Nonparametric Discrete Choice Experiments with Machine Learning Guided Adaptive Design

Mingzhang Yin MINGZHANG.YIN@WARRINGTON.UFL.EDU
Department of Marketing, University of Florida
Gainesville, FL 32611, USA

Ruijiang Gao RUIJIANG@UTEXAS.EDU
Department of Information, Risk and Operations Management, University of Texas at Austin
Austin, TX 78712, USA

Weiran Lin WEIRAN.LIN@UFL.EDU
Department of Marketing, University of Florida
Gainesville, FL 32611, USA

Steven M. Shugan STEVEN.SHUGAN@WARRINGTON.UFL.EDU
Department of Marketing, University of Florida
Gainesville, FL 32611, USA

Abstract

Designing products to meet consumers' preferences is essential for a business's success. We propose Gradient-based Survey (GBS), a discrete choice experiment for multiattribute product design. The experiment elicits consumer preferences through a sequence of paired comparisons for partial profiles. GBS adaptively constructs paired comparison questions based on the respondents' previous choices. Unlike the traditional random utility maximization paradigm, GBS is robust to model misspecification by not requiring a parametric utility model. Cross-pollinating the machine learning and experiment design, GBS is scalable to products with hundreds of attributes and can design personalized products for heterogeneous consumers. We demonstrate the advantage of GBS in accuracy and sample efficiency compared to the existing parametric and nonparametric methods in simulations.

Keywords: Experiment Design, Discrete Optimization, Conjoint Analysis

1. Introduction

Identifying an optimal product based on consumers' preferences is essential for the success of a business. Such a problem is prevalent in companies where the product consists of multiple attributes such as health insurance, cell phone plans, movies, pizzas, automobiles, logos, and email advertisements (Balakrishnan et al., 2004; Bertsimas and Mišić, 2017; Ellickson et al., 2022; Netzer and Srinivasan, 2011). In this paper, we focus on products with discrete attributes represented as multivariate binary variables, as is the case in A/B testing [1].

Selecting appropriate products from a choice set is complicated for decision-makers. First, the desired product should meet the unobserved consumer preferences. The latent preferences are often revealed by survey experiments known as conjoint analysis (Green and Rao, 1971; Luce and Tukey, 1964). Since the milestone work of Louviere and Woodworth (1983), the choice-based conjoint (CBC) analysis became one of the most widely used meth-

1. A discrete attribute with more than two levels can be coded as multiple binary variables.

ods to quantify multiattribute preference (Hein et al., 2020). A prevalent assumption is that the preference for an attribute is quantified by a part-worth score, and the total utility of a product profile is the sum of part-worths (Green and Srinivasan, 1978). This parametric assumption, however, may oversimplify how respondents encode and evaluate products (Allenby et al., 2005). Second, the choice set for product design grows exponentially with the number of attributes, making the problem NP-hard (Kohli and Krishnamurti, 1989). The problem is more evident with the development of technology when more and more components are integrated into a single product. For example, the design of a smartphone might need to consider hundreds of attributes from a digital camera, screen display, connectivity modules, software applications, and a range of sensors. The high-dimensional attributes pose a scalability challenge to the extant product design methods. Lastly, consumer preference is heterogeneous, and it is desired to provide a customized product aligned with individual tastes. Nowadays, an increasing number of products are presented as digital content, making personalized product design feasible.

Our proposed GBS is robust to model misspecification of the utility function, scalable to high-dimensional attributes, and applicable to single or personalized product design. The idea of GBS is to combine gradient-based machine learning with discrete choice experiments (DCEs). We model the product attributes as random variables following Bernoulli distributions. The inference of the optimal product is by computing the gradient of the objective, such as the market share, with respect to the distribution parameters. However, computing the gradient is challenging because of the unknown functional form of respondent choice and the infeasibility in computing a gradient with non-continuous variables. To address these challenges, we adopt the score function method to compute the gradient using data from DCEs. Building on the recent development of discrete optimization in machine learning, we develop new variance reduction tools for the score function gradient. The unbiased and low-variance gradient is then used to generate the next survey questionnaire and update the optimal product's distribution parameters by stochastic gradient descent (SGD).

We demonstrate that GBS scales to hundreds of attributes efficiently and can infer the optimal products accurately in our experiments compared with parametric and nonparametric baselines. Over a variety of utility functions, GBS is more robust in model specification than parametric methods and is more sample-efficient than neural networks. Finally, we apply GBS to learn an individual policy with combinatorial actions via experiments. We discuss the connection of GBS and related work in Appendix A.

