst scaling. Existing adaptive methods have been largely limited to conjoint, adapting at both the aggregate level, Arora and Huber (2001), and the individual level, Toubia et al. (2004). Yet best-worst methods, such as MaxDiff, have continued to emerge as important and commonly used in areas including marketing research and public health. We introduce a first step to make MaxDiff adaptive in a principled manner by using MAB methods. We accommodate an increasingly accepted method of solving multi-armed bandit problems, Thompson Sampling. We introduce two versions: MaxDiff-Thompson Sampling and its generalization MaxDiff ϵ-diffuse Thompson Sampling.

The ϵ-diffuse Thompson Sampling nests traditional TS. We combine ideas from ϵ-greedy in machine learning and Bayesian approaches to produce a more robust version of TS, which a manager can control with tuning parameters. For instance, if a manager wants to be extra conservative and maintaining more samples drawn from diffuse posteriors, she will make the algorithm robust to changing environments, even extreme non-stationarity.

The rest of the paper is structured as follows. In the next section we introduce the illustrative problem of maximum difference scaling focusing on its use in practice with large numbers of items. We then briefly introduce the multi-armed bandit framework. After

Figure 1 **MaxDiff is becoming more popular over time with Sawtooth Software users**



formalizing the discrete choice model of maxdiff analysis, we introduce our two algorithms that implement this method for Maxdiff Surveys. Then we present our empirical results in regular setting and then present variants of our main results. We implement the methods using MaxDiff survey implemented by Sawtooth Software with a large consumer packaged goods manufacturer.

2. Framework and literature

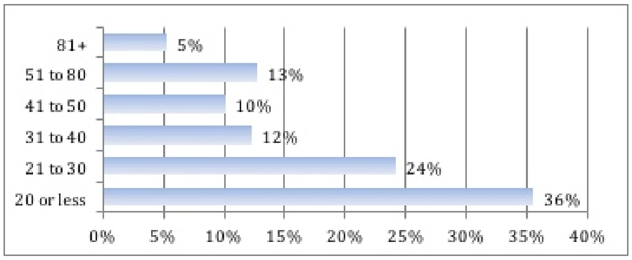
2.1. Maximum Difference Scaling

MaxDiff, Maximum Difference Scaling, is a preference measurement and item scaling method. In a MaxDiff questionnaire the researcher asks respondents for their best and worst item out of a set, then repeats this choice task. Initially proposed by Louviere and Woodworth (1991), MaxDiff was first released as a software system in 2004 by Sawtooth Software, a top marketing research software company in North America. Since its release its popularity increased steadily with penetration of the technique now reaching 68% of all Sawtooth users in 2015 (Figure 1).

MaxDiff offers benefits over alternative methods. MaxDiff provides more discrimination among items and between respondents on the items than traditional rating scales Cohen and Orme (2004). Besides enhanced discrimination, it avoids the scale use bias so problematic with traditional ratings scales (CITE).

MaxDiff, while it is a distinct type of choice experiment, is closely related to conjoint. A key difference is that it involves both best and worst choices instead of only one. But in its most common form, MaxDiff may be thought of as a one-attribute CBC study with many levels.

Figure 2 Maximum Number of Items Studied via MaxDiff over Last 12 Months



N = ??, Mean= 40, Median=30, Maximum=400

2.2. Studying More Items with MaxDiff: Survey of Market Research Practitioners

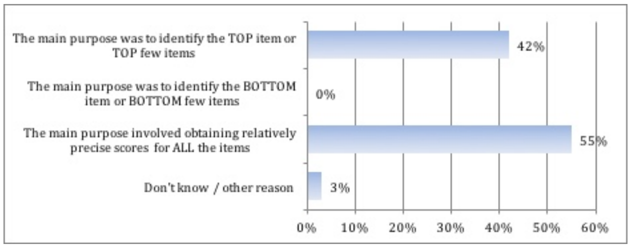
MaxDiff has proven so useful that market researchers increasingly find reasons to use MaxDiff for a large number of items. How many is a large number of items? In their 2007 paper, Hendrix and Drucker described “large sets” as about 40 to 60 items, proposing variants to MaxDiff called Augmented and Tailored MaxDiff to handle such large problems Hendrix and Drucker (2007). In their 2012 paper, Wirth and Wolfrath also investigated variants to MaxDiff called Express and Sparse MaxDiff for handling what they described as “very large sets” of items Wirth and Wolfrath (2012). Very large to these authors meant potentially more than 100 items. To support their findings, they conducted a study among synthetic robotic respondents with 120 items and a real study among humans with 60 items.

For Hendrix and Drucker 40 to 60 items was large, for Wirth and Wolfrath 120 items was very large. For this current paper, we’re referring to huge numbers of items as potentially 300 or more. The motivation of our research is beyond academic curiosity, as marketing researchers are seeking such applications pushing MaxDiff further than it was perhaps ever intended.

In Sawtooth Software’s 2015 Customer Feedback Survey, we asked respondents to tell us the largest number of items they had included in a MaxDiff study during the last 12 months (Figure 2). Nearly one-fifth of respondents indicated their firms had conducted a study with 51 or more items. The maximum number of items studied was 400!

To some it may seem bizarre and overwhelming that some researchers are conducting MaxDiff studies with 81+ or even 400 items. However, when we consider that individual MaxDiff items may actually represent conjoined elements that constitute a profile (say, a combination of packaging style, color, claims, and highlighted ingredients), then it can

Figure 3 Main Purpose for MaxDiff Study with 41+ Items



make much more sense to do 400-item studies. If the profiles involve multiple highly interactive attributes that pose challenges for CBC, then MaxDiff with huge numbers of items could be a viable alternative (given the new approach we demonstrate further below).

We also asked Sawtooth Software customers what the main purpose was for that study with the reported maximum number of items. For studies involving 41 or more items, the main reasons are displayed in Figure 3.

For 42% of these large MaxDiff studies, the main purpose was to identify the TOP item or TOP few items. Our research shows that if this is the main goal, then traditional design strategies are very wasteful. An adaptive approach using Thompson Sampling can be about 4x more efficient. Without the Thompson Sampling approach, you are potentially wasting 75 cents of every dollar you are spending on MaxDiff data collection.

The problem is that current MaxDiff approaches don't scale well to increasing the number of items. More items require commensurately longer questionnaires, larger sample sizes, and larger data collection costs with more tired respondents. If the researcher is concerned about obtaining robust individual-level estimates for all the items, then the current methodologies especially don't scale well to large lists of items. Respondents just tire out with such long surveys. In contrast, our approach employs an adaptive divide-and-conquer aggregate approach that leverages prior learning to create more efficient questionnaires and more precise aggregate score estimates.

2.3. Overview of approach: Bandit MaxDiff with Thompson Sampling

Thompson Sampling has been proposed as an efficient solution for solving the multi-armed bandit problem. Thompson Sampling involves allocating resources to an action in proportion to the probability that it is the best action Thompson (1933). Any bandit method must find an appropriate balance between exploring to gain information and exploiting that knowledge.

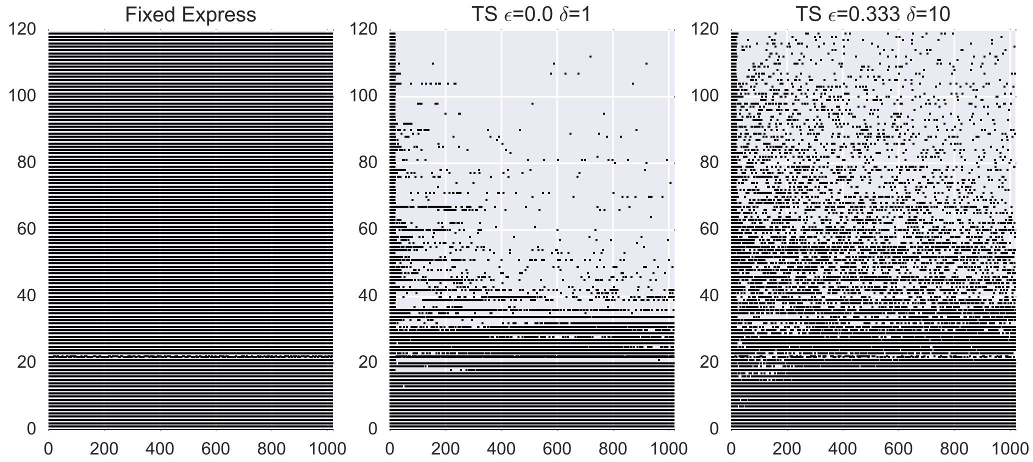
On the one hand, we want to learn about the relative scores of a large number of items within a MaxDiff problem. On the other hand, we want to utilize what we have learned so far to focus our efforts on a targeted set of actions that will likely yield greater precision regarding the items of most interest to the researcher. While there are many methods to accomplish this, Thompson Sampling has proven very useful for these types of problems. For a marketing application of Thompson Sampling and a review of the literature, see Schwartz et al. (2016).

The traditional MaxDiff design approach shows each item an equal number of times across all respondents x tasks. However, if the main goal is to identify the top few items for the sample, after the first, say 20, respondents it seems reasonable to start paying attention to the already-collected MaxDiff responses and oversampling the items that are already viewed as most preferred (the stars). We can use aggregate logit to estimate both preference scores and standard errors at any point during data collection (say, after the 20th, 40th, ... respondent has completed the survey).

Thompson Sampling makes a new draw from the vector of item preferences using the estimated population preferences (aggregate logit scores) plus normally distributed error, with standard deviations equal to the standard errors of the logit weights. As the sample size increases, the standard errors of course tighten.

To understand the logic of the algorithm, consider a snapshot in time during the data collection. Imagine we have just collected data from 100 respondents, and we decide to summarize their preferences (for each of 100+ items) with aggregate logit. Then, to generate a MaxDiff task for the 101st respondent, we could generate a draw from the population preferences leveraging the population means and normal errors with standard deviations equal to the empirically estimated standard errors. We then can sort that newly sampled vector of preference scores from the most to the least preferred item. The five most preferred items might be taken into the first task to show to the 101st respondent. The process (with or without updating the logit weights after recording the first task's answer) could be repeated to choose the five items to show in the second task for the 101st respondent, etc. To reduce the load on the server managing the data collection, perhaps only after every 20th respondent has completed the survey, the logit weights and standard errors would be

Figure 4 Respondent-by-item counts



updated.

