

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

Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.

# Best Arm Identification and Active Learning for Choice Experiments: Applications to Large-scale Adaptive MaxDiff and Idea Screening PRELIMINARY! DO NOT DISTRIBUTE

Eric Schwartz

University of Michigan, ericsch@umich.edu,

Alexander Zaitzeff

University of Michigan, azaitzef@umich.edu,

Kenneth Fairchild

Sawtooth Software, ,

Bryan Orme

Sawtooth Software, ,

Based on an earlier 2015 Sawtooth Conference Paper

with Kenneth Fairchild, Bryan Orme, Eric Schwartz.

Marketing research often measure consumers' preferences in order to identify their most preferred items, such as product features, marketing messages, or innovations, among a large set of hundreds of candidates. Common techniques are idea screening and choice tasks, including conjoint and best-worst scaling (MaxDiff).

.... **to be continued** ...

Best arm identification.

We implement the methods using MaxDiff survey from a large consumer packaged goods manufacturer. We show our approach outperforms current methods, but we illustrate under which conditions it performs even better; it exhibits even greater improvement for larger problems.

*Key words:* idea screening, maximum-difference surveys, adaptive conjoint, active learning, multi-armed bandit, best-arm identification, Bayesian bootstrap, best-worst scaling, Thompson Sampling, uncertainty minimization

## 1. Introduction

Firms collect more customer data than ever. While much of that is behavioral data (e.g., purchases, clicks, visits, etc.), market research still 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 more efficiently – more information gained per respondent used.

One common managerial decision driving such data collection is to select which are most preferred items – such as, ideas, attributes, messages – among a large set of hundreds of possible items. For instance, consider these illustrative managerial decisions:

1. *Idea screening.* Technology companies have long invited consumers to provide ideas for new products and improvements. Automotive companies Complex products, such as, self-driving cars are may contain hundreds of novel features and are sold with different add-ons, which are bundles of features. So consumer surveys could reveal how the best features are preferred, to help the firm decide which features can be included in different models.

2. *Marketing communications.* Consumer packaged goods companies use marketing research, after the products are developed, to decide which consumer benefits they should emphasize. Out of hundreds of potential benefits, they have to select a small number of them for messages.

3. *Product offerings.* Retailers are increasingly able to respond to demand and consumer preferences quickly by changing their available product offerings. In anticipation of a new fashion season, an ecommerce retailer may survey customers to identify most preferred styles and lines, and then select the range of items to make available for purchase.

These settings have a common challenge: identify the best set of items out of many. This problem setting is distinct from closely related problems but existing methods do not suffice due to scale and objective.

Typical choice experiment methods, such as conjoint, are infeasible, too costly, 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.

We focus on settings where the objective is identifying a top set of items as quickly, cheaply, and accurately as possible. Identifying a top set is not the same objective as estimating all parameters as precisely as possible, as in conjoint. In this work, we illustrate how those objectives diverge. So even with improving computation and larger budgets, this problem has a distinct objective and demands a tailored methods.

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 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. Which is the next question, or set of product profiles, to best reduce overall uncertainty?

But the market researcher’s ultimate goal is typically not to only obtain precise estimates of utility partworths 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 new adaptive choice experiment methods. Our approach departs from traditional adaptive survey methods like adaptive conjoint analysis (ACA): they aim to improve the precision of each of the parameters, but we aim to identify the most preferred items, and as a by product, we estimate those best items more precisely than less preferred items.

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.

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 and AL methods. We build on an increasingly accepted method of solving multi-armed bandit problems, Thompson Sampling. We introduce multiple versions: MaxDiff-**TS**, its generalization MaxDiff  $\epsilon$ - $\delta$ -**diffuse TS**, and  $\epsilon$ - $\delta$ -**TS-thres**. **(eric) Should we discuss  $\epsilon$ - $\delta$ -diffuse TS here?** The  $\epsilon$ - $\delta$ -**diffuse TS** nests traditional TS. We combine ideas from  $\epsilon$ -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. We also build on **max-misclass** and **max-uncert** with perturbations, common in active learning settings. **(eric) Should we first introduce max-misclass and max-uncert here?**

Our problem setting, identifying the top set of items with choice data collection, serves as an illustration of a broader framework. More broadly, we align the choice data collection process with managerial objectives. By aligning the two, we can improve efficiency and scalability for big data collection via choice experiments. We find we can “get more for less,” by improving precision only where it matters for decision making, using a smaller sample size compared to existing methods.

While our work can generalize, we demonstrate its viability empirically with MaxDiff, as it is an increasingly popular and important choice experiment method. The adaptive idea screening problem is sufficiently similar. And we will consider the general choice-based conjoint case in our discussion section.

We also consider the practical implementation of the algorithms. The scalability and speed of each is considered. We provide alternative versions that are faster but may sacrifice theoretical statistical validity. Nevertheless, some fast heuristics perform empirically.

We also contribute ideas to the literature and practice. We frame choice task data collection as best arm identification multi-armed bandit problem and active learning problem. During the process of data collection, we adaptively yet strategically learn preferences in order to maximizing our accuracy of identifying the best set of items. The faster we begin identifying those best items, the more we include them in questions, improving the precision of estimates. Most importantly for practice, we can identify those items faster,

stopping data collection sooner, saving money in the form of number of respondents and time.

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 formalizing the discrete choice model of MaxDiff analysis, we introduce our 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.

**(eric)** Note on flow: Transition from edts to active learning by talking about adaptive conjoint. polyhedral method isn't quite right because including an item doesn't bisect the uncertainty ellipsoid. But picking the one with maximum certainty could still be good. And MaxError is natural too as it links to the actual decision.

## **2. Framework and literature**

### **2.1. Multi-armed bandit, best-arm identification, and active learning**

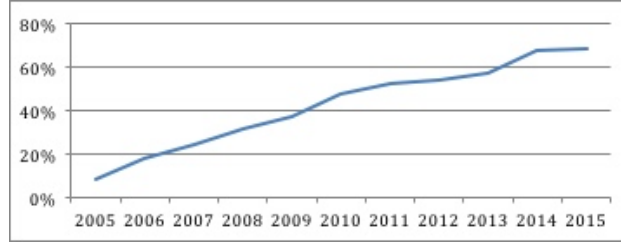
Our approach builds upon on state-of-the-art multi-armed bandit and active learning algorithms, both from machine learning and reinforcement learning literatures.

MAB problems balance learning and earning. MAB 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 multivariable parametric models is Thompson Sampling.

However, our problem differs from extant bandit applications in a number of ways. For a bandit problem the reward is observed immediately (e.g., click, acquisition, or purchase). While we have immediate observations (choice among a set), that is not our reward. Instead, whether we correctly identify the best items cannot even be evaluated in real-time during the data collection process.

While our problem is not exactly the typical bandit problem, it is more similar to a known MAB variant: best arm identification multi-armed bandit problem. Identifying the best of a set of arms has historically received less focus, but has recently attracted theoretical study in the computer science literature (Gabillon et al. 2012, Kalyanakrishnan et al. 2012,

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



Kaufmann et al. 2016, Kaufmann and Kalyanakrishnan 2013, Russo 2016). This is in part due to the increase in rigorous study of the practice of A/B testing in practice.

The best arm identification problem is, in part, distinct because it is a pure explore setting, where as the canonical bandit calls for explore-exploit tradeoff. While both are adaptive, the explore-exploit problem deals with maximizing earning while learning, but the pure explore setting is just about learning during the data collection.

Like the literature, we find bandit algorithms serve as a good starting point heuristic to achieve our learning objective. But the learning-focused

The best arm identification problem is more similar to a classic active learning problem. Active learning, while an established areas of research in the statistics and machine learning literature, has hardly appeared in marketing. One exception is Dzyabura et al. 2011. However, they do not consider active learning from the perspective of the manager, and instead consider active learning algorithms serve as decision making heuristics for individual consumers.

We also draw on ideas in closely related area, active learning. Recent theoretical advances, show the value of adding perturbations, such as, Follow The Perturbed Leader (FTPL) (Kalai and Vempala 2005). Others 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 resolves the learning-and-earning tradeoff for the stochastic MAB setting and improves pure learning for the active learning setting, but it also goes further, and does so in the adversarial setting. This suggests it is a strategy more robust to the non-random dynamic changes in an environment.

## **2.2. Idea Screening and 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).

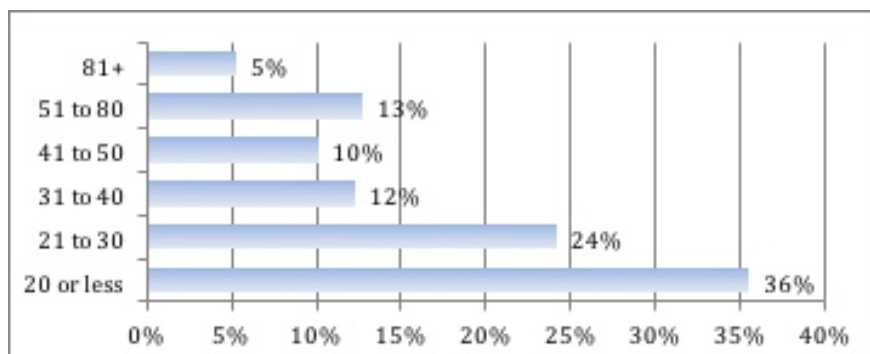
MaxDiff, while it is a distinct type of choice experiment, is closely related to conjoint. A key difference is that it involves both best and worst choices instead of only one. But in its most common form, MaxDiff may be thought of as a one-attribute CBC study with many levels.

## **2.3. Studying More Items with MaxDiff: Survey of Market Research Practitioners**

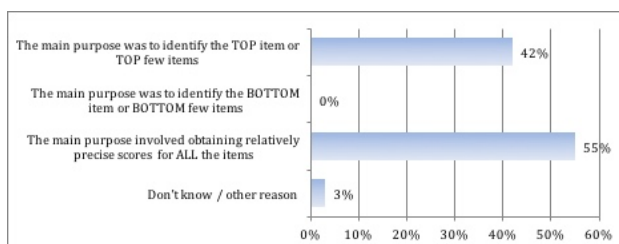
MaxDiff has proven so useful that market researchers increasingly find reasons to use MaxDiff for a large number of items. How many is a large number of items? In their 2007 paper, Hendrix and Drucker described “large sets” as about 40 to 60 items, proposing variants to MaxDiff called Augmented and Tailored 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.

**Figure 2** Maximum Number of Items Studied via MaxDiff during 2015. N = ??, Mean= 40, Median=30, Maximum=400



**Figure 3** Main Purpose for MaxDiff Study with 41+ Items



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 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.