2. Problem Set Up

Denote $Y_i(Z, Z_0)$ as the user i's choice of product given the focal product Z and a baseline product Z_0. The products are represented by K binary features $Z, Z_0 \in \{0,1\}^K$. The baseline product Z_0 can be a competitor's product, the exisitng product in the market, or an empty set. The potential outcome $Y_i(Z, Z_0)$ is distributed according to the unknown data generating process. $Y_i(Z_1, Z_0) = 1$ if user i chooses Z_1 and $Y_i(Z_1, Z_0) = 0$ if Z_0 is chosen.

We first consider how to identify a single optimal product. The objective is

$$\max_{Z} V(Z) = \mathbb{E}[Y_i(Z, Z_0)], \tag{1}$$

which represents the potential market share of the proposed product Z in the presence of the baseline product. The difficulties in solving Eq. (1) are that the number of possible products grows exponentially with feature numbers, and the functional form of $Y_i(Z, Z_0)$ is unknown and might be complicated.

3. Gradient-based Survey Design

We adopt the Random Utility Maximisation (RUM) framework to model the choice behavior (McFadden, 1974). Each product Z_j is associated with a utility $U_i(Z_j)$ for individual i. The alternative Z_1 is chosen from the pair (Z_1, Z_0) if and only if $U_i(Z_1) > U_i(Z_0)$. In RUM, the utility is decomposed as $U_i(Z_j) = V_i(Z_j) + \epsilon_{ij}, j = 0, 1$ where $V_i(Z_j)$ is often called the representative utility (Train, 2009). We assume the random ϵ_{ij} independently follows type I extreme value (Gumbel) distribution. Accordingly, the probability of choosing item Z is $p(Y_i(Z, Z_0) = 1) = \exp(V_i(Z))/(\exp(V_i(Z)) + \exp(V_i(Z_0)))$. Notice that the product Z that maximizes the choice probability $p(Y_i(Z, Z_0) = 1)$ is irrelevant to what the baseline product Z_0 is. We do not make a parametric assumption for the representative utility $V_i(Z)$.

Instead of optimizing the discrete features Z directly, we consider Z as random variables following distribution $p(Z; \pi) = \prod_{k=1}^{K} \mathrm{Bern}(z_k; \pi_k),\ \pi_k \in [0, 1]$. We then transform the problem in Eq. (1) to an equivalent problem with the same optimal solution as

$$\max_{\pi} V(\pi) = \mathbb{E}_{Z\sim p(Z;\pi)}\mathbb{E}[Y_i(Z, Z_0) \,|\, Z]. \tag{2}$$

The equivalence of probabilistic reformulation is shown in Yin et al. (2020) Theorem 1. To facilitate the optimization in an unconstrained space, we parameterize the probability by the sigmoid function $Z \sim \prod_{k=1}^{K} \mathrm{Bern}(z; \pi_k = \sigma(\phi_k)), \phi \in \mathbb{R}^K,\ \sigma(x) = 1/(1 + e^{-x})$, and optimize $V(\phi) := V(\pi = \sigma(\phi))$ with the logits ϕ.

The gradient of Eq. (2) can be computed by the score function estimator (a.k.a. REINFORCE) as $\nabla_\phi V(\phi) = \mathbb{E}_{Z\sim p(Z;\phi)}[\nabla_\theta \log p(Z; \phi)\mathbb{E}[Y_i(Z, Z_0) \,|\, Z]]$. However, score function gradients often suffer from high variance, and many works have been devoted to reducing the variance. With a direct application of the antithetic sampling and control variates (Yin and Zhou, 2019) to the product design problem, we have the following Lemma.

Lemma 1 *An unbiased gradient of the objective $V(\phi)$ is*

$$\nabla_\phi V(\phi) = \mathbb{E}_{u\sim\prod_{k=1}^{K} \mathit{Unif}(0,1)}\Big[\mathbb{E}\big[(Y_i(Z_1(u), Z_0) - Y_i(Z_2(u), Z_0))(u - \frac{1}{2}) \,|\, u\big]\Big], \tag{3}$$

where $Z_1(u) = \mathbf{1}[u > \sigma(-\phi)], Z_2(u) = \mathbf{1}[u < \sigma(\phi)]$.