We note that the translation of MaxDiff into a bandit problem is not obvious. Our goal is identifying the utilities of the top set of items with maximum precision. Selecting an arm corresponds to including it in the survey for the next respondent. The reward, however, is not as clear. On the one hand, suppose each task included all possible items. Then the reward would be clear: whether the item is chosen as the "best." But it is not feasible to show so many items to respondents, so that's not a relevant scenario. On the other hand, given we only select a subset of items, the item receiving the "best" label doesn't translate directly to a reward. Indirectly, that choice data does enable us to infer how it ranks among all items, exactly aligning with our goal. We use the alignment of the managerial goal and the MAB algorithm's balancing of learning and earning to our advantage.

3. Specifically Bandit + Conjoint = Thompsons + MaxDiff

We formalize the proposed procedure in three stages. First, we introduce MaxDiff as a best-worst scaling discrete choice task, and we relate it to the standard multinomial logit model. Next, we introduce Thompson Sampling as the natural method of resolving the learn-vs-earn tradeoff as in a multi-armed bandit problem. Then we generalize that Thompson Sampling algorithm to be more robust by sampling a proportion of items from a more diffuse distribution.

3.1. MaxDiff choice model

Every respondent selects both the best and the worst option from an available set of options in each discrete choice task. The model for that data comes from a class of probability models known as best-worst scaling, and MaxDiff is one such model. We adopt the framework from the best-worst scaling literature. For a review, see Marley and Pihlens (2012) Marley 2010, Marley and Louviere (2005). We have K possible items, and we select a set S for each choice task. To describe the items, we define two random variables, best B_z and worst W_z, for each $z \in S$. We then define a third random variable, best-worst $BW_{r,s}$, for any $r, s \in S$.

Following a random utility framework, these utilities have deterministic and stochastic components.

A consistent extreme value random utility model is a Thurstone random utility model where each ε_z has the extreme value distribution. Consistent model means $B_z = -W_z = U_z$ and $BW_{r,s} = U_r - U_s$, then

$$B_z = v_z + \varepsilon_z$$
$$W_z = -v_z - \varepsilon_z$$
$$BW_{r,s} = v_r - v_s + \varepsilon_r - \varepsilon_s$$

We can write the probability that an item is the best and the probability that the item is the worst.

$$B_S(x) = Pr(B_x = \max_{z \in S} B_z)$$
$$W_S(y) = Pr(W_y = \max_{z \in S} W_z)$$

Without even using the particular utility or scale of items we can derive choice probabilities. In the most general, form we suppose $b()$ and $w()$ are separate interval scales. Then the resulting probability of an item being best or worst is

$$B_S(x) = \frac{b(x)}{\sum_{z \in S} b(z)}$$
$$W_S(y) = \frac{w(y)}{\sum_{z \in S} w(z)}$$

For any pair of items, $x, y \in S$, we can write the joint probability that x is best and y is worst. However, when considering the best and worst jointly the scales b and w are not separately identified. So we fix their ratio for the same item by setting $w(z) = \frac{c}{b(z)}$. The resulting joint probability is a function of the ratios of scales for pairs of items,

$$BW_S(x,y) = Pr(U_x > U_z > U_y | z \in S - \{x,y\}, x \neq y)$$
$$BW_S(x,y) = \frac{b(x)/b(y)}{\sum_{r,s \in S, r \neq s} b(r)/b(s)}$$

To accommodate the standard utility structure, we let utility $u(z) = \log(b(z))$ Then each of the probabilities

$$B_X(x) = \frac{e^{u(x)}}{\sum_{z \in X} e^{u(z)}}$$
$$W_X(y) = \frac{e^{u(y)}}{\sum_{z \in X} e^{u(z)}}$$
$$WB_X(x,y) = \frac{e^{u(x)-u(y)}}{\sum_{r,s \in X, r \neq s} e^{u(r)-u(s)}}$$

We can derive the same representation from the random utility model Marley and Louviere (2005).

We can view best-worst choice as a generalization of the classic multinomial logit for the choice of the best only.

3.2. Estimation

From our perspective as researchers, all of the utilities are unknown. We can estimate the MaxDiff choice model in a variety of ways. One way to do this exactly is to enumerate all possible pairs of items x and y and then we describe their joint probability of being best and worst, which is the probability of being having the largest difference $BW_S(x,y)$. The pairwise approach scales quadratically in the number of items. We adopt an alternative approach, which reflects the literature and practice and is shown to be a near exact approximation Cohen et al. (2003). This allows us to estimate the best model and worst model independently, without explicitly estimating the best-worst probability.

$$B_S(x) = Pr(B_x = \max_{z \in S} B_z) = \frac{1}{1 + e^{X_B \theta + e_B}}$$
$$W_S(y) = Pr(W_y = \max_{z \in S} W_z) = \frac{1}{1 + e^{X_W \theta + e_B}}$$

We describe the data for any individual-task combination. Let $Y_{B_S}(z)$ be the binary choice variable, which equals 1 if the item is selected as best in the set S, and 0 otherwise. Then $Y_{W_S}(z)$ is the indicator of whether item $z \in S$ is selected as the worst. The design matrix X_{B_S} (of size $|S|$-by-K) contains indicator variables taking on value of 1 for each item in the current set S and 0 otherwise. To signal the item as worst, we set $X_{W_S} = -X_{B_S}$, so X_{W_S} contains values of 0 or -1. Taken together, we express the model of the choice data as a multinomial logit with choice probabilities in vector notation as follows,

$$\begin{bmatrix} p_b \\ p_w \end{bmatrix} = \text{logit}^{-1}(\begin{bmatrix} X_B \\ X_W \end{bmatrix} \theta + \begin{bmatrix} e_b \\ e_w \end{bmatrix})$$

The rows in this matrix representation represent every respondent-task-item combination, $N * J * |S|$, repeated twice. The probability vectors represent the collection of all best and worst likelihoods for each item in each set for all respondents, $[p_b, p_w] = \{B_S(z), W_S(z) | \forall z \in S, \forall S \in MDD_i, \forall i\}$. The link between the models of best choice and worst choice is the parameter $\theta = \{\theta_1, \ldots, \theta_k\}$. This common parameter vector represents the overall utility of each item $1, \ldots, k$. For a more positive θ_k, the item k has a larger probability of being chosen as best; the more negative, the more likely the item will be chosen as worst.

We clarify language about utility, which may diverge from conjoint language or language for multi-attribute profiles. Since X is an indicator, the θ only represents the utility of item k being included versus excluded. If $\theta_k > \theta_{k'}$, then we say item k is “more preferred,” “more important,” or simply, “better.”

Due to the sparse nature of MaxDiff for huge numbers of items plus the desire for rapid real time updates, we decided to use aggregate MNL rather than a Bayesian approach.

3.3. Adaptive MaxDiff

Since we are uncertain about the each parameter value, we continue collecting data. But we do not need to reduce that uncertainty equally for each one. To translate our current beliefs about parameters into action, we use Thompson Sampling to select which items to present to the next respondent.

$$w_k = \int_\theta Pr(u_k(\theta) = u^*(\theta)|D) p(\theta|D) d\theta$$

$$k \sim \text{multinomial}(w)$$

$$\arg\max_k u_k^{(g)} u_k^{(g)} \sim p(u|D)$$

Table 1 $O(w)$ **on 24 slices of simulated data with** $w = 6$

Algorithm	
MaxDiff TS	
MaxDiff ϵ-Diffuse TS	
ϵ-greedy	
ϵ-Diffuse TS closest to the threshold	
ϵ-greedy closest to the threshold	Take the $(1-\epsilon)L$ items that have estimated utility closest to $\frac{\theta_k+\theta_{k+1}}{2}$
Misclassification Minimization with random perturbation	
Greatest Uncertainty with random perturbation	

3.4. Algorithm MaxDiff TS

```
BEGIN:
  1. For respondents n =1,...,initial
    (a) Act: select items for MaxDiff design
      i. Sample $L$ items from the population
      ii. Create and serve MaxDiff questionnaire to next respondent $J$ tasks of $|S|$
Qs=MDDesign($L$,$|S|$)
  2. For respondents n = initial + 1,..., N
    (a) Estimate: update model $p(D|\theta)$
      i. Sample model parameters $u_1, \ldots, u_k \sim p(u|D)$
    (b) Value: rank items by utility
      i. Rank and take top $L$ subset $u^{(1)}, u^{(2)} \ldots, u^{(L)}$
    (c) Act: select items for MaxDiff design
      i. Create and serve MaxDiff questionnaire to next respondent $J$ tasks of $|S|$
Qs=MDDesign($L$,$|S|$)
END ALGORITHM
```

We follow the algorithm Bandit MaxDiff-TS. After an initial number of respondents, we begin adapting. We update beliefs about the model parameters. We can draw from an exact or approximate posterior distribution. [1]

From a single sample of parameter values, we rank the items by utility. We select the top

[1] In the case of the multinomial logit, we can use MCMC to obtain samples or, as we do in the empirical application, we can sample from asymptotic distribution via MLE implied by the estimated mean and standard errors.

L items, and we generate a MaxDiff design of J best-worst tasks, each with a choice set of $|S|$ items. We use $L = 30$, $J = 18$, $|S| = 5$ as a default since it is a standard number of items used in MaxDiff studies Wirth and Wolfrath (2012). While typical Bayesian or numerical integration methods call for large numbers of draws (or sufficient number) to achieve coverage of the full distribution of parameter values, our approach relies on the variability of a single sample. There is value in the sample to sample differences in the value of parameters and their relative rank ordering. It allows

The algorithm learns over time to intuitively achieve the goal of identifying the items with truly high utility with high precision. Early on, we still have substantial uncertainty. The independent samples will differ substantially in rank order of item utilities. This yields MaxDiff designs across respondents with less overlap in items. Later, the uncertainty is reduced most around the truly high-utility items. Across independent samples, the ranking of items will be highly correlated near the top of the ranking (but not near the bottom where uncertainty remains large). As a result, the top subset of J items selected converges to the same group for each respondent.

There are two practical issues we highlight. First, we want to avoid repeating extremely similar questions to the same respondent. A natural consequence of Thompson Sampling is convergence: as the sample size grows, items that are most preferred by the population will achieve high preference scores with smaller standard errors. Without any additional restrictions, the same few items will eventually tend to be drawn into adjacent MaxDiff tasks for the same respondent, causing much annoyance due to the severe degree of item repetition. Although this is statistically most efficient, it would drive human respondents mad. To avoid this, we use Thompson Sampling to draw a fixed number of items (e.g., 20 or 30) to show each respondent. Those draws of 30 items are shown to each respondent in a balanced, near-orthogonal design, leading to a palatably low degree of repetition of items across adjacent sets. The attentive reader will notice that our approach is quite similar to Wirth's Express MaxDiff approach, except that the logic for selecting the 30 items for each respondent is adaptive, using Thompson Sampling, leveraging information from the previous respondents-focusing the most recent respondent's efforts on discriminating among items that already have been judged likely to be the stars.