#### **2.4. 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).

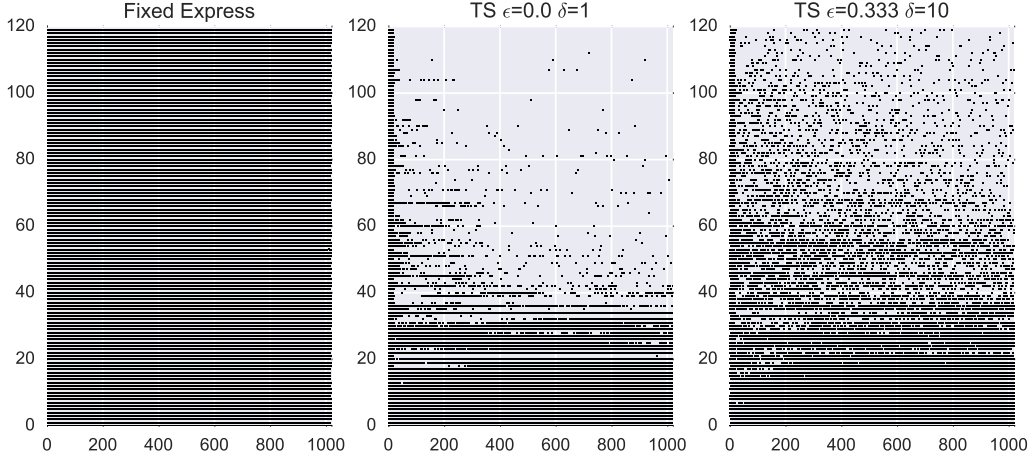
The traditional MaxDiff design approach shows each item an equal number of times across all respondents x tasks. However, if the main goal is to identify the top few items for the sample, after the first, say 20, respondents it seems reasonable to start paying attention to the already-collected MaxDiff 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

**Figure 4** Respondent-by-item counts



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 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.

**Emphasize our goal is to identify the top items.**

### 3. Adaptive MaxDiff

We formalize the proposed procedure in two 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 adaptive techniques as the natural method for solving the pure exploration problem.

#### 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  $\varepsilon_z$  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

$$B_S(x) = \frac{e^{u(x)}}{\sum_{z \in S} e^{u(z)}}$$

$$W_S(y) = \frac{e^{u(y)}}{\sum_{z \in S} e^{u(z)}}$$

$$WB_S(x, y) = \frac{e^{u(x)-u(y)}}{\sum_{r, s \in S, r \neq s} e^{u(r)-u(s)}}$$

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. 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- $N$ ) 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 negative likelihood of the choice data as a multinomial logit with choice probabilities in vector notation as follows,

$$-\Pi \frac{\exp\left(\begin{bmatrix} Y_B \\ Y_W \end{bmatrix} \theta\right)}{\exp\left(\begin{bmatrix} X_B \\ X_W \end{bmatrix} \theta\right)}$$

The rows in this matrix representation represent every respondent-task-item combination,  $N * J * |S|$ , repeated twice. The link between the models of best choice and worst choice is the parameter  $\theta = \{\theta_1, \dots, \theta_m\}$ . This common parameter vector represents the overall utility of each item  $1, \dots, m$ . For a more positive  $\theta_i$ , the item  $i$  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  $\theta$  only represents the utility of item  $i$  being included versus excluded. If  $\theta_i > \theta_{i'}$ , then we say item  $i$  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.

The log likelihood can be written in summation notation where  $S_n$  denotes the  $n$ th set of choices as follows

$$LL(\theta) = - \sum_{n=1}^N \sum_{x \in S_n} (Y_{B_{S_n}}(x) \log \frac{e^{\theta_x}}{\sum_{z \in S_n} e^{\theta_z}} + Y_{W_{S_n}}(x) \log \frac{e^{-\theta_x}}{\sum_{z \in S_n} e^{-\theta_z}})$$

In our work we find  $\theta$  by minimizing the negative log likelihood using Newton-Rapson.

#### 4. Adaptive MaxDiff

For each respondent we select  $L$  items, and we generate a MaxDiff design of  $J$  best-worst tasks, each with a choice set of  $|S|$  items. We use  $L = 20$ ,  $J = 12$ ,  $|S| = 5$  as a default since it is a standard number of items used in MaxDiff studies Wirth and Wolfrath (2012). We begin by selecting the  $L$  items uniformly. After an initial number of respondents, we are still uncertain about the each parameter value, so we continue collecting data. But we do not need to reduce that uncertainty equally for each one, so we begin adapting. To translate our current beliefs about parameters into action, we can draw from an exact or approximate posterior distribution and use an adaptive method, explained in detail in the following sections and summarized in table 1, to decide what questions to show the next respondent. In the case of the multinomial logit, we can use MCMC to obtain samples, sample from asymptotic distribution via MLE implied by the estimated mean and standard errors, or as we do in the empirical application, use Bayesian bootstrapping. Using Bayesian Bootstrap a draw  $u$  is made in the following way.

$$\begin{aligned} \beta_1, \beta_2, \dots, \beta_N &\sim \exp(1) \\ LL(\theta; \beta) &= - \sum_{n=1}^N \beta_n \sum_{x \in S_n} (Y_{B_{S_n}}(x) \log \frac{e^{\theta_x}}{\sum_{z \in S_n} e^{\theta_z}} + Y_{W_{S_n}}(x) \log \frac{e^{-\theta_x}}{\sum_{z \in S_n} e^{-\theta_z}}) \\ u &= \arg \min_{\theta} LL(\theta; \beta) \end{aligned}$$

The adaptive method then use these draws to decide the next  $L$  items to show.

There are two practical issue we highlight. First, We want to avoid repeating extremely similar questions to the same respondent. A natural consequence of adaptive methods is convergence: as the sample size grows, certain items 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, after drawing a fixed number of items (e.g., 20 or 30) to show each respondent, those draws of 20 items are shown to each respondent in a balanced, near-orthogonal design, leading to a palatable 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 20 items for each respondent is adaptive 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 20. We also avoid repetition. A particularly annoying alternative is to draw independent samples ranking each choice task for the same respondent. This results in very repetitive choice tasks as parameter estimates. The second is frequency of updating the data. If people are taking the survey sequentially then all the data from the previously respondents can be used to generate the surveys for the newcomer. Often this is not the case, as an alternative you can update the data once every  $b$  people and decide the questions for the next  $b$  people. We call  $b$  the batch size and in our empirical analysis we let  $b = 20$ .

Both of these considerations avoid is latency during a survey adn across respondents. Since all the questions for the next  $b$  people are set, no computation takes place before or inbetween questions for a respondent. This allows for these methods to be put into practice.

In summary, given a adaptive method we serve up questions in the following way

$$S = \#\{\mathcal{S}\}$$

We briefly describe to simple benchmarks before moving on to describe the more sophisticated proposed methods.

**4.0.1. Fixed Express MaxDiff .** We uniformly randomly drew  $L$  of  $K$  items (without replacement) to show to each respondent. This achieves balance across items and respondents. For instance, in a problem with  $L = 20$  of the  $K = 120$ , then each item appears  $\frac{J*S}{L} = \frac{12*5}{20} = 3$  times per respondent, on average.

---

**Algorithm 1** General adaptive method

---

- 1: Initialize
  - 2:
  - 3: Collect new data from respondents
  - 4: Estimate model parameters
  - 5: Select next questions for respondents
  - 6: Check stopping rule
- 

- 
- 1: given:  $K, L, J, S, b$
  - 2: Initialize first set of questions by sampling  $L$  items uniformly
  - 3: Design MaxDiff questionnaire covering  $L$  items with  $J$  tasks of  $S$  questions each,  $\text{MDDesign}(L, S, J)$ , to the next batch of  $b$  respondents.
  - 4: Collect new data from respondents, currently with  $n = b * t$  respondents
  - 5: Bayesian Bootstrap sample weight replacement using weights  $(\alpha_1, \dots, \alpha_n) \sim \text{expon}(1)$  obtaining some subset of  $n$  previous respondents.
  - 6: Estimate model parameters.  $\theta_t$ . Obtain estimated utilities  $u = (u_1, \dots, u_K)$ .
  - 7: Choose  $L$  items using draws from the empirical posterior
  - 8: Act Select next questions for respondents
- 

**4.0.2. Algorithm  $\epsilon$ -greedy** As a baseline for the adaptive methods will we test the  $\epsilon$ -greedy. Let  $\theta$  be the current estimated parameters. Take the  $(1 - \epsilon)L$  items with the greatest  $\theta$  value and choose the remaining  $\epsilon L$  uniformly from the remaining items. For our empirical analysis we use  $\epsilon = \frac{1}{4}$  but tested others. Note that greedy alone  $\epsilon = 0$  would always serve the top  $L$  items with the highest estimated average utility to the next respondents.

**4.0.3. Algorithm MaxDiff TS** 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, this 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.

The algorithm learn 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



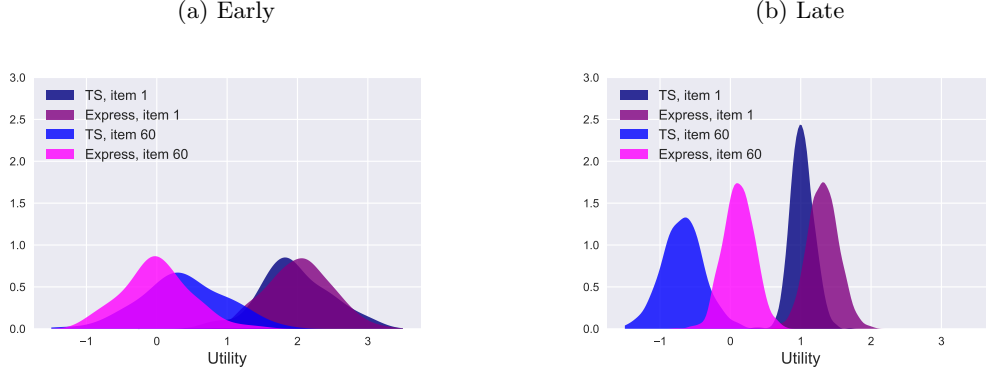
**Table 1** Summary of Adaptive MaxDiff Algorithms

| Algorithm   | Short Description   |
|---|---|
| MaxDiff TS  | Sample from the current posterior distribution and serve up the $L$ items with the highest sampled utility.   |
| MaxDiff $\epsilon$ -Diffuse TS                          | Sample from the current posterior distribution serve up the $(1 - \epsilon)L$ items with the highest sampled utility and sample from the current diffuse posterior distribution and serve up the $\epsilon L$ items with the highest sampled utility not in $(1 - \epsilon)L$ . |
| $\epsilon$ -greedy                                      | Take the $(1 - \epsilon)L$ with the greatest current estimated $\theta$ . Take the remaining $\epsilon L$ uniformly from the remaining items.   |
| TS closest to the threshold                             | Take the $L$ items that have sampled utility closest to $\frac{u_k + u_{k+1}}{2}$ .   |
| $\epsilon$ -Diffuse TS closest to the threshold         | Take the $(1 - \epsilon)L$ (from posterior) and $\epsilon L$ (from diffuse posterior) items that have sampled utility closest to $\frac{u_k + u_{k+1}}{2}$ .  |
| $\epsilon$ -greedy closest to the threshold             | Take the $(1 - \epsilon)L$ items that have estimated utility closest to $\frac{\theta_k + \theta_{k+1}}{2}$ and $\epsilon L$ items uniformly from the remaining.  |
| Misclassification Minimization with random perturbation | Take the $L$ items that have most likely been misclassified (bottom items that should be top items and vice-versa). Add perturbation to the that probability.   |
| Greatest Uncertainty with random perturbation           | Take the $L$ items whose probabilities of being a top item are closest to 50%. Add perturbation to the that probability.  |

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.