For completeness, the proof is in the Appendix. The gradient in Eq. (3) is guaranteed to reduce the variance of score function gradient for non-negative objective (Yin and Zhou, 2019). A Monte-Carlo estimate of the gradient in Eq. (3) can be computed by asking respondent i to choose between products Z_1 and Z_0, between Z_2 and Z_0, and take the difference of the choices. However, such question design might be sensitive to the selection of Z_0. A strong or weak Z_0 may make the choices $Y_i(Z_1, Z_0)$ and $Y_i(Z_2, Z_0)$ the same frequently, resuting in a zero gradient. Moreover, a respondent's preferences across two choices may not be consistent and comparable.

To mitigate these problems, we marginalize out the zero gradients using Lemma 2.

Lemma 2 $Y(Z_1, Z_0) - Y(Z_2, Z_0) \,|\, Y(Z_1, Z_0) \neq Y(Z_2, Z_0) \overset{d}{=} (2Y(Z_1, Z_2) - 1)$

With Lemmas 1 and 2, we have the following result.

Lemma 3 *$\tilde{g} = (2Y_i(Z_1(u), Z_2(u)) - 1)(u - \frac{1}{2})p(\mathcal{A}_i)$ satisfies $\mathbb{E}[\tilde{g}] = \nabla_\phi V(\phi)$, where $Z_1(u)$, $Z_2(u)$ are defined as in Lemma 1,$u \sim \prod_{k=1}^K$ Unif$(0, 1)$, event $\mathcal{A}_i = \{Y_i(Z_1, Z_0) \neq Y_i(Z_2, Z_0)\}$.*

Computing $\tilde{g}$ takes a choice $Y_i(Z_1(u), Z_2(u))$ from a random respondent i between two partial profiles $Z_1(u)$ and $Z_2(u)$. The baseline product Z_0 is no longer needed. The partial profile comparison alleviates cognitive burden and elicits respondents' preferences more accurately than a choice from a large action space (a demonstration is in Appendix Fig. 4). $Z_1(u)$ and $Z_2(u)$ differ in feature k with probability $1 - |2\pi_k - 1|$, which increases monotonically from 0 to 1 with the variance $\pi_k(1 - \pi_k)$. It is aligned with the intuition that in order to maximize the information gain from each question, features with high uncertainty should have a high chance of being asked. Though the quantity $p(\mathcal{A}_i)$ is unknown, it is a constant shared by all the elements of the stochastic gradient $\tilde{g}$. Therefore, the constant can be absorbed in the stepsize of the SGD and does not affect the convergence. We propose the gradient estimate for GBS with $Z_1(u), Z_2(u)$ defined in Lemma 1 as

$$g_{\text{GBS}} = (2Y_i(Z_1(u), Z_2(u)) - 1)(u - \frac{1}{2}), \quad u \sim \prod_{k=1}^{K} \text{Unif}(0, 1). \tag{4}$$

The steps of GBS are summarized in Alg. 1 in Appendix E.

The data collection of GBS shares the same form with the paired CBC analysis, which has been widely applied to understanding individual preferences in marketing, politics, and other computational social science for decades (Egami and Imai, 2019; Goplerud et al., 2022; Hainmueller and Hopkins, 2015; Luce and Tukey, 1964; Toubia et al., 2003). GBS is thus compatible with many existing survey systems. The paired choice design used in practice requires a single choice for each question and rules out the situation of choosing both or none, which echos the step of zero gradients marginalization in Eq. (3). In a nutshell, GBS uses the gradient information to automatically and adaptively design experiments, and on the other hand, uses the data from the experiments to estimate a low-variance stochastic gradient for optimal product identification.

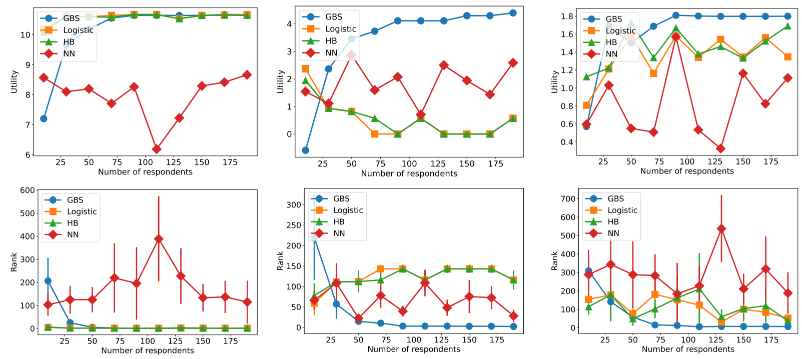
4. Individualized Policy Learning

A single product is often not optimal for all users. For example, a personalized email designed based on individual shopping history may improve the consumers' open rate. We consider learning an individualized policy that assigns a customized product to each user.