One issue this approach avoids is the respondent tiring out. Instead of trying to create a MaxDiff design with 120 items, which would lead to an unreasonable number of best-worst tasks per respondent, we use the smaller subset of 30. We also avoid repetition. A particularly annoying alternative is to draw an independent Thompson sample ranking for each choice task for the same respondent. This results in very repetitive choice tasks as parameter estimates.

The second practical issue with Thompson Sampling is robustness to changes overtime. On the one hand, there is built-in robustness. Recall the algorithm is stochastic and adapts continuously. If the early data leads the algorithm astray, then it will self-correct, eventually finding and converging to the truly best items. Suppose the early respondents made choices, by chance, leading us to believe certain items were the best when they were not. The respondents that immediately follow will start receiving these truly poor items. However, as this continues, the uncertainty is reduced around these poor items to reveal there are many other items with probability of being better. By sampling from the joint belief distribution of item utilities, we will be less likely to draw those poor items. One concern is that such sampling could be too aggressive.

3.5. Algorithm MaxDiff ϵ-Diffuse TS

Perhaps the natural parameter uncertainty is not enough or is perhaps too slow to adjust, so we propose an algorithm MaxDiff epsilon-diffuse TS, which has an extra layer of self-correction, making it more robust to non-stationarity or respondent self-selection. The way the ϵ-δ version hedges its bets on the best items intuitively. As in illustration, instead of sampling 30 items from TS, we use 20 of the items drawn using standard Thompson Sampling (L=30, ϵ=1/3), and 10 of the items drawn using Thompson Sampling with a much more diffuse prior (variance of estimate multiplied by δ=10). For ϵ-δ case, its density does not become a spike as quickly as it does without ϵ-δ . This speed is controlled through two parameters: epsilon, the proportion of items sampled from the diffuse distribution, and δ, factor increasing variance for the diffuse distribution. The larger the ϵ and δ, the more exploration and slower the algorithm settles on its set of items. The epsilon comes from the popular ϵ-greedy. The ϵ-δ generalization is just like a two-component mixture model, which nests a pooled model without segments. As the epsilon is zero, the diffuse distribution is never used, so the algorithm collapses to regular TS. As δ is 1, the diffuse distribution is equivalent to the non-diffuse distribution. In Toubia and Florès (2007) a similar idea is used to avoid misclassifying an item that will not be sampled further. The author derived insight from genetic algorithms where "decreasing the mutation rate over the number of iterations (or generations) leads to higher performance compared to a fixed mutation rate." This suggests sampling from a diffuse distribution gives benefit over uniform sampling over the remaining items as the variance of the diffuse distributions shrink with the number of respondents interviewed. It should be noted that in our application the ϵ-greedy algorithm preformed on par the ϵ-δ algorithm for $\epsilon = \frac{1}{3}$.

3.6. Variant MaxDiff epsilon-Diffuse TS

```
BEGIN:
1. For respondents $n = 1, \ldots, N_{initial}$
   (a) Act: select items for MaxDiff design
       i. Sample $L$ items from the population
       ii. Create MaxDiff questionnaire
       iii. Survey next respondent with $J$ tasks of $|S|$ Qs=MDDesign($L$,$|S|$)
2. For respondents $n = N_{initial} + 1, \ldots, N$
   (a) Estimate: update model $p(D|\theta)$ to obtain population-level estimates,$\bar{u}$ and $\Sigma$.
       i. Diffuse TS: Sample model parameters from diffuse distribution, by using rescaled larger variance-covariance $\delta\Sigma$, with $\delta > 1$
              $u_D \sim N(\bar{u}, \delta\Sigma)$
       ii. Regular TS: Sample model parameters
              $u_R \sim N(\bar{u}, \Sigma)$
   (b) Value: rank items by utility
       i. Take top $\epsilon L$ items ranked diffuse sample $u_D^{(1)}, u_D^{(2)} \ldots, u_D^{(\epsilon L)}$
       ii. Take top $(1-\epsilon)L$ items ranked regular sample $u_R^{(1)}, u_R^{(2)} \ldots, u_R^{((1-\epsilon)L)}$
       iii. Combine diffuse and regular samples into $L$ items. (Remove duplicates items. Take the next best unique item from the diffuse sample not already selected.)
   (c) Act: select items for MaxDiff design
       i. Create MaxDiff questionnaire.
       ii. Survey next next respondent with $J$ tasks of $|S|$ Qs=MDDesign($L$,$|S|$)
END ALGORITHM
```

3.7. Algorithm ϵ-greedy

Let β be the current estimated parameters. Take the top $((1-\epsilon)L)$ and choose the remaining (ϵL) uniformly from the remaining items.

3.8. Variant Closest to the Threshold

In Toubia and Florès (2007), They introduce Closest to Threshold based on Bradlow and Wainers (1998) recommendation ("select ideas near the cutoff"). We also we do that to find the top k. Based on the estimated utilities calculate the cutoff $c = \frac{\theta_k + \theta_{k+1}}{2}$ then for a draw u calculate a score $s_i = |c - u^i|$ for every item. Then take the L items with the lowest score. Similarily as before we can define ϵ diffuse version where. We have two draws u_R and u^D and from those scores s_R and s_D and Combine diffuse and regular samples into L items. (Remove duplicates items. Take the next best unique item from the diffuse sample not already selected.) You can also do the ϵ - greedy version where the scores are $s_i = |c - \beta^i|$ you take the $((1-\epsilon)L)$ lowest scores and the rest uniformly.

3.9. Algorithm Misclassification Minimization

Also taken from Toubia and Florès (2007). You define a score, s_i for each item as follows. If the ith item is estimated to be a bottom item the score is $s_i = \mathbb{P}(\text{item i is in the top k})$. If the ith item is estimated to be in the top k then the score is $s_i = \mathbb{P}(\text{item i is not in the top k})$. You can estimate this quanity by taking a large number of draws from the posterior distribution. You take the L items with the highest score.

3.10. Algorithm Greatest Uncertianty

You define the score, s_i for each item as follows. $s_i = |.5 - \mathbb{P}(\text{item i is in the top k})|$. You can estimate this quanity by taking a large number of draws from the posterior distribution. You take the L items with the lowest score.

3.11. Variant Random Pertubation

Using bayes bootstrapping for estimating probabities required in Misclassification Minimization and Greatest Uncertianty takes a good deal of time. As an alternative you can estimate the scores once every b people, Let us call b the batch size. And then for each person perturb the scores by a normal vector with mean zero and varaince $\frac{1}{n_i}$, Where n_i is the number of people who have seen item i on previous surveys, and serve up the L items accordingly.

3.12. Main Empirical Analysis

We compared our proposed bandit MaxDiff approaches to existing non-adaptive fixed MaxDiff design strategies using simulation based on inferred preferences from an actual MaxDiff survey. We used MaxDiff data from a survey conducted by Procter & Gamble with Sawtooth Software. The study involved 981 respondents and 120 items from a sparse MaxDiff study. The subject matter and exact item text was hidden for confidentiality purposes. They provided us with the individual-level posterior mean utilities for all respondents and items, which were obtained via Markov Chain Monte Carlo sampling for a hierarchical Bayes (HB) logit model. We call those individual-level utilities the true HB utilities. These HB utilities offered realistic patterns of preferences across the items and respondents for use in our respondent simulations. Our simulated respondents mimicked the actual respondents' preferences, on average: to answer each new MaxDiff task, we perturbed those true HB utilities by iid Gumbel error.

For each sample of simulated respondents, we ran aggregate logit and compared the current rank order of the estimated aggregate item scores to the true rank order for the known true utilities. We define hit rate to reflect how accurately we could recover the top few items as observed in the true preferences for the Procter & Gamble dataset. To stabilize the hit rate results (since there was a random component to the responses), we ran the simulations each 100 of times.

We used two hit rate measures.

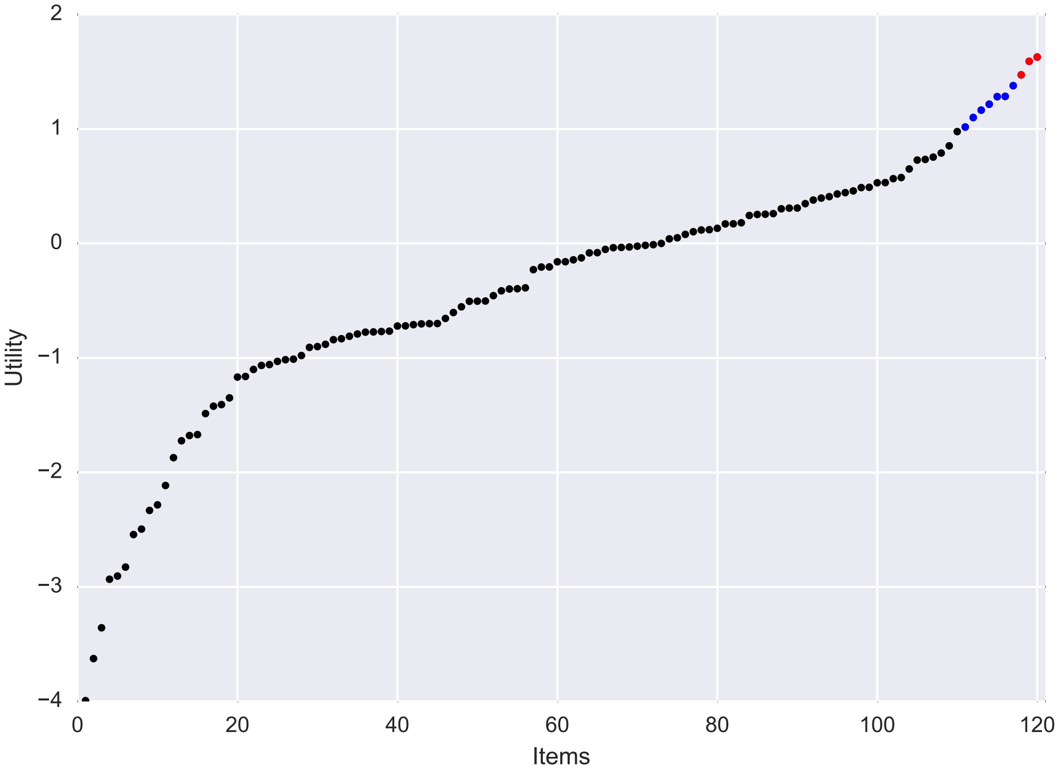
Top 3 hit rate: what percent of the top 3 true items the estimated scores using robotic respondents identified.