One 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

**Figure 5** MaxDiff TS Illustration (eric) AZ will do this using the updated version of TS with Bayesian Bootstrapping(alexander) to ES: Feedback

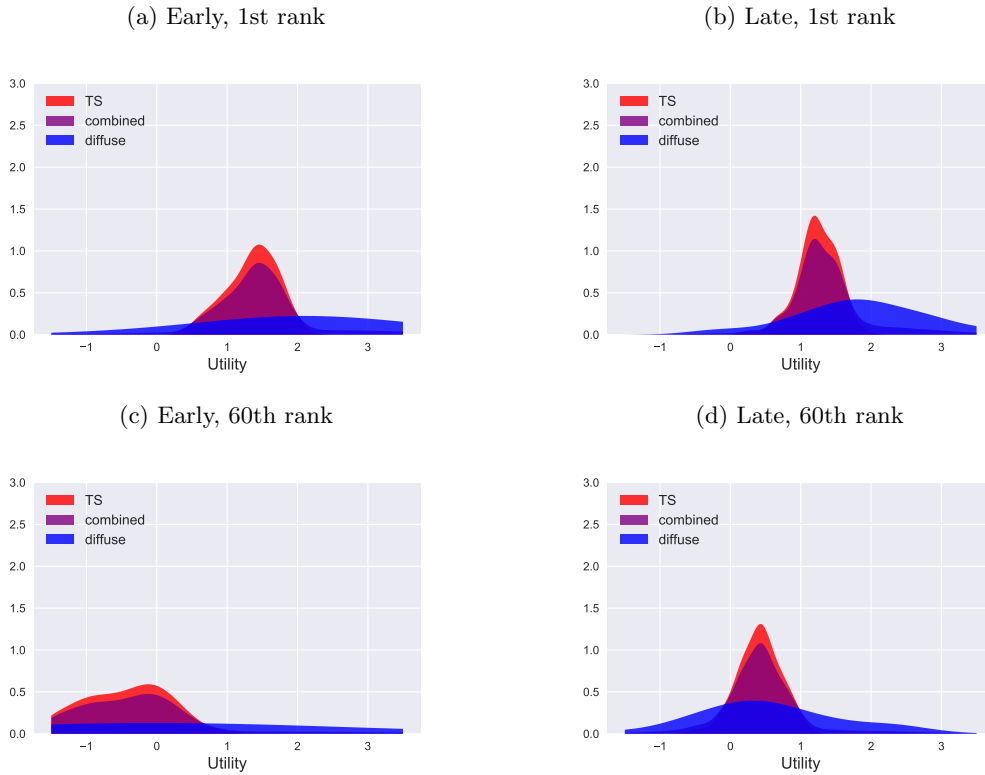


this continues, the uncertainty is reduced around these poor items to reveal there are many other items with probability of being better. By sampling from the joint belief distribution of item utilities, we will be less likely to draw those poor items. One concern is that such sampling could be too aggressive.

**4.0.4. Variant-MaxDiff exploration-diffuse ( $\epsilon$ - $\delta$ ) TS Key: Distinguish Point Estimate vs. Bayesian Bootstrap.** With point estimate  $\delta$  is directly variation inflation factor for the estimator. But with Bayesian Bootstrap,  $\delta$  is a percentage of the data, indirectly inflating the estimator's variance. 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 epsilon comes from the popular  $\epsilon$ -greedy, which mixes greedy with uniform sampling. But instead of being greedy for  $1 - \epsilon$  samples, this algorithm, follows regular Thompson Sampling. And on  $\epsilon$  samples, it does not necessarily sample items uniformly, but it does explore more than regular Thompson Sampling, by drawing parameters a more diffuse posterior distribution, where the variance inflation is controlled by  $\delta$ .

The  $\epsilon$ - $\delta$ -diffuse TS is a generalized version of TS, which directs nests regular TS. This is analogous to the way a two-component mixture model nests a pooled model without segments. As the  $\epsilon \rightarrow 0$ , the diffuse distribution is never used, so the algorithm collapses to regular TS. As  $\delta \rightarrow 1$ , the diffuse distribution becomes equivalent to the regular (non-diffuse) distribution.

**Figure 6** Illustration of **TS** v  $\epsilon$ - $\delta$ -diffuse **TS** for two items at two points in time. (eric) Show a 4 plots in a 2x2 in this figure. Show the **TS** ,  $\epsilon$ - $\delta$ -diffuse **TS** belief distributions and show the underlying diffuse distribution that is one component of  $\epsilon$ - $\delta$ -diffuse **TS** . If it looks good, include expresstoo, but don't worry. Show this for 1st ranked item (top row of plots) and 60th ranked item (bottom row) to show . Show for early beliefs after respondent 40 (left column of plots) and late beliefs after respondent 500 (right) show learning. If it looks good for respondent 40, great; otherwise, find earlier or a little later respondents where this is illustrative.(alexander) to ES: Feedback, I did resp 60

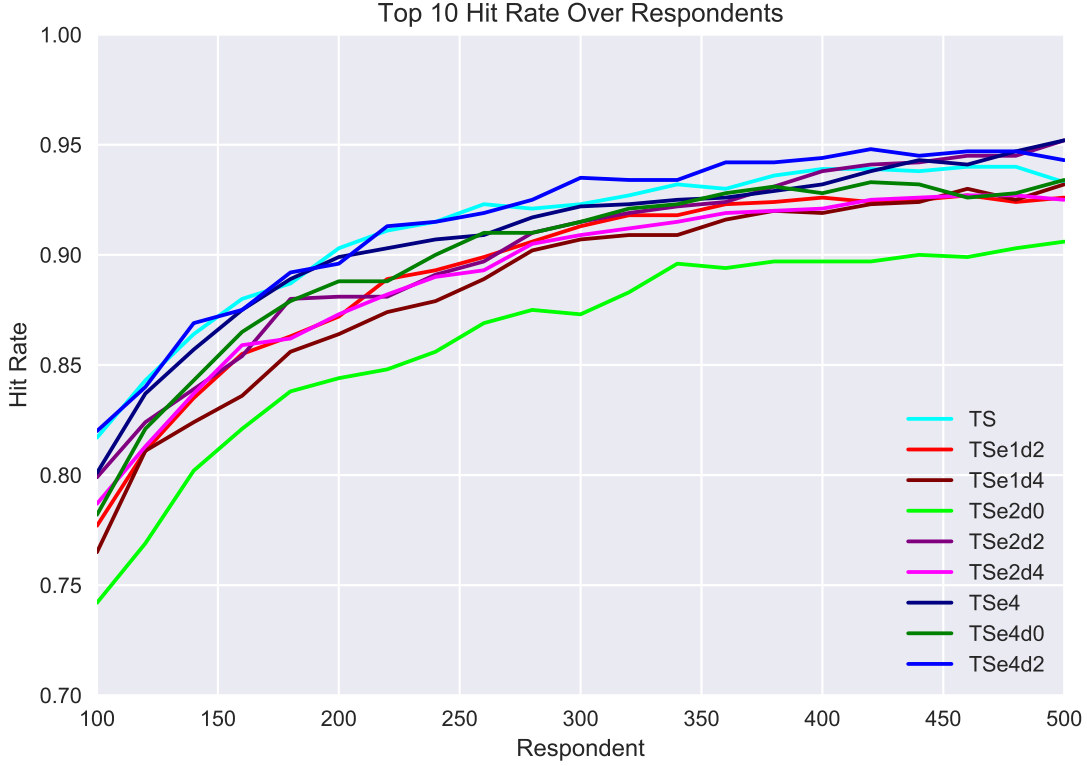


The  $\epsilon$ - $\delta$ -diffuse **TS** version intuitively hedges its bets on the best items . As in illustration, instead of sampling 20 items all from **TS** , we only sample 15 of 20 items using standard **TS** (hence,  $L = 20$ ,  $\epsilon = 1/4$ ), and the remaining 5 of 20 items are drawn using **TS** with a much more diffuse prior (subsampling  $\delta=.25$  of the data with replacement and drawing  $u$  as a Bayes bootstrap of the subsampled data).

Figure 6 illustrates how  $\epsilon$ - $\delta$ -diffuse **TS** differs from **TS** .

For  $\epsilon$ - $\delta$  case, its density does not become a spike as quickly as it does without  $\epsilon$ - $\delta$  . This speed is controlled through two parameters: epsilon, the proportion of items sampled from the diffuse distribution, and  $\delta$ , factor increasing variance for the diffuse distribution. The larger the  $\epsilon$  and the smaller the  $\delta$ , the more exploration and slower the algorithm settles on its set of items.

Figure 7 Respondent-by-item counts



We show how  $\epsilon$ - $\delta$ -diffuse TS performance varies with different values of  $\epsilon$  and  $\delta$  in Section 5. (eric) AZ will re-run tests on  $\epsilon$ - $\delta$ -diffuse TS for different combinations of values of  $\epsilon = 1/2, 1/4, 0$  and  $\delta = 1, 1/2, 1/4, \text{effectively uniform}$ , almost do the whole crossproduct except avoid redundancies. (alexander) to ES: Which ones do you want to keep? see figure 7 For our empirical analysis, we use  $(\epsilon = \frac{1}{4}, \delta = \frac{1}{4})$ . Also tested were  $(\epsilon = \frac{1}{2}, \delta = \frac{1}{4})$  and  $(\epsilon = 1, \delta = \frac{1}{4})$  which performed on par with  $(\epsilon = \frac{1}{4}, \delta = \frac{1}{4})$ . Our recommended ranges for the parameters are  $\frac{1}{5} \leq \epsilon \leq \frac{1}{2}, \frac{1}{4} \leq \delta \leq \frac{1}{2}$ .

**4.0.5. 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 will do that to find the top  $k$ . For a draw  $u$  calculate the cutoff  $c = \frac{u_k + u_{k+1}}{2}$  a score  $s_i = |c - u^i|$  for every item. Then take the  $L$  items with the lowest score. Similarly, we can define  $\epsilon$  diffuse version here. For two draws  $u_R$  (posterior) and  $u^D$

(diffuse posterior) and calculate scores  $s_R$  and  $s_D$ . Take  $(1 - \epsilon)L$  items with lowest  $s_R$  scores and the  $\epsilon L$  items with the lowest  $s_D$  scores not already included. We will also test the  $\epsilon$  - greedy version where the scores are  $s_i = |c - \theta^i|$  for  $c = \frac{\theta_k + \theta_{k+1}}{2}$  you take the  $((1 - \epsilon)L)$  lowest scores and the rest uniformly.

|        | utilities |   |   |   |   |   | rank ordering |     |     |     |     |        |
|--------|-----------|---|---|---|---|---|---------------|-----|-----|-----|-----|--------|
| item : | a         | b | c | d | e | f | max(1)        | (2) | (3) | (4) | (5) | (6)min |
| draw1: | 6         | 7 | 7 | 8 | 8 | 9 | -->           | f   | d   | e   | c   | b a    |
| draw2: | 7         | 5 | 5 | 9 | 8 | 6 | -->           | d   | e   | a   | f   | b c    |
| draw3: | 6         | 4 | 5 | 8 | 7 | 5 | -->           | d   | e   | a   | c   | f b    |

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

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

**4.0.8. Variant Random Perturbation** From Toubia and Florès (2007), after calculating the scores for a respondent, you perturb the scores by a normal vector with mean zero and variance  $c \frac{1}{n_i}$ , Where  $n_i$  is the number of questions that has had item  $i$  and  $c$  is some constant, we take it to be  $\frac{1}{10}$  (also tested ws  $c = \frac{1}{2}$  which preformed similar). Using the pertubed scores Serve up the  $L$  items accordingly. Our recommended range for the parameter is  $.01 \leq c \leq 1$

The reason for using a variance that decreases over time rather than a fixed variance can be found in Toubia and Florès (2007) where the author derived insight from genetic algorithms. This method avoids misclassifying an item that will not be sampled further.

Additionally if you are using a batch size greater than one, Misclassification Minimization and Greatest Uncertainty would show the same items to all of the respondents. After getting scores that are the same for the next  $b$  respondents adding the perturbation vector

adds variety to the items shown. This is useful because using Bayes bootstrapping for estimating probabilities required in Misclassification Minimization and Greatest Uncertainty takes a good deal of time since you are optimizing the negative log likelihood a large number of times.

## 5. Empirical Analysis: Main Results

We compared our proposed set of adaptive approaches to existing adaptive and non-adaptive strategies. We use simulated choice data based on inferred preferences from an actual MaxDiff survey.

### 5.1. Data

The data come from a survey conducted by a Procter & Gamble with Sawtooth Software. The study involved 981 respondents and 120 items. The subject matter and exact item text was hidden for confidentiality purposes. As is common in MaxDiff surveys, the items represented product features and benefits. The questions were came from a sparse MaxDiff study, so this was a fixed non-adaptive design, with balance across all items and respondents.

Instead of raw choice data, 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 and became our data-generating process for use in our respondent simulations. Therefore 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 distributed error.

### 5.2. Simulation Setup

To measure performance, we consider a range of metrics: the estimated rank order of items' utilities, the hit rate of whether the estimated top-ranked set of items is correct, the value of the utilities of that estimated top set, and the variability around the estimated utilities. We obtain these measures by comparing estimates to true values. For each batch of respondents, we ran aggregate logit and compare the current rank order of the estimated utilities to the true rank order for the known true utilities, which are the aggregate average across

all respondents’ true individual-level utilities. We ran the simulations 100 independent runs to obtain a distribution of measures.

Since we motivated the problem with identifying a top set of items, our primary performance measure is **top  $k$  hit rate**. **Top  $k$  hit rate**: What percent of the true top  $k$  items appear in the estimated top  $k$  set? For instance, for the top-10 hit rate, if the estimated scores identified 7 of the true top 10 items, irrespective of order, the hit rate was 70%. We use  $k = 3, 10, 20, 40$ .

Hit rate is a natural consideration in an active learning problem. It evaluates the quality of the adaptive learning procedures with respect to the eventual decision. Hit rate is also related to regret in a typical multi-armed bandit problem. It reflects how far we are from always selecting the truly best set of arms for all time periods. Again, this is only related but not equivalent to a bandit setting, since it is not exactly an observed reward in the sense of the usual bandit setting (e.g., clicks, purchases).

The difficulty of identifying the truly best items is related to the differences among those top item utilities (Figure 8). With 120 items in the dataset, we observed the true preferences for the approximately the top 15 were close in terms of utility (within 1.0 on logit scale). There were no runaway winners. Due to how tightly the top items are clustered, the hit rate measures we employed were quite discriminating between competing methods.

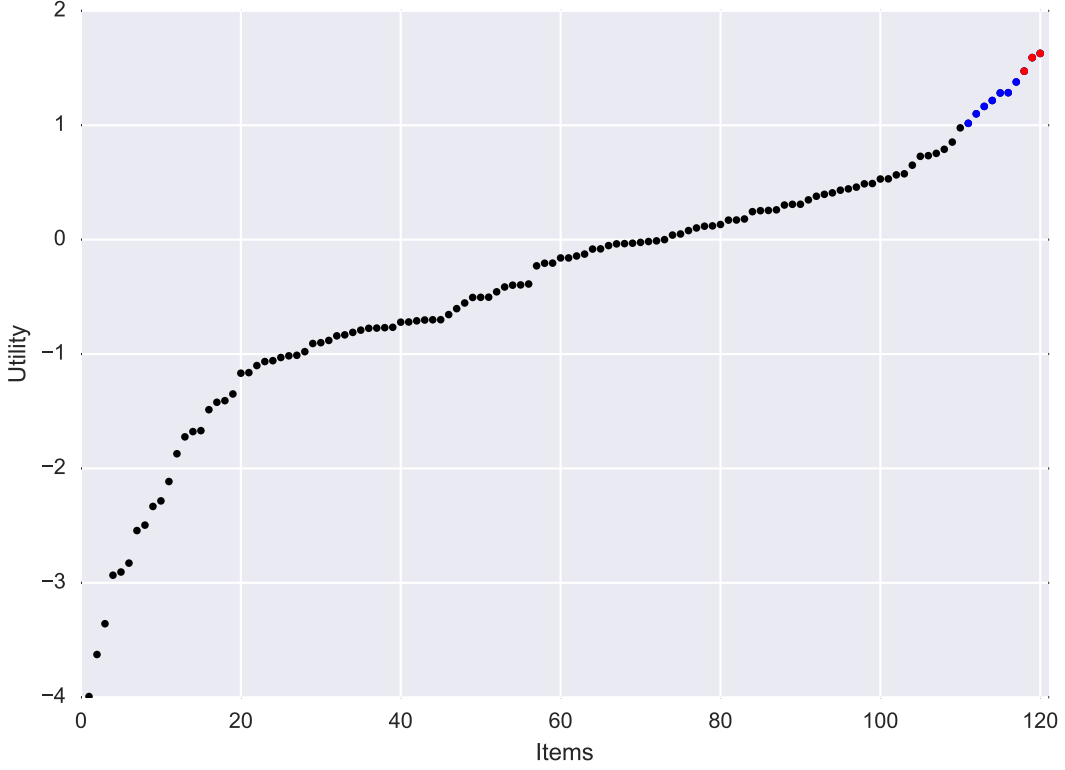
We describe the *base setting*. We simulate the process of collecting survey data from  $N = 500$  respondents, in batches of  $b = 20$  respondents per period, using bootstrap sampling, with replacement from the original 981 individuals. We consider the first 20 respondents to be the initial group, which always received  $L = 20$  items uniformly selected for all methods. All adaptive methods only begin after the 20th respondent. Each simulated respondent completed  $J = 12$  choice sets (best-worst tasks), where each set included  $S = 5$  items.

As described earlier, we tested the different adaptive approaches found in Table 1. For our base settings, we use  $(\epsilon = \frac{1}{4}, \delta = \frac{1}{4})$  for  $\epsilon - \delta$ TS methods. The natural benchmark is the existing fixed MaxDiff approach.

### 5.3. Results: Greedy, Thompson Sampling (TS), $\epsilon - \delta$ TS vs Fixed Express

The first result shows adaptive methods, even simple ones, are better than static ones. But a simple greedy algorithm, which adapts, but does not explicitly incorporate learning, is not good enough. Table 9 shows how TS improves substantially over greedy, and  $\epsilon - \delta$  TS adds even more improvement in hit rate.

**Figure 8** True utilities of each item learn. The plot shows the rank ordering and values of the utilities on the logit scale for all 120 items, highlighting the top 10 (blue) and top 3 (red). These are the means for unobserved data-generating process.



For example, after the first 200 respondents, the Fixed Express Design obtains a top 3 hit rate of 60% whereas the TS-based approaches achieve a top 3 hit rate of about 80%. Practically, if a marketing researcher wants a hit rate of 80%, for example, then these methods can achieve that with a smaller sample size. A good adaptive method is at least *three times more efficient* than the standard Fixed Express MaxDiff approach. This holds for the different values tested in the basic setting.

Figure 10 shows the way the  $\epsilon$  and  $\delta$  affect the performance.

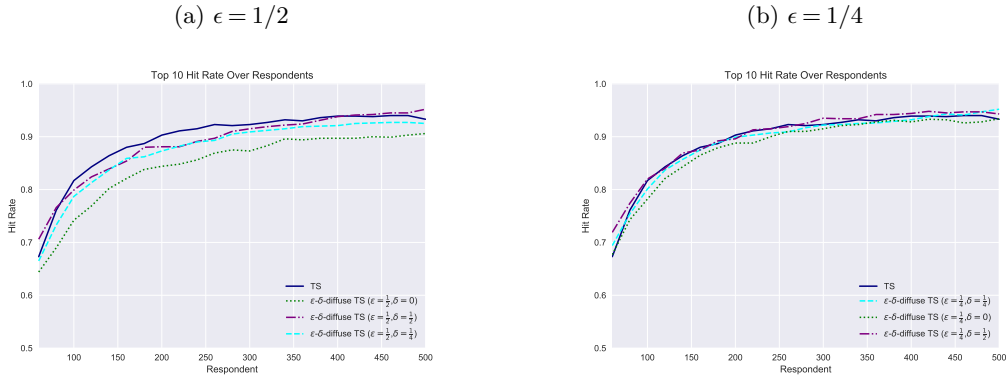
We will later show the performance gap between **TS** and  $\epsilon$ - $\delta$ -diffuse **TS** widens dramatically when considering non-stationary settings, such as with a misinformed starts in Section 6.



**Figure 9** Adaptive vs Static: TS and  $\epsilon$ - $\delta$ -diffuse TS improve hit rate. (eric) Only show express,  $\epsilon$ -greedy, TS,  $\epsilon$ - $\delta$ -diffuse TS here for  $k=3$ ,  $K=120$ ,  $L=20$ . Performance is hit rate over respondents. This is a placeholder plot. (alexander) DONE



**Figure 10** How  $\epsilon$ - $\delta$ -diffuse TS varies with  $\epsilon$  and  $\delta$ . (eric) Show typical plot of hit rate over respondents. Show many  $\epsilon$ - $\delta$ -diffuse TS versions:  $\epsilon = 1/2$  (left) and  $\epsilon = 1/4$  (right), with multiple lines in each plot,  $\delta = 1/2$  (one color) and  $\delta = 1/4$  (another color). And in each of the plots also show regular TS ( $\epsilon=0, \delta=1$ ). The plots below are just placeholders.