Top 10 hit rate: what percent of the top 10 true items the estimated scores using robotic respondents identified. For instance, if the estimated scores identified 7 of the true top 10 items (irrespective of order), the hit rate was 70%.

Hit rate is analogous to regret in a typical bandit problem. It measures how far we are from always selecting the truly best arm for all time periods. Although it is not exactly an observed reward in the sense of the usual bandit setting (e.g., clicks, purchases), this serves as the measure of success, *ex post*.

The difficulty of identifying the truly best items is related to the differences among those top item utilities. With 120 items in the dataset, it shouldn't surprise us if the true preferences for the top 15 or so items were very close in terms of utility. We certainly

Figure 5 **Distrubution of True Utilities**

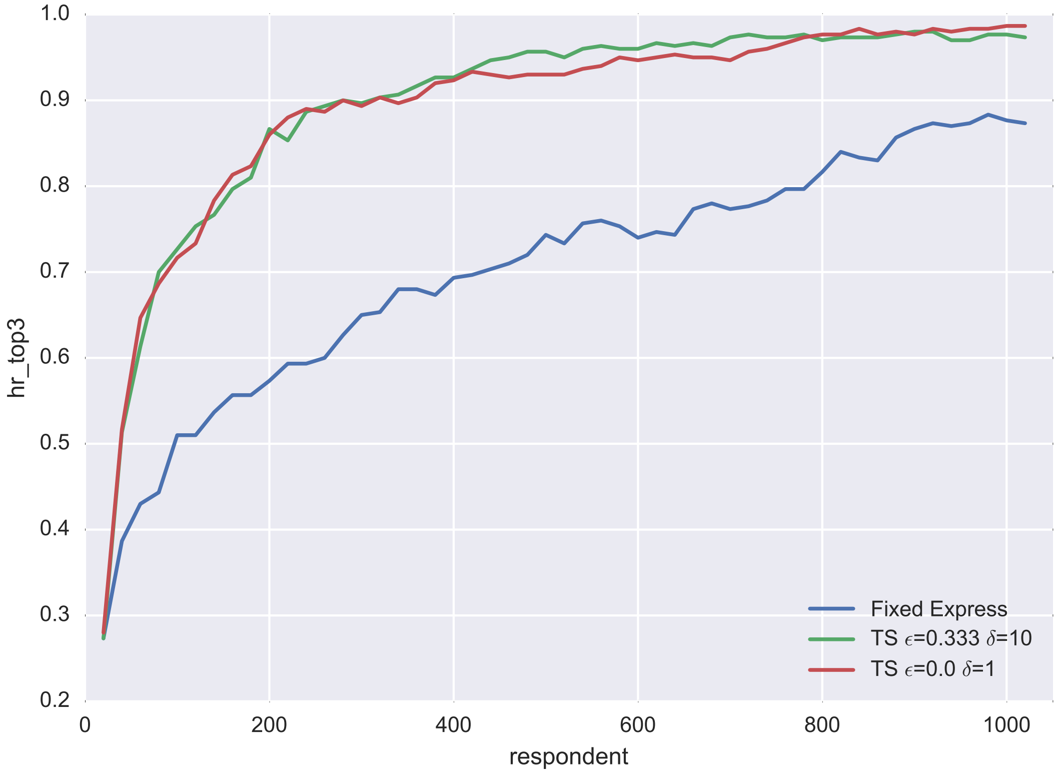


observed that with this dataset (Figure 5). Due to how tightly the top items in preferences clustered (there were no runaway winners), the hit rate measures we employed were quite discriminating between competing methods.

Using bootstrap sampling (sampling with replacement), we simulated the process of collecting respondent data up to sample sizes of $N = 1020$. We considered the first 20 respondents to be the initial group, and all adaptive methods only begin after the 20th respondent. Each robotic respondent completed 18 choice sets, where each set included 5 items, which was viewed as fairly typical of larger MaxDiff studies in practice. The MaxDiff approaches we tested were the following. For our actual implementation of Bandit MaxDiff, we used Thompson Sampling to draw 30 of the 120 items for each respondent (tending to oversample the "stars" based on aggregate logit estimates from previous respondents). As described earlier, we tested two different Bandit Maxdiff approaches:

Bandit MaxDiff TS: All 30 items drawn via standard Thompson Sampling.

Figure 6 **3 Hit Rate with 120 items**



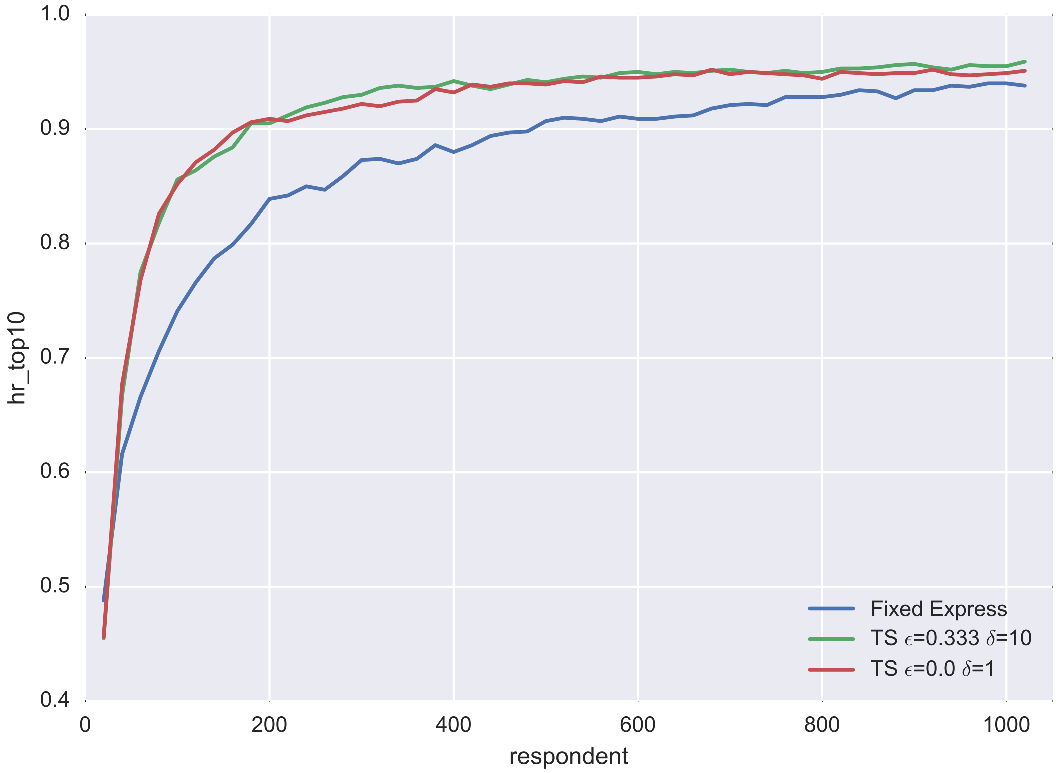
Bandit MaxDiff epsilon-diffuse TS: We begin with $\frac{1}{3}-\frac{2}{3}$ split ($\epsilon=\frac{1}{3}$, δ=10). So 20 of the items drawn using standard Thompson Sampling; 10 of the items drawn using Thompson Sampling with a much more diffuse prior (standard errors multiplied by 10). We also tested a pair of existing fixed MaxDiff approaches.

Fixed Sparse MaxDiff: we showed each item to each respondent an equal number of times (if possible). With 120 items, 18 sets, and 5 items per set each item appeared on average $\frac{18*5}{120}=0.75$ times per respondent.

Fixed Express MaxDiff: we randomly drew 30 of the 120 items to show to each respondent. Each item appeared $\frac{18*5}{30}=3$ times per respondent. Across respondents, each item appeared the same number of times. We primarily use Express MaxDiff as our base case.

Figure 6 shows the results for the 120-item dataset. The X-axis indicates the number of cumulative respondents interviewed, and the Y-axis reports the hit rate obtained at each

Figure 7 **10 Hit Rate with 120 items**



cumulative sample size. For example, after the first 200 respondents, the Fixed Express Design obtains a hit rate of less than 60% whereas both Bandit MaxDiff approaches achieve a hit rate of about 85%.

The key takeaway from Figure 6 is simple: Thompson Sampling is about 4x more efficient than the standard Fixed Express MaxDiff approach. We show in Table 2. After about 160 respondents, we've obtained an 80% hit rate; which we wouldn't achieve with traditional Express MaxDiff until about the 800th respondent. As shown, we can accomplish with about 260 respondents what it takes 1000 respondents to do with Express MaxDiff. Either comparison shows that you can obtain equally good hit rates using the adaptive Bandit MaxDiff methodology with $\frac{1}{4}$ the sample size.

Figure 6 also shows the two Thompson Sampling approaches achieve nearly identical results. But we will show later this isn't the case if we use a misinformed start (Section Misinformed Start)

Figure 7 shows the results for Top-10 hit rate for our 120-item MaxDiff, which is a

Respondent	Fixed Express	TS $\epsilon=\frac{1}{3}$ $\delta=10$	TS $\epsilon=0$ $\delta=1$
60	43	61.33	64.67
100	51	72.67	71.67
160	55.67	79.67	81.33
200	57.33	86.67	86
260	60	89.33	88.67
300	65	89.67	89.33
360	68	91.67	90.33
400	69.33	92.67	92.33
460	71	95	92.67
500	74.33	95.67	93
560	76	96.33	94
600	74	96	94.67
660	77.33	96.67	95
700	77.33	97.33	94.67
760	79.67	97.33	96.67
800	81.67	97	97.67
860	83	97.33	97.67
900	86.67	98	97.67
960	87.33	97	98.33
1000	87.67	97.67	98.67

Table 2 Top 3 Hit Rate for Various Algorithms

broader measure of success that requires obtaining a high degree of precision for an even broader reach of items than the Top-3 hit rate. The conclusions are fairly similar as with Figure 6. It takes about 400 respondents to accomplish with Bandit MaxDiff what we accomplish with 1000 respondents under the non-adaptive express MaxDiff approach. This is supported in Table 2

3.13. Empirical Evidence of TS Focusing on Top Items

In figure 8 we have the average (over 100 trials) precision for the top ranked item and 60th ranked item given by Bandit MaxDiff TS and Fixed Express MaxDiff. Figure 9 shows the