#### 5.4. Results: Thresholding, Uncertainty reduction (max-uncert), Misclassification (max-misclass)

However, we next examine the other proposed methods. Recall each algorithm selects  $L = 20$  items to serve to every respondent. But we examine hit rate for the top  $k$  items, for  $k = \{3, 10, 20, 40\}$ . Overall, a smaller  $k$  makes the problem more challenging to do well

**Table 2** Top k Hit Rate for Various Algorithms after the 260th, 500th Respondent with 120 Items (eric) Check the values to make sure they correspond with the Figures. I'm looking at TSe4 and TSthres columns and see discrepancies with the figures. (eric) And for tables, reorder columns: fixed\_express, greedy, greedy\_thres, TS, TSed, TS\_thres, TSed

| (a) After 260 respondents |               |        |             |              |       |              |            |         |              |
|---------------------------|---------------|--------|-------------|--------------|-------|--------------|------------|---------|--------------|
| k                         | fixed_express | greedy | greedythres | mismin       | TS    | TSe4         | TSregthres | TSthres | uncert       |
| 3                         | 0.650         | 0.853  | 0.860       | 0.883        | 0.863 | 0.873        | 0.850      | 0.893   | 0.840        |
| 10                        | 0.825         | 0.880  | 0.902       | 0.907        | 0.923 | 0.909        | 0.909      | 0.907   | 0.901        |
| 20                        | 0.824         | 0.784  | 0.863       | 0.885        | 0.825 | 0.820        | 0.869      | 0.879   | 0.886        |
| 40                        | 0.857         | NA     | 0.884       | 0.908        | NA    | NA           | 0.895      | 0.888   | 0.897        |
| (b) After 500 respondents |               |        |             |              |       |              |            |         |              |
| k                         | fixed_express | greedy | greedythres | mismin       | TS    | TSe4         | TSregthres | TSthres | uncert       |
| 3                         | 0.780         | 0.950  | 0.933       | 0.950        | 0.940 | <b>0.957</b> | 0.917      | 0.947   | 0.943        |
| 10                        | 0.877         | 0.925  | 0.928       | 0.939        | 0.933 | <b>0.952</b> | 0.940      | 0.947   | 0.945        |
| 20                        | 0.862         | 0.827  | 0.914       | <b>0.928</b> | 0.863 | 0.868        | 0.916      | 0.914   | 0.919        |
| 40                        | 0.887         | NA     | 0.926       | 0.937        | NA    | NA           | 0.929      | 0.927   | <b>0.938</b> |

earlier, since a hit rate for the top 3 out of 120 is a higher bar than the hit rate for 40 out of 120, especially in light of the distribution of true utility values (Figure 8).

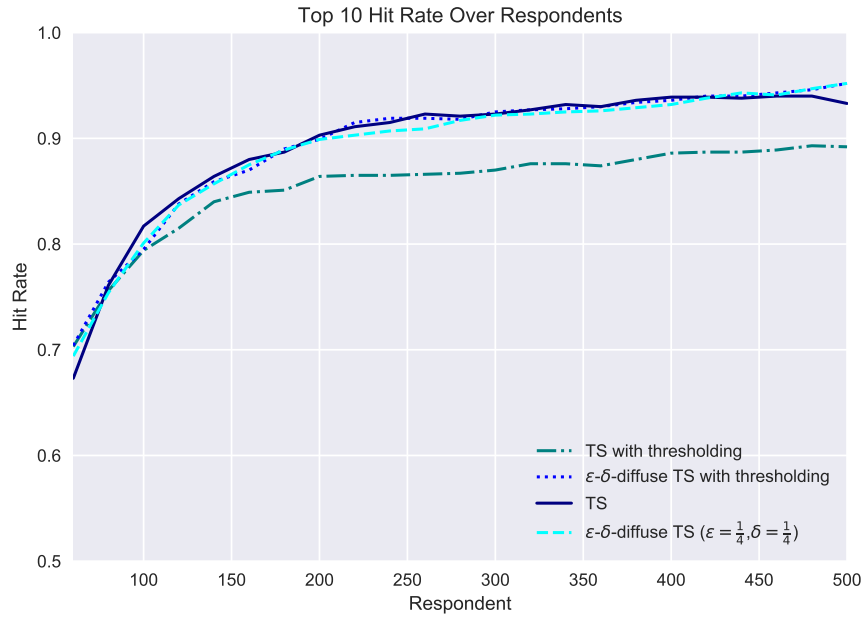
Figures 14 and 14 and Table 2 show the results for the  $K = 120$ -item dataset serving  $L = 20$  questions per respondent, where all methods were tested.

We find performance depends on the hit rate measure relative to the number of items selected ( $k$  vs  $L$ ). For  $k \geq L$  the better performers are Greatest Uncertainty with random perturbations (**max-uncert**) and Misclassification Minimization with random perturbations (**max-misclass**). Indeed these two methods also perform nearly as well as the other winners in the  $k < L$  condition. But the  $\epsilon$ -diffuse Thompson sampling with thresholding ( $\epsilon$ - $\delta$ -**TS-thres**) and without thresholding ( $\epsilon$ - $\delta$ -**diffuse TS**) perform best when  $k < L$ .

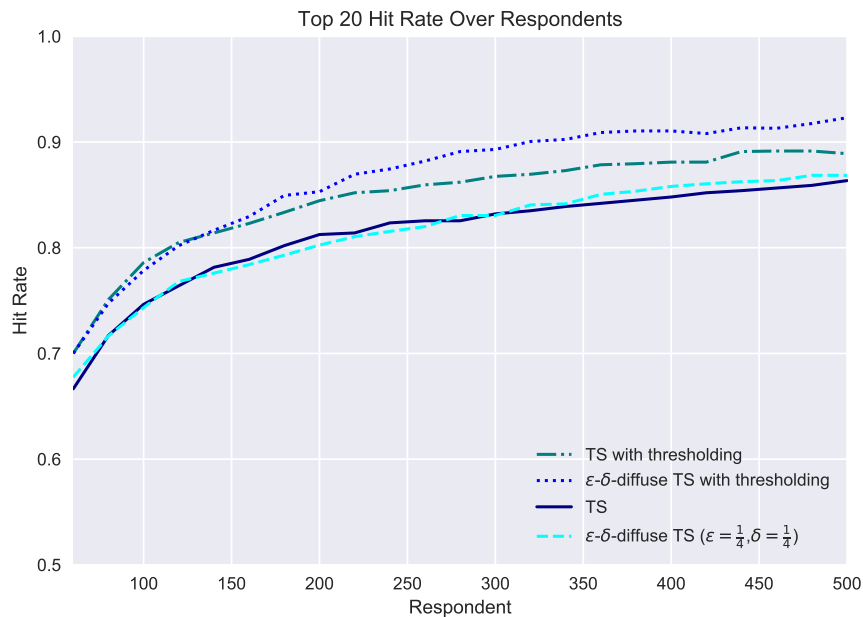
But the differences among the other algorithms are also telling. We find that Thompson Sampling (**TS**) alone has its limits. It is constructed to select the best items (and eventually, only the best item), so it can do quite well if we only care about identifying a small number of items (the cases  $k = 3, 10$ ). But is not suited for the active learning problem generally, especially when being evaluated on identifying a larger top set, for instance,

**Figure 11** Effects of Thresholds for Hit Rates for Top  $k = \{10, 20\}$  with 120 items. The cumulative hit rate obtained (y-axis) improves with the number of respondents interviewed (x-axis), but it does so at different rates for each algorithm. (eric) Please make y-axes the same across all  $k$  for the K120 L20 problem. Only show express,  $\epsilon$ -greedy, TS,  $\epsilon$ - $\delta$ -diffuse TS with usual values, and their thresholding counter parts,  $\epsilon$ -greedythres, TS-thres,  $\epsilon$ - $\delta$ -TS-thres. Why is TS better than TS-thres for  $k = 10$ ? Explain why in a good way TS-thres always does worse than  $\epsilon$ - $\delta$ -TS-thres.

(a) Top 10

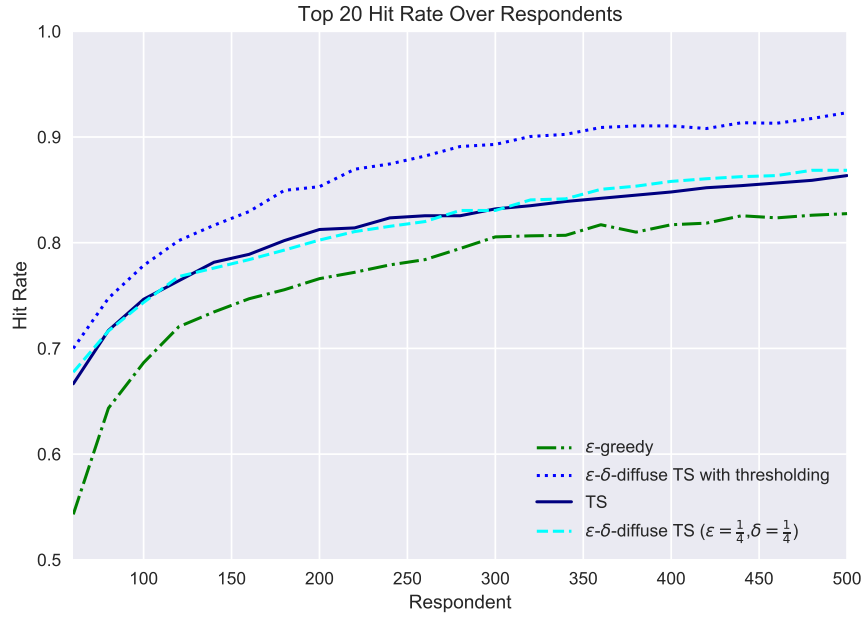


(b) Top 20

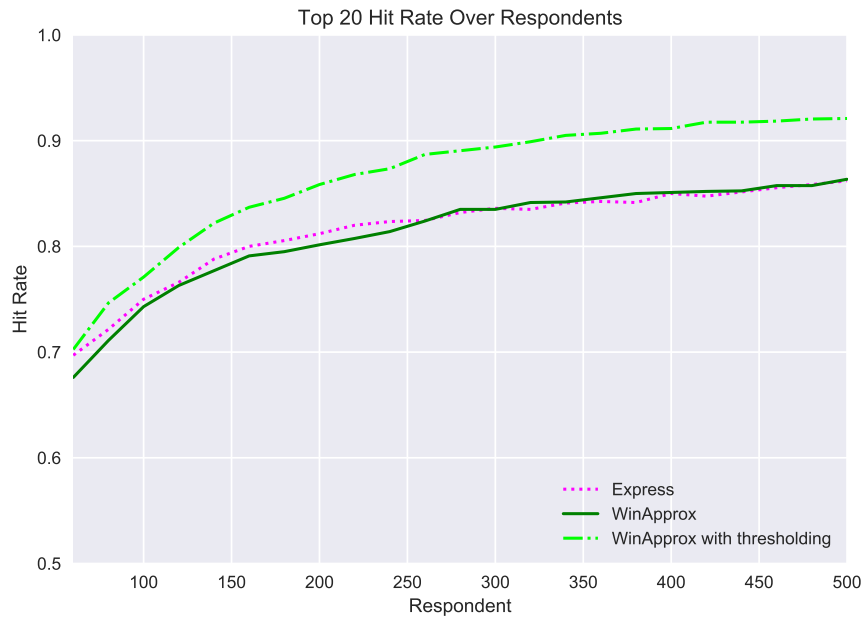


**Figure 12** Hit Rates for Top  $k = 20$  with 120 items, 20 items per person. The cumulative hit rate obtained (y-axis) improves with the number of respondents interviewed (x-axis), but it does so at different rates for each algorithm. (eric) express,  $\epsilon$ -greedy, WinApprox (for later), TS,  $\epsilon$ - $\delta$ -diffuse TS with usual values,  $\epsilon$ - $\delta$ -TS-thres .

(a) Top 20



(b) Top 20

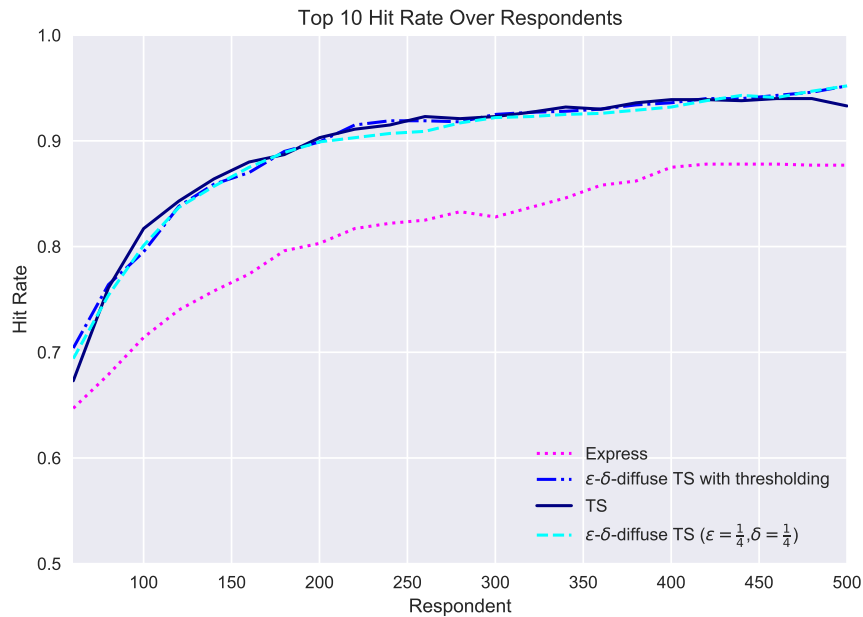


**Figure 13** Hit Rates for Top  $k = \{3, 10\}$  with 120 items. The cumulative hit rate obtained (y-axis) improves with the number of respondents interviewed (x-axis), but it does so at different rates for each algorithm. (eric) Please make y-axes the same across all k for the K120 L20 problem. Only show express, TS,  $\epsilon$ - $\delta$ -diffuse TS with usual values, TS-thres,  $\epsilon$ - $\delta$ -TS-thres. Why is TS better than TS-thres for  $k = 10$ ?

(a) Top 3



(b) Top 10

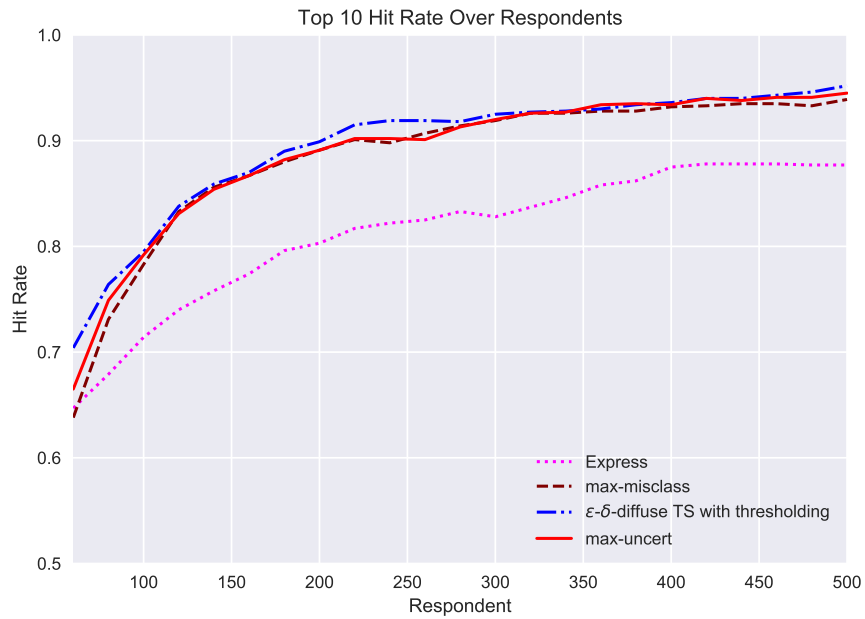


**Figure 14** Hit Rates for Top  $k = \{3, 10\}$  with 120 items. The cumulative hit rate obtained (y-axis) improves with the number of respondents interviewed (x-axis), but it does so at different rates for each algorithm. (eric) Please make y-axes the same across all k for the K120 L20 problem. Only show express,  $\epsilon$ - $\delta$ -TS-thres , max-misclass, max-uncert

(a) Top 3

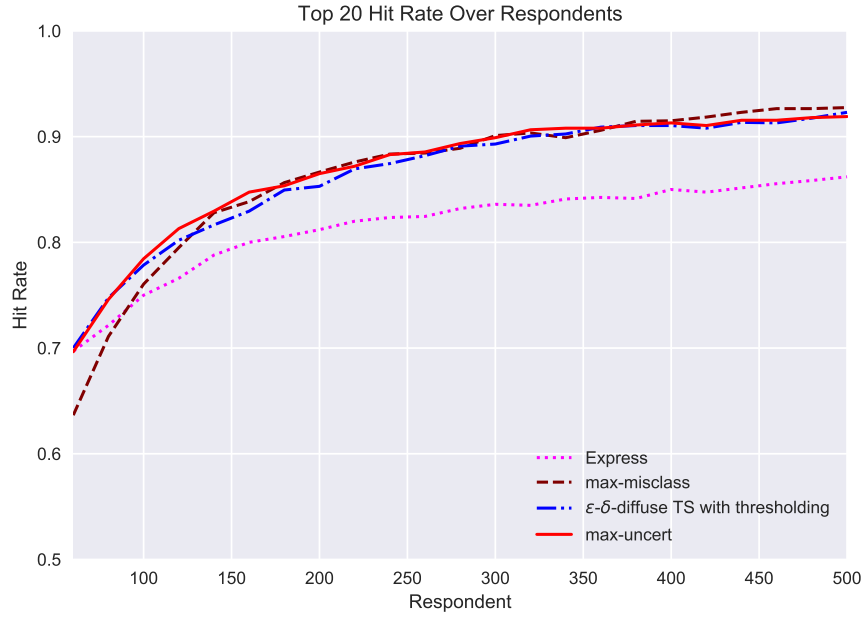


(b) Top 10

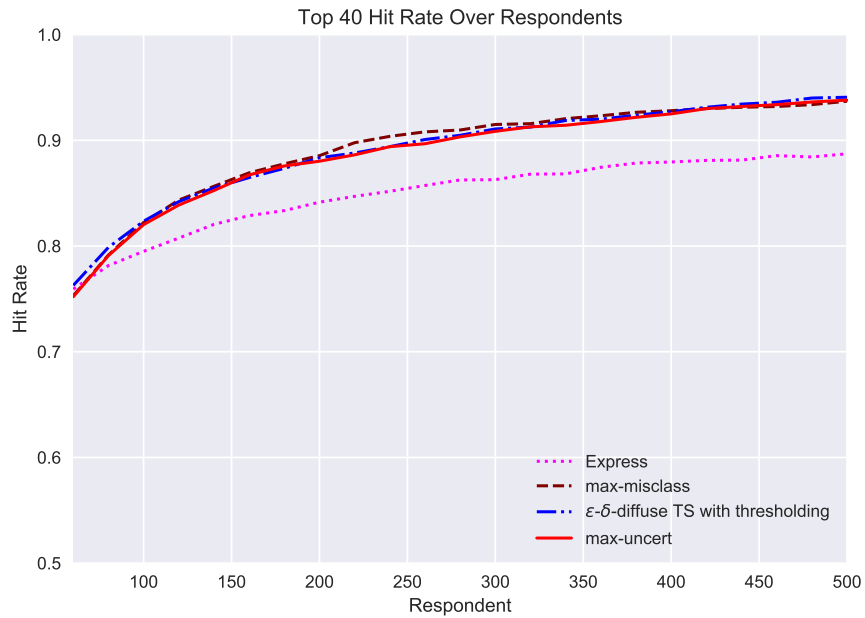


**Figure 15** Hit Rates for Top  $k = \{20, 40\}$  with 120 items. (eric) Please make y-axes the same across all k for the K120 L20 problem. Only show express,  $\epsilon$ - $\delta$ -TS-thres , max-misclass, max-uncert

(a) Top 20



(b) Top 40



| Resp | fixed_express | greedy | greedythres | mismin | TS    | TSe4  | TSregthres | TSthres | uncert |
|------|---------------|--------|-------------|--------|-------|-------|------------|---------|--------|
| 260  | 0.09          | 0.297  | 0.237       | 0.253  | 0.067 | 0.347 | 0.067      | 0.357   | 0.27   |
| 500  | 0.247         | 0.57   | 0.49        | 0.54   | 0.127 | 0.537 | 0.1        | 0.64    | 0.503  |

**Table 3 Top 3 Hit Rate for Various Algorithms at the 260th and 500th Respondent for the 120 item data set with Misinformed Start**

when we care about the hit rate for all items in the survey (the case  $k = L = 20$ ). While the  $\epsilon$ - $\delta$ -diffuse TS adds exploration to improve its performance for the other cases, that extra exploration does not direct resources toward the places that matter most in this active learning problem: the areas of high uncertainty at the decision boundary, not just near the top of the list and not scattered randomly.

Therefore, why do the threshold-based policies,  $\epsilon$ - $\delta$ -TS-thres , max-uncert and max-misclass, outperform those without thresholds? (eric) Add plots highlighting these differences under a few conditions

(eric) Add a Takeaway from each section beginning here. In conclusion, we find ...

## 6. Robustness tests

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

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 Adaptive 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 table 3, 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 Adaptive MaxDiff



| k  | fixed_express | greedy | greedythres | mismin | TS    | TSe4  | TSregthres | TStgres | uncert |
|----|---------------|--------|-------------|--------|-------|-------|------------|---------|--------|
| 3  | 0.307         | 0.593  | 0.477       | 0.543  | 0.597 | 0.640 | 0.537      | 0.647   | 0.580  |
| 10 | 0.462         | 0.612  | 0.540       | 0.682  | 0.630 | 0.662 | 0.576      | 0.674   | 0.694  |
| 20 | 0.550         | 0.599  | 0.593       | 0.722  | 0.606 | 0.658 | 0.590      | 0.714   | 0.719  |
| 40 | 0.617         | NA     | 0.616       | 0.705  | NA    | NA    | 0.638      | 0.702   | 0.721  |

**Table 4** Top k Hit Rate for Various Algorithms at the 260th Respondent for the 300 item data set

| k  | fixed_express | greedy | greedythres | mismin | TS    | TSe4  | TSregthres | TStgres | uncert |
|----|---------------|--------|-------------|--------|-------|-------|------------|---------|--------|
| 3  | 0.387         | 0.763  | 0.747       | 0.773  | 0.720 | 0.793 | 0.617      | 0.800   | 0.777  |
| 10 | 0.578         | 0.759  | 0.718       | 0.794  | 0.702 | 0.784 | 0.626      | 0.792   | 0.792  |
| 20 | 0.676         | 0.718  | 0.764       | 0.842  | 0.673 | 0.764 | 0.633      | 0.821   | 0.846  |
| 40 | 0.717         | NA     | 0.751       | 0.813  | NA    | NA    | 0.693      | 0.799   | 0.828  |

**Table 5** Top k Hit Rate for Various Algorithms at the 500th Respondent for the 300 item data set

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 our adaptive approaches performed well in simulations with well-behaved respondents, but fell apart under more realistic conditions.