Figure 8 **Precision**

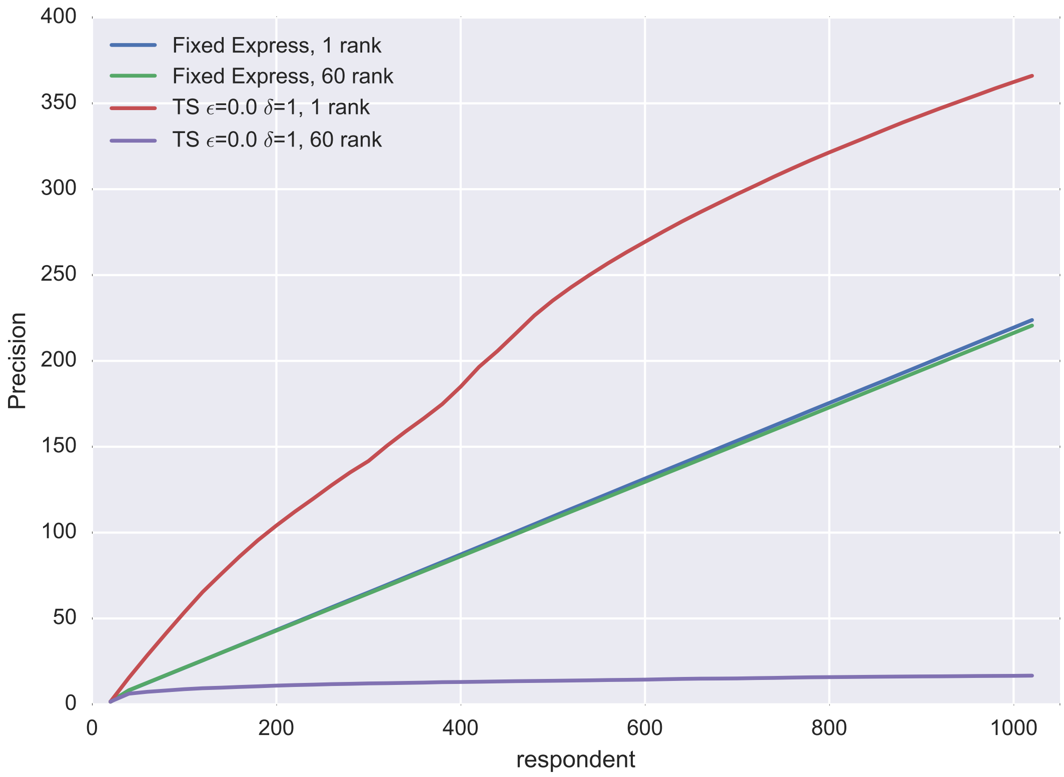
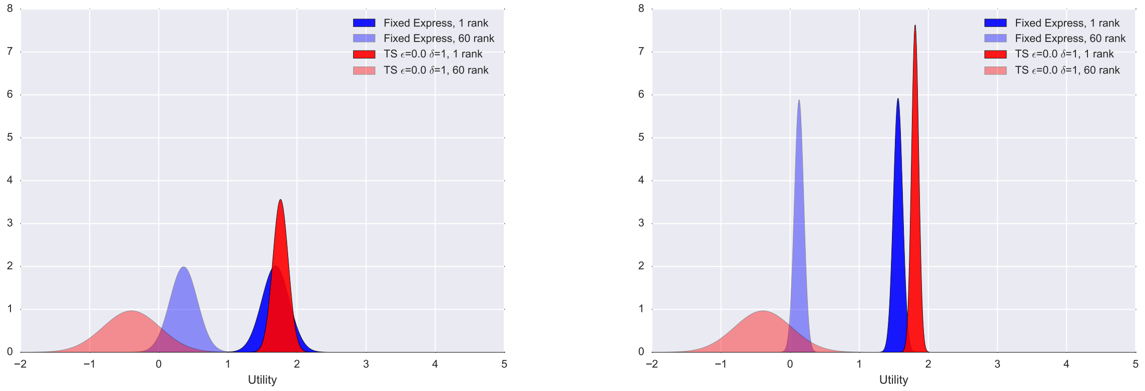


Figure 9 **Distrubution of Utility Belief Distribution**

(a) At 120th Respondent

(b) At 1020th Respondent



average posterior distribution of the two items at the 120th and 1020th respondent. The take away is that between the 120th and 1020th respondent the rank 60th item gets almost no update from TS, while TS has a large precision on rank 1 item. Fixed Express on the other hand updates them both equally as evidence to our claim why TS works well in this context, TS increases the precision a lot on the higher ranked items of those items while not learning the lower ranked items while Sparse Express spends its resources equally.

4. Robustness tests

4.1. Misinformed Starts (When Early Responders Are Horribly Non-Representative)

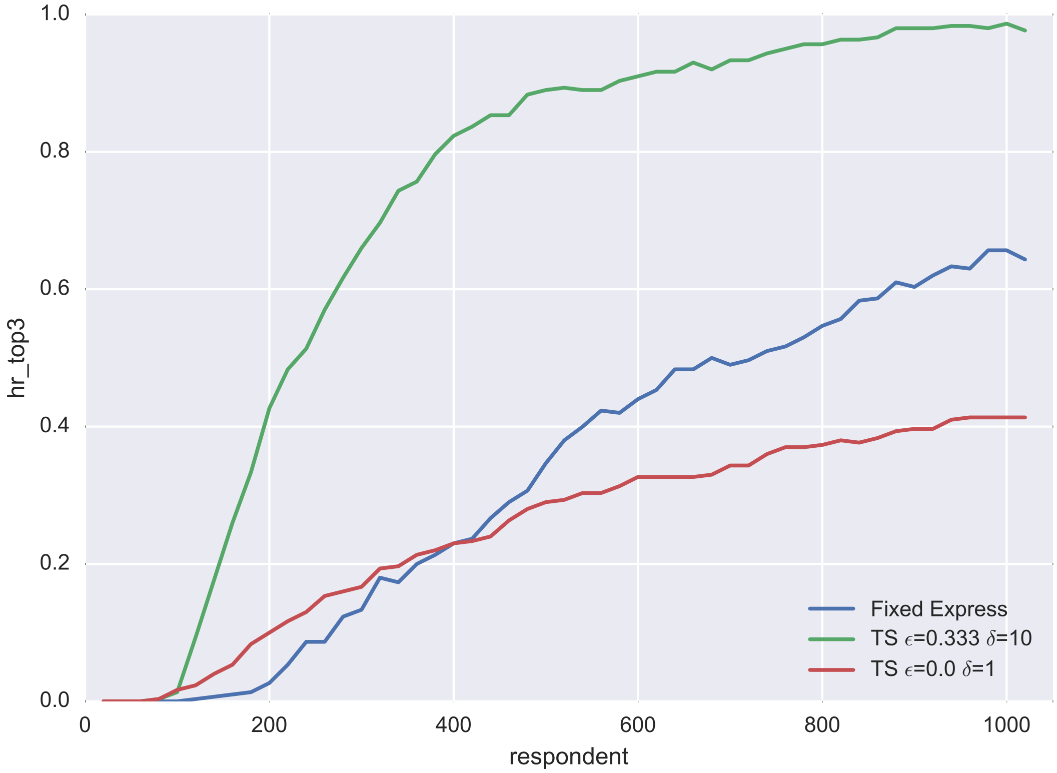
To this point, it seems like the adaptive Bandit MaxDiff approach using Thompson Sampling is the clear winner. However, what would happen if the first 50 respondents we interviewed were actually not very representative of the average preferences for the sample? What if we tried to throw Bandit MaxDiff off the scent? In fact, let's consider a worst-case scenario: the first responders actually believe *nearly the opposite* from the rest of the sample

For the simulations reported in Figure 6, the first 50 robotic respondents mimicked randomly drawn human vectors of utilities as before but were diabolically manipulated to behave as if the top 3 true items were actually nearly the worst in preference (we set the utilities for the top 3 true items for the population equal to the bottom 25th percentile utility item for each respondent). After this misinformed start, the remaining respondents represented well-behaved respondents drawn using bootstrap sampling as before, with true individual-level preferences as given in the original dataset donated by Procter & Gamble.

Our diabolical simulation of a misinformed start is worse than anything you would realistically see in practice, so it is a strong test of the robustness of the Bandit MaxDiff approach. This suggests robustness to non-stationarity in preference or self-selection of respondents during the sampling window. We created 50 misinforming early responders because in practice we are never guaranteed that the first responders represent a fair and representative draw from the population. In fact, depending on how rapidly we invite a panel of respondents to take the survey, the first 50 respondents may share some atypical characteristics (e.g. anxious and available to take the survey at your launch time). It would be a bad thing if the Thompson Sampling approach performed well in simulations with well-behaved respondents, but fell apart under more realistic conditions.

We illustrate the robustness in Figure 10. First, the Bandit MaxDiff ϵ-diffuse TS approach is much better in the face of misinformed starts than the standard Bandit MaxDiff TS

Figure 10 **3 Hit Rate with 120 items with Misinformed Start**



approach. The more diffuse prior on the 10 out of 30 items within the split allows us to continue investigating the value of some lesser chosen items with enough frequency among later respondents, even if the prior respondents seem to have generally rejected them. Second, even in the face of a misinformed start, the Bandit MaxDiff ϵ-diffuse TS achieves equally good results as the standard Fixed Express MaxDiff without the misinformed start after 290 respondents. Finally, the conclusions were qualitatively similar when examining Top-10 hit rate.

4.2. Changing exploration parameters

We introduced two tuning parameters – ϵ and δ – into an algorithm that previously had no such parameters explicitly guiding exploration. To understand the impact these exploration parameters have and each one's effect on the outcome, we have to test their values empirically.

Respondent	TS $\epsilon=0$ $\delta=1$	TS $\epsilon=\frac{1}{3}$ $\delta=10$	TS $\epsilon=\frac{1}{3}$ $\delta=20$	TS $\epsilon=\frac{1}{2}$ $\delta=10$	TS $\epsilon=\frac{1}{2}$ $\delta=20$
100	71.67	72.67	74	74.67	71.33
200	86	86.67	84.67	89.67	82.67
300	89.33	89.67	92.67	94	91.33
400	92.33	92.67	94	97.33	95.33
500	93	95.67	98.67	98.33	96.33
600	94.67	96	99.33	99	97.33
700	94.67	97.33	98.67	98.67	96.67
800	97.67	97	99	98.67	98.33
900	97.67	98	99	98.33	98.67
1000	98.67	97.67	98.67	99.33	98.33

Table 3 Top 3 Hit Rate for Various Tuning Parameters

We run a simulation experiment with a 5-cell design $2(\epsilon=\frac{1}{3},\frac{1}{2})$-by-$2(\delta=10,20)+1(\epsilon=0,\delta=1)$. The standard TS has $\epsilon=0$ and $\delta=1$ and the Fixed Express MaxDiff is equivalent to $\epsilon=1$ and $\delta=\text{inf}$, rendering any parameter distribution to be uninformative, as it is effectively uniform.