We illustrate the robustness in Table 3. One critical point to note is that the standard Bandit MaxDiff TS approaches perform worst than Fixed Express MaxDiff under Misinformed Starts. The extra randomness Bandit MaxDiff  $\epsilon$ -diffuse TS approach as well as in the other methods is essential in allowing them 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.

## 6.2. Effect of increasing number of items

Would the benefits of Adaptive MaxDiff we observed with 120 items continue for 300 items? While we did not have a dataset of utilities from human respondents on 300 items,

| k  | fixed_express | greedy | greedythres | mismin | TS    | TSe4  | TSregthres | TSthres | uncert |
|----|---------------|--------|-------------|--------|-------|-------|------------|---------|--------|
| 3  | 0.897         | 0.973  | 0.970       | 0.940  | 0.973 | 0.973 | 0.973      | 0.960   | 0.943  |
| 10 | 0.869         | 0.897  | 0.912       | 0.907  | 0.926 | 0.905 | 0.930      | 0.911   | 0.916  |

**Table 6** Top k Hit Rate for Various Algorithms at the 260th Respondent for the 40 item data set

| k  | fixed_express | greedy | greedythres | mismin | TS    | TSe4  | TSregthres | TSthres | uncert |
|----|---------------|--------|-------------|--------|-------|-------|------------|---------|--------|
| 3  | 0.930         | 0.990  | 0.990       | 0.983  | 0.990 | 0.987 | 1.000      | 0.997   | 0.990  |
| 10 | 0.903         | 0.919  | 0.933       | 0.925  | 0.936 | 0.936 | 0.941      | 0.930   | 0.932  |

**Table 7** Top k Hit Rate for Various Algorithms at the 500th Respondent for the 40 item data set

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 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. Tables 4 and 5 shows the well-informed start results. The adaptive approaches show substantial gains over the Fixed Express MaxDiff approach on the top-3,10,20,40 hit rate criterion.

### 6.3. What about a smaller set of 40 Items?

Our adaptive models have 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. By reducing the number of items to 40, 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 Adaptive MaxDiff are still better than traditional MaxDiff.

### 6.4. Simulated data sets

We created two types of simulated preferences to see if these results would generalize to other data sets. The first, labeled the uncorrelated data set, each respondents preference utility for 100 items are sampled from a uniform distribution. So each respondent’s preferences are uncorrelated with every other respondent (the worst possible environment). We

| k  | fixed_express | greedy | greedythres | mismin | TS    | TSe4  | TSregthres | TSthres | uncert |
|----|---------------|--------|-------------|--------|-------|-------|------------|---------|--------|
| 10 | 0.212         | 0.232  | 0.220       | 0.260  | 0.236 | 0.228 | 0.244      | 0.220   | 0.244  |
| 20 | 0.337         | 0.343  | 0.349       | 0.319  | 0.335 | 0.340 | 0.348      | 0.337   | 0.349  |

**Table 8** Top k Hit Rate for Various Algorithms at the 260th Respondent for the uncorrelated data set

| k  | fixed_express | greedy | greedythres | mismin | TS    | TSe4  | TSregthres | TSthres | uncert |
|----|---------------|--------|-------------|--------|-------|-------|------------|---------|--------|
| 10 | 0.248         | 0.324  | 0.308       | 0.344  | 0.318 | 0.268 | 0.332      | 0.296   | 0.320  |
| 20 | 0.363         | 0.399  | 0.412       | 0.394  | 0.377 | 0.402 | 0.410      | 0.401   | 0.428  |

**Table 9** Top k Hit Rate for Various Algorithms at the 500th Respondent for the uncorrelated data set

| k  | fixed_express | greedy | greedythres | mismin | TS    | TSe4  | TSregthres | TSthres | uncert |
|----|---------------|--------|-------------|--------|-------|-------|------------|---------|--------|
| 10 | 0.772         | 0.878  | 0.884       | 0.888  | 0.870 | 0.892 | 0.882      | 0.892   | 0.878  |
| 20 | 0.874         | 0.862  | 0.922       | 0.929  | 0.875 | 0.904 | 0.925      | 0.935   | 0.933  |

**Table 10** Top k Hit Rate for Various Algorithms at the 260th Respondent for the correlated data set

| k  | fixed_express | greedy | greedythres | mismin | TS    | TSe4  | TSregthres | TSthres | uncert |
|----|---------------|--------|-------------|--------|-------|-------|------------|---------|--------|
| 10 | 0.846         | 0.922  | 0.924       | 0.932  | 0.930 | 0.938 | 0.916      | 0.940   | 0.934  |
| 20 | 0.902         | 0.895  | 0.954       | 0.957  | 0.896 | 0.927 | 0.947      | 0.961   | 0.963  |

**Table 11** Top k Hit Rate for Various Algorithms at the 500th Respondent for the correlated data set

take the top  $k$  highest mean preference scores across the respondents to be the top  $k$  true items. The results are in tables 8 and 9.

The second, labeled the correlated data set, we sample a  $\theta_i$  from a uniform distribution for  $i = 1, \dots, 100$ . Then each respondent preference utility is sampled from a normal distribution with mean  $\theta_i$  and variance 1. The rankings for each respondent are highly correlated with each other. Once again we take the top  $k$  highest mean preference scores across the respondents to be the top  $k$  true items. The results are in tables ?? and ??.

The take away is that our previous results generalize to other data sets.

## 7. Alternative Performance Measure: 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

| $S$         | 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 12** Percent true utility for Various  $S$  for Top 3 from the Procter & Gamble data

that differentiates the two is Percent True Utility (PTU).

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

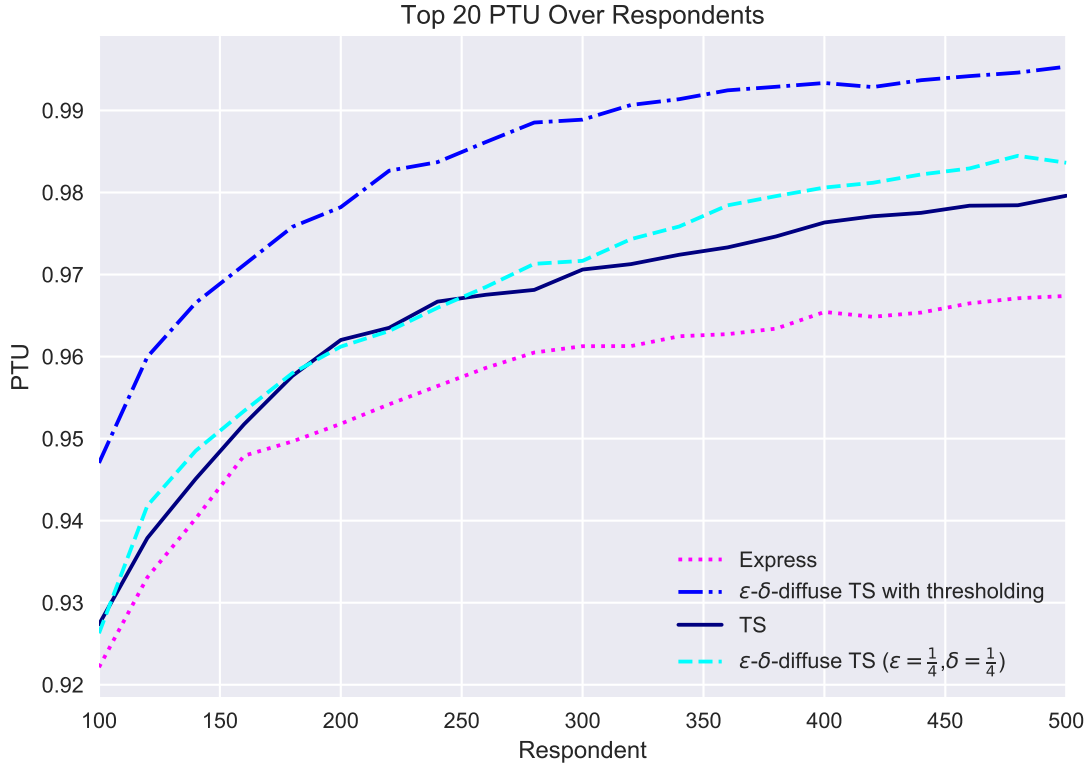
$$\mu_S = \sum_{x \in S} b(x) = \sum_{x \in S} e^{u(x)}$$

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

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

This can be seen as a generalization of Hit-Rate. It has the advantage of differentiating as in the case the section started with though one loses the ability to easily interpret the result. In Table 12 we show the PTU for various subsets of the data. Recall that for  $k = 20$  using the hit rate metric that Fixed Express, TS, and  $\epsilon$  diffuse TS were on par with each other. Using  $k = 20$  PTU, in figure 16 we see that  $\epsilon$  diffuse TS has a slight edge over TS which is better than Fixed Express. This means that both TS approaches put better ranked items in the top 20 than Fixed Express. In this wider consideration the methods are better than Fixed Express even though the top 20 hit rate was about the same.

Figure 16 Percent True Utility of the Top 20 with 120 items



## 8. Stopping Rules: Posterior Distribution Regret and Value Remaining

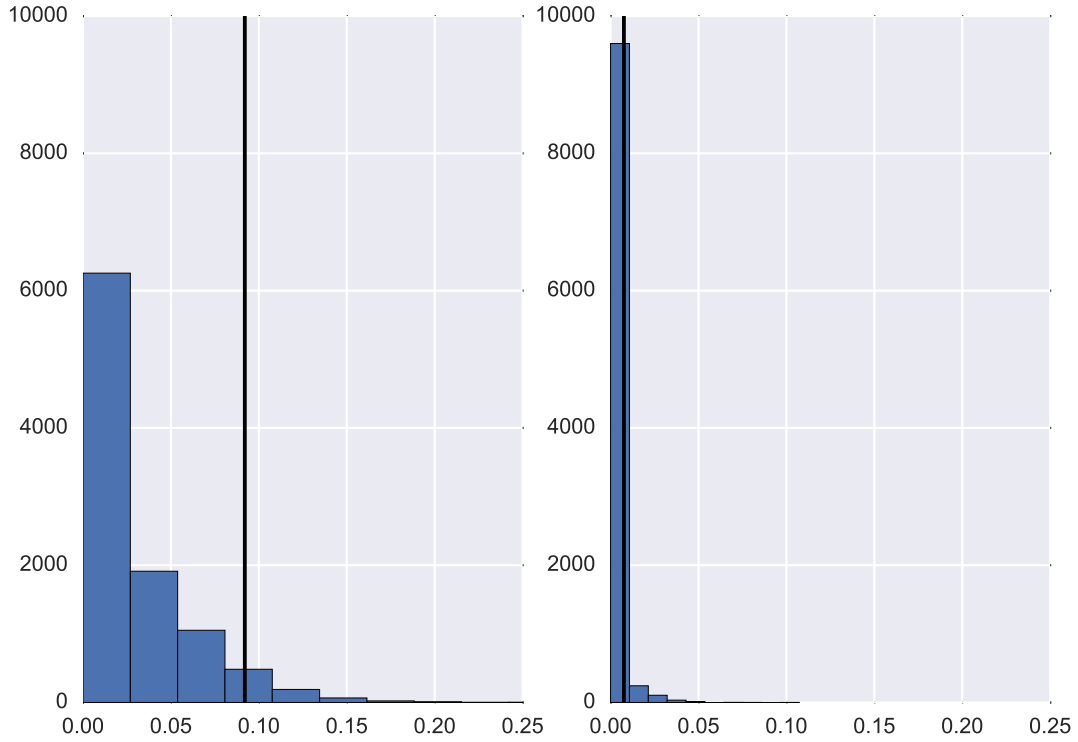
Almost all of the results show diminishing results as we give more surveys. Thus it becomes inefficient to keep giving surveys after a certain point. In practice we do not know this point *a priori* but we would still like to know when we should stop.

Adapted 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  $\mu_{S^*}$  is the largest value of the exponential weighted utility and  $\mu_S$  is the exponential weighted utility of the set that is most likely to be optimal, denoted  $S$ . This is constructed as follows, take  $n$  Bayes Bootstrap draws from the posterior. Let  $\mu_{S^*}^m$  be the max exponential weighted utility of draw  $m$  and  $\mu_S^m$  be the utility using the draw  $m$  using the set  $S$ . Let  $\Delta^m = \frac{\mu_{S^*}^m - \mu_S^m}{\mu_S^m}$ .

As an example see table 13 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.

|        | Current belief: rank order of items by utility |      |      |      |      |      |      |      |
|--------|--|------|------|------|------|------|------|------|
|        | 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 13** Draws of the exponential utility of the items after 100 iterations

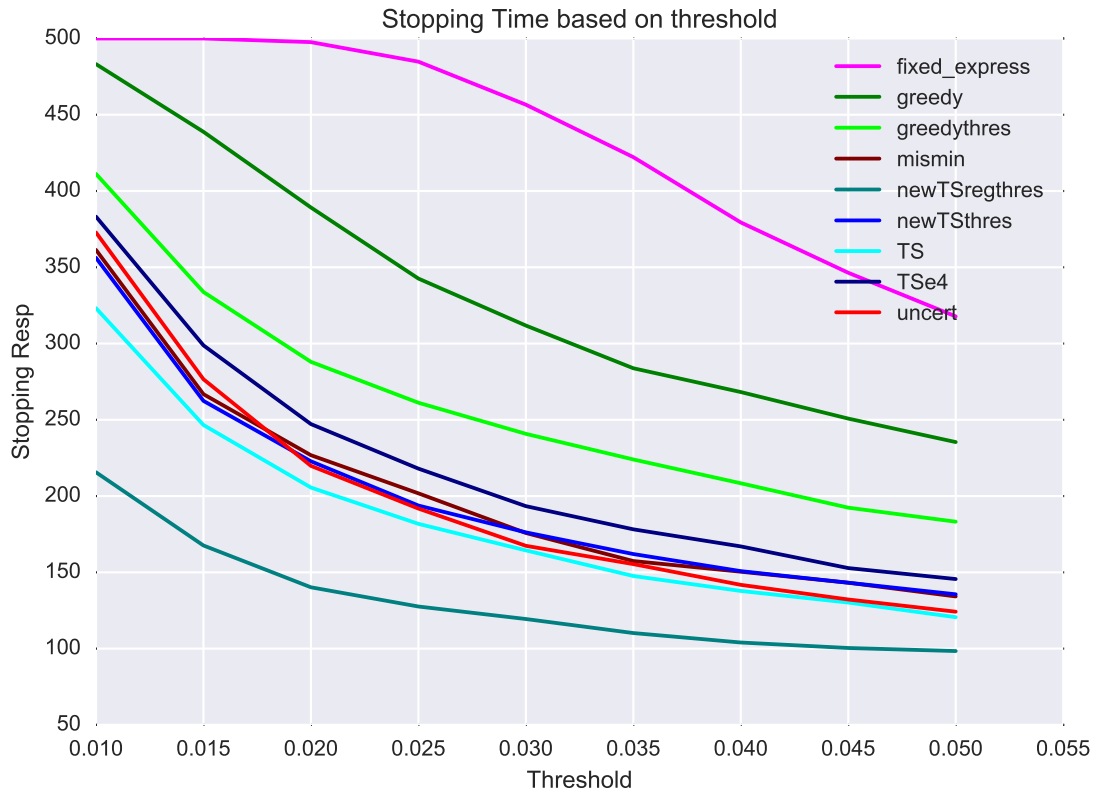


**Figure 17** Two histograms of  $\Delta$ . Left: After 100 iterations, the potential value remaining is .092. Right: After 220 iterations, the potential value remaining is .008

Then  $S$  is the set containing what is currently ranked as the first, second and third ranked item. Then  $\mu_S^1 = 4.02 + 3.50 + 5.08 = 12.6$  and  $\mu_{S^*}^1 = 5.08 + 4.41 + 4.16 = 13.65$  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  $\Delta$  after 100 iterations and 220 iterations is shown in figure 17.

The ‘potential value remaining’ (PVR) is the 95 quantile of the distribution  $\Delta$  see figure 17. After 100 iterations the PVR was 0.092. Scott’s way to interpret this number is “we

**Figure 18** Average stopping time with defferent thresholds



|        | fixed_express | greedy | greedythres | mismin | TS    | TSe4  | TSregthres | TSthres | uncert |
|--------|---------------|--------|-------------|--------|-------|-------|------------|---------|--------|
| Avg ST | 318           | 235    | 183         | 134    | 121   | 146   | 98         | 136     | 124    |
| Avg hr | 0.854         | 0.88   | 0.852       | 0.848  | 0.846 | 0.866 | 0.795      | 0.857   | 0.837  |

**Table 14** Average Stopping Time and Average Hit Rate With a threshold of .02 PVR for top 10 items

|        | fixed_express | greedy | greedythres | mismin | TS    | TSe4  | TSregthres | TSthres | uncert |
|--------|---------------|--------|-------------|--------|-------|-------|------------|---------|--------|
| Avg ST | 498           | 389    | 288         | 227    | 206   | 247   | 140        | 223     | 220    |
| Avg hr | 0.893         | 0.918  | 0.902       | 0.911  | 0.907 | 0.912 | 0.822      | 0.911   | 0.909  |

**Table 15** Average Stopping Time and Average Hit Rate With a threshold of .05 PVR for top 10 items

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. See figure 18 for the average stopping time for different algorithms based on threshold for

finding the top 10 items. Tables 14 and 15 show the average stopping time and average hit rate at those stopping time for threshold values 0.02 and 0.05. The trade-off is accuracy for how many respondents you survey. If you are using regret as a stopping rule, keep in mind that in general higher  $k$  gives smaller PVR values.

## 9. Other Sampling Schemes and Methods

### 9.1. Asking for Bests instead of Best-Worst

Because a key assumption for using an adaptive 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.

### 9.2. What about Double Adaptivity?

In Orme (2006), one of the authors presented a paper on Adaptive MaxDiff that featured within-respondent adaptation 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 top 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.

### 9.3. Drawing from the Asymptotic Distribution

As an alternative to Bayesian bootstrapping to get draws from the posterior, one can sample from asymptotic distribution via MLE implied by the estimated mean and standard errors, sampling  $u \sim N(\theta, H^{-1})$ , where  $\theta$  is the minimizer of the negative loglikelihood and  $H$  is the Hessian of the negative loglikelihood evaluated at  $\theta$ . The sampling methods performed about the same using draws from either Bayesian bootstrapping or the asymptotic distribution.

Finally, yet another faster allows for online updates for streaming data, without model estimation. A benefit of MaxDiff tasks is the number of pairwise comparison is implies with one best-worst pair. One, albeit limited way to use this information, would be to act as if the respondent actually did state all of those comparisons. Then for each pair, we can assign a 'win' the more preferred item and a 'loss' to the others. By merely tallying the number of wins out of comparisons for each item, we have its own 'win percentage' in its contests against other items that it has faced so far. This is a noisy measure of proxy for its underlying utility. Appreciating this is an average as  $\hat{p}_{win}$ , we can describe its standard error,  $se_{win}$ , to be used for Thompson Sampling as follows:

$$\hat{p}_{k,win} = n_{k,wins} / n_{trials} \tag{1}$$

$$se_{k,win} = \sqrt{\frac{\hat{p}_{k,win}(1 - \hat{p}_{k,win})}{n_{trials}}} \tag{2}$$

$$\tag{3}$$

The mean and variance can be used for sampling from the distribution of ‘winning percentages’. This can be done using **TS** or  $\epsilon$ - $\delta$ -diffuse **TS**. For **TS**, for example, after sampling one  $p_{k,win}^{ts}$  for each of the  $k = 1, \dots, K$  items independently, then simply rank order and identify the top set of those values. For  $\epsilon$ - $\delta$ -diffuse **TS**, do this for two different Normal distributions, one with  $se_{k,win}$  and one with inflated standard error,  $\delta se_{k,win}$ .

#### 9.4. What about Sparse MaxDiff vs. Express MaxDiff?

In Wirth and Wolfrath (2012), the authors compared non-adaptive Sparse MaxDiff and Express MaxDiff at the Sawtooth Software Conference.

**Fixed Sparse MaxDiff:** we showed each item to each respondent an equal number of times (if possible). With 120 items, 12 sets, and 5 items per set each item appeared on average  $\frac{12 \cdot 5}{120} = 0.5$  times per respondent.

We compared the results using our simulation and found a modest edge in performance for Express MaxDiff (Sparse ending at 85.6% for top 10 hit rate and Express ending at 87.7%).

## 10. 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 MaxDiff approaches can be 3x more efficient than standard Express MaxDiff designs. You are potentially wasting 66 cents of each dollar spent on data collection by not using adaptive MaxDiff.

Adaptive MaxDiff leverages information from prior respondents to show more effective trade-offs to later respondents (tending to oversample the stars). Greatest Uncertainty with random perturbations, Misclassification Minimization with random perturbations,  $\epsilon$ -diffuse Thompson sampling with thresholding, and  $\epsilon$ -greedy with thresholding under any choice of  $k$ . Additionally  $\epsilon$ -diffuse Thompson Sampling works well when  $k < L$  and  $\epsilon$ -greedy works well when  $k \ll L$ . One author prefers Misclassification Minimization with random perturbations and  $\epsilon$ -diffuse Thompson sampling with thresholding in all cases.

Even in the face of diabolically imposed misinformed starts (horribly unrepresentative first responders), these adaptive MaxDiff approaches are robust and self-correcting.

Although our simulations involve 120-item and 300-item tests, we expect that even greater efficiency gains (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 adaptive approaches. 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 Adaptive MaxDiff as a commercial tool. Sawtooth Software may perhaps one day soon offer Adaptive 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.
- Gabillon V, Ghavamzadeh M, Lazaric A (2012) Best arm identification: A unified approach to fixed budget and fixed confidence. *Advances in Neural Information Processing Systems*, 3212–3220.
- 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.
- Kalyanakrishnan S, Tewari A, Auer P, Stone P (2012) Pac subset selection in stochastic multi-armed bandits. *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 655–662.
- Kaufmann E, Cappé O, Garivier A (2016) On the complexity of best arm identification in multi-armed bandit models. *Journal of Machine Learning Research* 17(1):1–42.
- Kaufmann E, Kalyanakrishnan S (2013) Information complexity in bandit subset selection. *Conference on Learning Theory*, 228–251.
- 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.