Table 3 shows how these parameters can affect performance; these effects, however, are small. The best performing combinations are $\epsilon=\frac{1}{2}$, $\delta=10$ and $\epsilon=\frac{1}{3}$, $\delta=20$. This suggests there is an “interior solution,” as there are non-monotonic effects of the exploration parameters. There is a value of ϵ, which makes ϵ-δ TS better than TS, and the same is true for δ. This pattern is consistent with empirical implementations of ϵ-greedy strategies. In fact, the table shows the best performing algorithm tested uses only one exploration parameter set high, and the worst performing algorithm tested has both parameters set to their higher-exploration level.

4.3. Effect of increasing number of items

Would the benefits of Bandit MaxDiff we observed with 120 items continue for 300 items? While we didn’t have a dataset of utilities from human respondents on 300 items, we did our best to generate such a data set by leveraging the 120-item data set Procter & Gamble shared with us. To generate preferences across an additional set of 180 items, we randomly combined pairs of existing items according to a randomly distributed weighting

Respondent	Fixed Express	TS $\epsilon=\frac{1}{3}$ $\delta=10$	TS $\epsilon=0$ $\delta=1$
100	22.33	44.33	47.67
200	31	70	67
300	41.33	76	72
400	42.67	79	78.33
500	46.33	85.33	79.33
600	48	89.33	82
700	53.67	88.67	85.67
800	54.67	90.67	86.33
900	55.67	91.33	88
1000	59	91.33	89

Table 4 **Top 3 Hit Rate for Various Algorithms with 300 items**

scheme, with additional random variation added. The result was a 300-item MaxDiff data set based on the original preferences of the 981 respondents.

The advantages seen in the 120-item results are improved upon in the results with 300 items. Table 4 shows the well-informed start results. The Bandit MaxDiff approach was 6x more efficient than the Fixed Express MaxDiff approach on the top-3 hit rate criterion.

4.4. What about a smaller set of 40 Items?

Our Bandit model has a great advantage over fixed designs for very large numbers of items, but what happens if we have a more traditional “large” MaxDiff list of 40 items. Using a random 40-item subset from our original set of 120 items, we reran our simulations. For this smaller subset, we also reduced the number of tasks per respondent to 12 and we drew 20 items per respondent using TS rather than 30 items and used a 15/5 split between standard TS and diffuse TS.

By reducing the number of items to 40 and changing the number of items drawn to 20, the Express MaxDiff design can now show each item to every other respondent on average, which is much more than in larger item cases. Nevertheless, our results for Bandit MaxDiff are still much better than traditional MaxDiff (Table 5). We see about a 2x efficiency gain compared to the Express MaxDiff. We also still see a tremendous advantage in the case of a misinformed start.

Respondent	Fixed Express	TS $\epsilon=\frac{1}{4}$ $\delta=10$	TS $\epsilon=0$ $\delta=1$
60	71.33	81	84.33
100	79.33	88	89.33
160	86.33	93	93.33
200	91	95.67	94.67
260	92.67	96.33	96.33
300	94.33	96.67	97
360	94.67	98.33	98
400	94.33	98	98.33
460	94.67	99	99
500	95.33	99.33	98.33

Table 5 **Top 3 Hit Rate for Various Algorithms with 40 items**

S	Percent True Utility
{1,2,3}	1.0
{1,2,4}	.973
{1,2,5}	.948
{1,3,4}	.934
{1,3,5}	.910
{1,4,5}	.882
{2,3,4}	.922
{2,3,5}	.897
{2,4,5}	.870
{3,4,5}	.831

Table 6 **Percent true utility for Various S for a Top 3 scheme from the Procter & Gamble data**

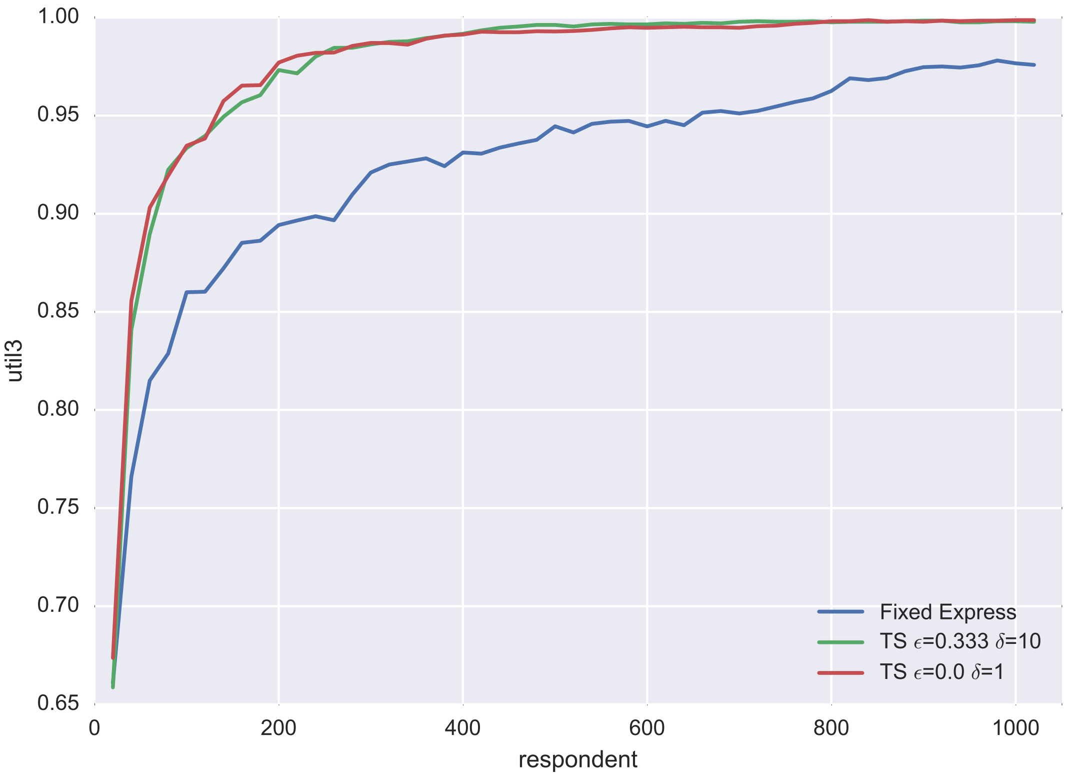
4.5. Using True Utility as a Generalization of Hit-Rate

One might like to differentiate between an algorithm puts the top 9 items and the 11th item in the 10 top and one puts the top 9 items and the 40th item in the 10 top. A measure that differentiates the two is Percent True Utility (PTU).

Exponential Weighted Utility: For a set S the exponential weighted utility is

$$\mu_S = \sum_{x \in S} b(x) = \sum_{x \in S} e^{u(x)}$$

Figure 11 **Percent True Utility of the Top 3 with 120 items**



Percent True Utility of the Top 3: The Percent True Utility (PTU) is the exponential weighted true utility of the top 3 items that the robotic respondents identified over the maximum exponential weighted true utility that could be attained with 3 items. Mathematical if $\tilde{S}$ is the top 3 items and S is the current 3 top items identified robotic respondents then PTU is

$$\frac{\mu_S}{\mu_{\tilde{S}}}$$

This can be see as a generalization of Hit-Rate. It has the advantage of differentiating as in the case above though one loses the ability to easy interpret the result. In Table 6 we show the PTU for various subsets of the data. In figure 11 we see after 200 respondents TS on average is about .97 so from table 6 we can deduce that usually TS has the top 2 items in the top 3 and trying to decide between the 3rd and 4th item. After 200 respondents Sparse Express is about .89 which means that on average after 200 iterations has top 5 items in its top 3. Compare that to before, where Sparse Express a 3 hit rate score of 0.58. Combined

Current	1st	2nd	3rd	4th	5th	6th	7th	8th
Draw 1	4.02	3.50	5.08	4.16	4.22	4.41	3.65	3.27
Draw 2	4.18	4.72	3.49	3.48	3.63	3.60	3.56	3.70
Draw 3	4.81	5.23	5.04	3.96	4.17	4.37	3.58	2.99

Table 7 Draws of the exponetial utility of the items after 100 iterations

we know that Sparse Expressed finds which the tops items but has trouble distinguishing what is the correct ranking of those items.

4.6. Using Regret as a Stopping Rule

From figure 11 after 600 respondents TS finds the top 3 items most of the time. Thus it becomes inefficient to keep giving surveys after this point. In practice we do not know the True Utilities *a priori* but we would still like to know when we should stop.

Taken from Scott (2015) and Scott (2010) for MAB, The value remaining in the experiment is the posterior distribution of $\frac{\mu_{S*}-\mu_S}{\mu_S}$ where μ_{S*} is the largest value of the exponential weighted utility and μ_S is the exponential weighted utility of the set that is most likely to be optimal, denoted S. This is constructed as follows, take n Monte Carlo draws from $p(\mu|y_t)$. Let μ^m_{S*} be the max exponential weighted utility of draw m and μ^m_S be the utility using the draw m using the set S. Let $\Delta^m = \frac{\mu^m_{S*}-\mu^m_S}{\mu^m_S}$.

As an example I took draws after respondent 100 in TS $\epsilon = .33$ $\delta = 10$ for 120 items see table 7 for draws of exponential weighted utility of a single item. I put the columns in current rank order and only show the top 8 for convenience. Then S is the set containing what is currently ranked as the first, second and third ranked item. Then $\mu^1_S = 4.02 + 3.50 + 5.08 = 12.6$ and $\mu^1_{S*} = 5.08 + 4.41 + 4.16 = 13.65$ So $\Delta^1 = \frac{13.65-12.6}{12.6} = .083$. Likewise $\Delta^2 = \frac{12.6-12.39}{12.39} = .017$ and $\Delta^3 = \frac{15.08-15.08}{15.08} = 0$ (Note $\Delta^m = 0$ when the S contains the top utilities). The histogram of Δ after 100 iterations and 220 iterations is shown in figure 12. The 'potential value remaining' (PVR) is the .95 quantile of the distribution Δ. After 100 iterations the PVR was .092. Scott's way to interpret this number is "we do not know what the utility of S is, but whatever it is, a different set might beat it by as much as 9.2%."

A good stopping rule is to stop when the PVR drops below a certain threshold, (We use .05 and .02). One advantage of this is it handles ties (two sets with close exponential weighted utility scores) really well. Figure 13 shows the average PVR on a log scale. Figures 14a and 15a show on which respondent the iterations' PVR would reach .02 and .05 respectively

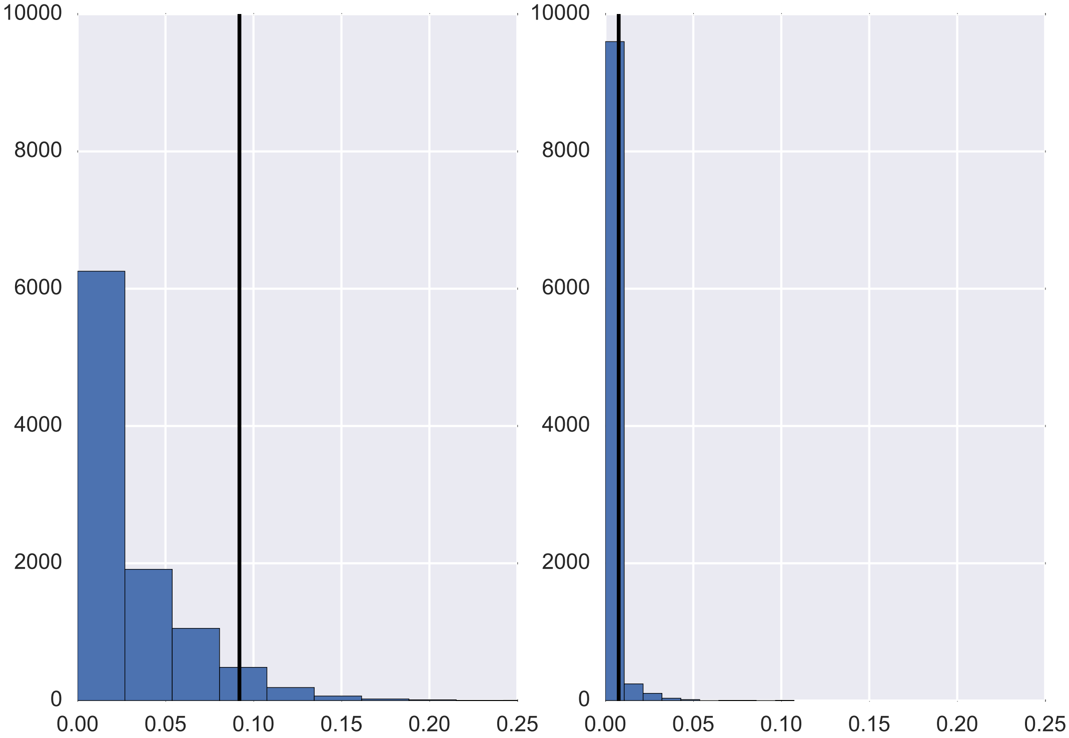


Figure 12 Two histograms of Δ. Left: After 100 iterations, the potential value remaining is .092. Right: After 220 iterations, the potential value remaining is .008

	Fixed Express	TS $\epsilon=\frac{1}{3}$ $\delta=10$	TS $\epsilon=0$ $\delta=1$
120 items			
Mean Stop	980.8	440.6	606.0
Mean PTU	.976	.997	.996
300 items			
Mean Stop	1020.0	674.2	929.4
Mean PTU	.904	.992	.988

Table 8 Mean respondent each algorithm would stop at and the mean percent true utility (PTU) for stopping when the PVR reaches .02

(with the max being 1020) and Figures 14b and 15b shows what the Percent True Utility would be at those points.

In tables 8 and 9 we see the means of the when the iterations would stop and percent true utility at those points for 120 and 300 items.

Figure 13 **Potential Value Remaining of the Top 3 with 120 items on a semi log plot, the dashed lines are at .05 and .02**

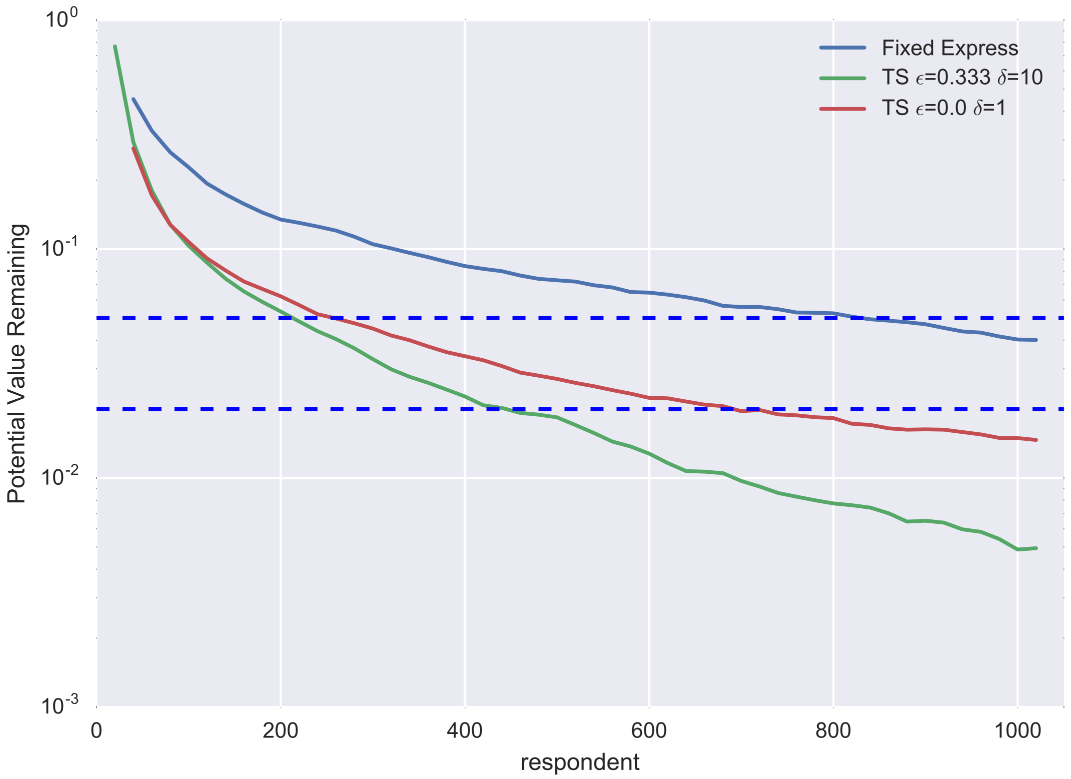
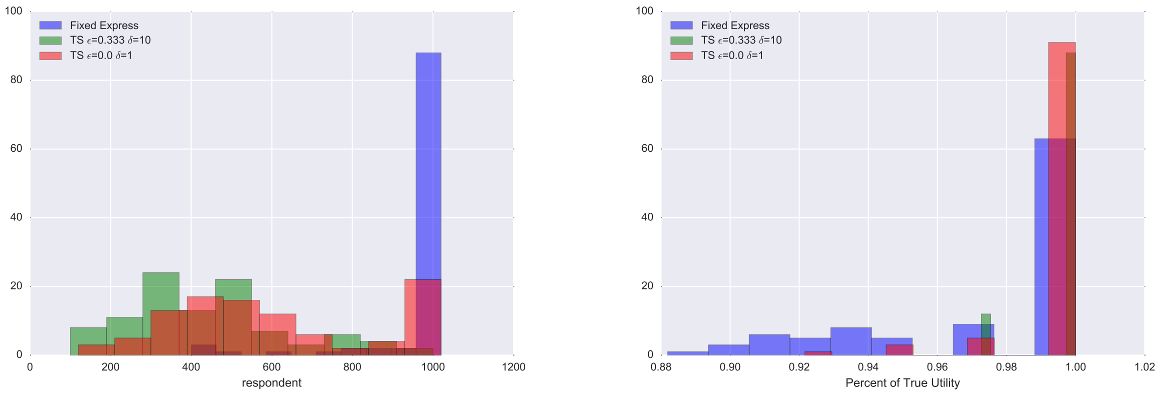


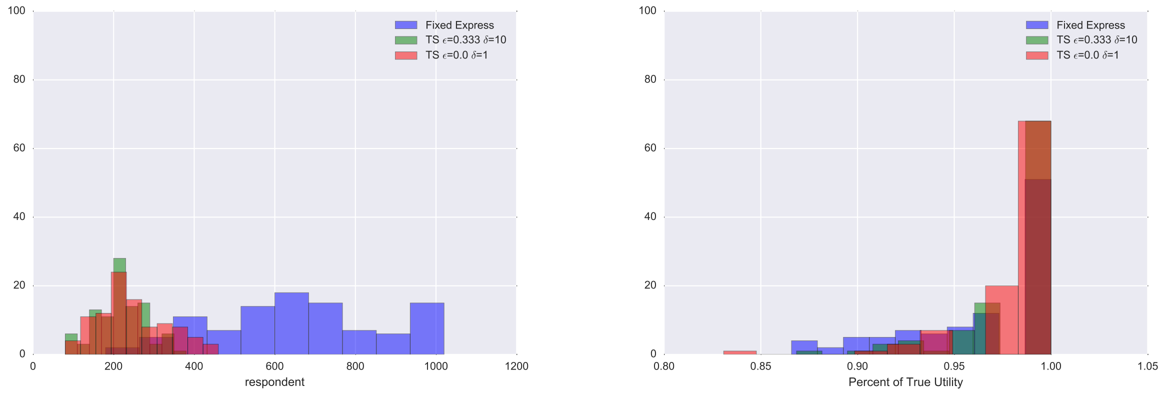
Figure 14 **Histogram of When Iterations Would Stop and PTU Using a .02 Stopping Rule for Top 3 for 120 Items**



(a) On which Respondent does Iteration Stops

(b) Percent True Utility When Iteration Stops

Figure 15 **Histogram of When Iterations Would Stop and PTU Using a .05 Stopping Rule for Top 3 for 120 Items**



(a) On which Respondent does Iteration Stops

(b) Percent True Utility When Iteration Stops

	Fixed Express	TS $\epsilon=\frac{1}{3}$ $\delta=10$	TS $\epsilon=0$ $\delta=1$
120 items			
Mean Stop	661.6	214.0	243.8
Mean PTU	.966	.984	.986
300 items			
Mean Stop	1001.2	334.6	526.8
Mean PTU	.904	.970	.980

Table 9 **Mean respondent each algorithm would stop at and the mean percent true utility (PTU) for stopping when the PVR reaches .05**

Another advantage we see from this for Bandit MaxDiff ϵ-diffuse TS approach is that on average it stops sooner that standard TS with about the same performance. This advantage grows with the number of items. Both Bandit MaxDiff TS stop much sooner and preform much better then Fixed Express. The choice between a 0.05 stopping rule or a 0.02 stopping rule is a trade-off between the having to survey extra respondents and getting the extra utility.

4.7. Asking for Bests instead of Best-Worst

Because a key assumption for using the Bandit MaxDiff approach is that the researcher is mainly interested in identifying the top few items, we wondered about the value of spending time asking respondents to identify the worst item within each MaxDiff set. What would happen if we asked our robotic respondents only to select the best item within each set? The results somewhat surprised us. The value of asking respondents to indicate both best and worst within each set more than compensated for the 40% additional effort we suppose these "worst" questions add to the total interview time when interviewing human respondents. In a five item set (A,B,C,D and E) there are 10 possible 2-way comparisons. If we assume A is preferred to B and B is preferred to C and so on, then asking about only the best item will let us know A>B, A>C, A>D and A>E (4/10 comparisons). By asking about worsts as well, for only one additional question we also add B>E, C>E, and D>E (7/10 comparisons), leaving only the order relationship between B, C, and D unknown. The case of asking for bests when all respondents have same preferences that turns into the marked-bandit problem in Simchowitz et al. (2016). In that paper the authors give different algorithms for pulling the arms and upper and lower bounds on how many queries it takes to identify the top k items with high probability.

4.8. What about Double Adaptivity?

In 2006, one of the authors presented a paper on Adaptive MaxDiff that featured within-respondent adaptation Orme (2006) rather than what we have shown here in Bandit MaxDiff based on Thompson Sampling, which is an across-respondent adaptive approach. For the within-respondent adaptive procedure, items that a respondent indicates are worst are dropped from further consideration by that same respondent through a round-robin tournament until eventually that respondent's best item is identified. We thought adding this additional layer of within-respondent adaptivity on top of the Bandit MaxDiff approach could additionally lift its performance. To our surprise, this double-adaptive approach actually performed worse than Bandit MaxDiff alone in terms of hit rates for the globally best 3 or 10 items for the sample. After some head-scratching (and much code checking), we determined that the lack of improvement was due to degree of heterogeneity across the robotic respondents. For example, if we are interviewing a respondent who doesn't agree much with the overall population regarding which are the top items, it is detrimental to allow that respondent to drop from further consideration (due to judging them worst) what actually are among the globally most preferred items. It serves the greater good for each respondent to spend increased effort judging among the items that previous respondents on average have judged as potentially best.

4.9. Other Sampling Schemes

In Toubia and Florès (2007), which also looks at finding the top k items in a list they take the items, their "utility" (referred to as score) for there top preforming algorithm is the probability that a top item should be a bottom item and that a bottom item should be a top item. Then they perturb each score with a random normal variable that decreases in variance over respondents. They take the items the highest score to be in the next survey. Likewise we could also choose our L items to be the next survey this way instead of taking the top L items. For example for finding the top 3, to find the score of a top item you sample from the posterior distribution N times (where N is large) and the score is the number of samples below the estimated utility score of the current rank 3 item over N and for a bottom item it is the number of samples above the estimated utility score of the forth ranked item over N.

In Russo (2016), the author analyses different Bayesian algorithms for identifying the best arm. One point he makes is that Thompson Sampling is not effective for purely exploration problems. TS leads to oversampling the top item and not getting good enough estimates the second best items to conclude with high enough confidence that the top item is the top item. He gives a variation Top-Two Thompson Sampling which is better for the MAB problem. This is choosing the best arm with probability β and choosing the second best arm with probability $1-\beta$. One way to implement this in MaxDiff surveys is top pick the top 30 items and for each item with rank i with probability $1-\beta$ replace that item with the rank $i+30$ ranked item.

While our problem is purely exploration, the nature of MaxDiff survey causes us to show L items to every respondent, putting a limit on how much we are sampling the each item. As a result when k is less L we do not run in to the same estimation problems as in MAB problem. Though as we saw in PVR, adding randomness can improve the speed of parameter estimates in the same spirit as Russo's paper.

4.10. What about Sparse MaxDiff vs. Express MaxDiff?

Wirth and Wolfrath compared non-adaptive Sparse MaxDiff and Express MaxDiff in their 2012 paper at the Sawtooth Software Conference Wirth and Wolfrath (2012). We compared the results using our simulation and found a modest edge in performance for Sparse MaxDiff (Sparse ending at 80% for top 10 hit rate and express ending at 75%).

5. Conclusions and Future Research

Our results suggest that if your main purpose in using large item lists in MaxDiff is to identify the top items for the population (not individual-level estimates), then adaptive Bandit MaxDiff approaches can be 4x more efficient than standard Express MaxDiff designs. You are potentially wasting 75 cents of each dollar spent on data collection by not using Bandit MaxDiff.

Bandit MaxDiff leverages information from prior respondents to show more effective tradeoffs to later respondents (tending to oversample the stars, based on the Thompson Sampling mechanism). Even in the face of diabolically imposed misinformed starts (horribly unrepresentative first responders), the Bandit MaxDiff approach with our 20/10 split is extremely robust and self-correcting.

Although our simulations involve 120-item and 300-item tests, we expect that even greater efficiency gains than 4x (compared to standard Express MaxDiff designs) may occur with 500-item (or more) MaxDiff studies. For studies using 40 respondents, our simulation showed a 2x advantage in efficiency over fixed MaxDiff designs. Though not as dramatic, this is still a sizable boost.

Future research should test our findings using human respondents. Using an adaptive process that focuses on comparing best items may result in a more cognitively difficult task than a standard level-balanced, near-orthogonal approach. The greater expected within-set utility balance may lead to higher response error which may counteract some of the benefits of the Bandit adaptive approach. However, based on previous research Orme (2006) that employed within-respondent adaptivity, the additional degree of difficulty that the Bandit adaptive approach could impose upon individual respondents (owing to utility balance) would probably not counteract the lion share of the benefits we've demonstrated using simulated respondents.

This paper also introduces a framework that links conjoint methods and multi-armed bandit methods. Much like adaptive conjoint methods began with aggregate adaptation and then progressed to individual-level adaptive, so we propose an aggregate adaptive approach. But future could explore methods using fully heterogeneous models, and adapting within each individual. A partially pooled model will be useful here, as it is with other adaptive conjoint methods.

Our method relates to existing adaptive conjoint just as M-efficiency criterion is to D-efficiency. Unlike M-efficiency designs where the researcher decides the managerial weight of different factors a priori, we know which of the items should receive more weight. Instead, that is exactly what we want to learn actively.

We should note that as of this article's publication date, Sawtooth Software does not offer Bandit MaxDiff as a commercial tool. Sawtooth Software may perhaps one day soon offer Bandit MaxDiff as an option within its commercially available MaxDiff software. As for the authors, we look forward to this possibility as we've been especially impressed by the potential cost savings and increased accuracy!

Acknowledgments

References

Abernethy JD, Lee C, Tewari A (2015) Fighting bandits with a new kind of smoothness. *Advances in Neural Information Processing Systems*, 2197–2205.

Arora N, Huber J (2001) Improving parameter estimates and model prediction by aggregate customization in choice experiments. *Journal of Consumer Research* 28(2):273–283.

Cohen S, Orme B (2004) What's your preference? asking survey respondents about their preferences creates new scaling decisions. *MARKETING RESEARCH.* 16:32–37.

Cohen S, et al. (2003) Maximum difference scaling: improved measures of importance and preference for segmentation. *Sawtooth Software Conference Proceedings, Sawtooth Software, Inc*, volume 530, 61–74.

Hauser JR, Urban GL, Liberali G, Braun M (2009) Website morphing. *Marketing Science* 28(2):202–223.

Hendrix P, Drucker S (2007) Alternative approaches to maxdiff with large sets of disparate items–augmented and tailored maxdiff. *SAWTOOTH SOFTWARE CONFERENCE*, 169.

Kalai A, Vempala S (2005) Efficient algorithms for online decision problems. *Journal of Computer and System Sciences* 71(3):291–307.

Kujala J, Elomaa T (2005) On following the perturbed leader in the bandit setting. *International Conference on Algorithmic Learning Theory*, 371–385 (Springer).

Louviere JJ, Woodworth G (1991) Best-worst scaling: A model for the largest difference judgments. *University of Alberta: Working Paper* .

Marley AA, Louviere JJ (2005) Some probabilistic models of best, worst, and best–worst choices. *Journal of Mathematical Psychology* 49(6):464–480.

Marley AA, Pihlens D (2012) Models of best–worst choice and ranking among multiattribute options (profiles). *Journal of Mathematical Psychology* 56(1):24–34.

Orme B (2006) Adaptive maximum difference scaling. *Technical Paper available at www. sawtoothsoftware. com* .

Russo D (2016) Simple bayesian algorithms for best arm identification. *arXiv preprint arXiv:1602.08448* .

Schwartz EM, Bradlow E, Fader P (2015) Customer acquisition via display advertising using multi-armed bandit experiments. *Ross School of Business Paper* (1217).

Scott SL (2010) a modern bayesian look at the multi-armed banditby steven l. scott: Rejoinder. *Applied Stochastic Models in Business and Industry* 26(6):665–667.

Scott SL (2015) Multi-armed bandit experiments in the online service economy. *Applied Stochastic Models in Business and Industry* 31(1):37–45.

Simchowitz M, Jamieson K, Recht B (2016) Best-of-k bandits. *arXiv preprint arXiv:1603.02752* .

Thompson WR (1933) On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25(3/4):285–294.

Toubia O, Florès L (2007) Adaptive idea screening using consumers. *Marketing Science* 26(3):342–360.

Toubia O, Hauser JR, Simester DI (2004) Polyhedral methods for adaptive choice-based conjoint analysis. *Journal of Marketing Research* 41(1):116–131.

Urban GL, Liberali G, MacDonald E, Bordley R, Hauser JR (2013) Morphing banner advertising. *Marketing Science* 33(1):27–46.

Wirth R, Wolfrath A (2012) Using maxdiff for evaluating very large sets of items. *SAWTOOTH SOFTWARE CONFERENCE*, 59–78.