Suppose the covariates X_i of individual i are observed. The optimization objective is

$$\max_\theta V(\theta) = \mathbb{E}_{X_i \sim p(X)} \mathbb{E}_{Z_i \sim p(Z; \pi_\theta(X_i))} \mathbb{E}[Y_i(Z_i, Z_0)], \tag{5}$$

where $p(Z; \pi_\theta(X_i)) = \prod_{k=1}^K \text{Bern}(z_k; \pi_\theta(X_i)_k)$. The policy $\pi_\theta(X_i) = \sigma(\phi_i)$ and the logits $\phi_i = g(X_i; \theta) \in \mathbb{R}^K$ are the output of an amortized neural network parameterized by θ with input X_i. Applying the results in § 3, an unbiased estimate of the gradient w.r.t. the logits



(a) Type 1 (b) Type 2 (c) Type 3

Figure 1: The utility (Top row, higher is better) and the ranking (Bottom row, lower is better) for the designed products with $K = 10$ features. Lines are averaged over 10 trials.

ϕ_i is $c_i(2Y_i(Z_1(u), Z_2(u)) - 1)(u - \frac{1}{2})$ where $u \sim \prod_{k=1}^{K} \text{Unif}(0, 1)$ and c_i is a scalar to be absorbed in the stepsize. Using the chain rule, the GBS gradient of policy parameter θ is

$$g_{\text{GBS}}(\theta) = (2Y_i(Z_1(u), Z_2(u)) - 1)(u - \frac{1}{2})^\top \frac{\partial g(X_i; \theta)}{\partial \theta}, \quad u \sim \prod_{k=1}^{K} \text{Unif}(0, 1) \tag{6}$$

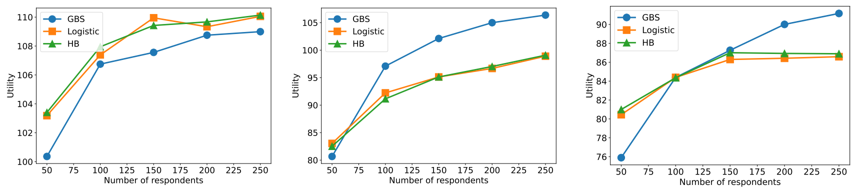
with $Z_1(u), Z_2(u)$ defined in Lemma 1. See Alg. 2 in Appendix E for the algorithm.

5. Empirical Study

We compare GBS with baseline models using simulated data similar to Toubia et al. (2007). The product Z is represented as K binary features. First, we study the problem of identifying a single optimal product. For each independent trial, the population-level marginal preferences for the features are generated by $\mu \sim \mathcal{N}(a, I)$, $a = (1, \cdots, 1)$. The individual preferences (partworths) are generated by $W_i \sim \mathcal{N}(\mu, I)$ for each individual i.

We consider three types of representative utilities 1) linear utility; 2)pairwise interactions; and 3) a pre-trained neural network, which includes higher-order interactions of the product features. For baselines, we consider 1) Logistic model; 2) hierarchical Bayes (HB) model which is widely used in conjoint analysis (Allenby et al., 2005); and 3) a neural network (NN)-based utility. For evaluation metrics, we consider the average utility for the selected product. It compares the relative performance of different methods. When computationally feasible, we also rank all the possible products according to their average utility on the population from high to low. The ranking allows the evaluation of absolute performance compared to the global optimum. See Appendix C for more details.

Fig. 1 shows the test utility and ranking of the estimated optimal product across a different number of respondents for a product with $K = 10$ features. When the utility function is correctly specified (Type 1), Logistic and HB reach the highest utility with a small number of respondents. However, when the utility model is misspecified (Type 2, 3), the performance of Logistic and HB with linearity assumption significantly deteriorates. For Type 2 utility, the ranking drops with an increasing number of respondents when the driving factor for the estimates shifts from variance to bias. NN assumes a flexible nonparametric utility



(a) Type 1 (b) Type 2 (c) Type 3

Figure 2: The utility (higher is better) with $K = 100$ features.

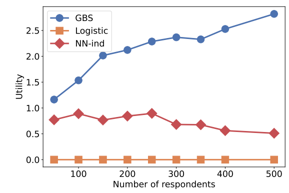


Figure 3: Personalized Product.

function, but when the sample size is small, its performance is dominated by the variances in the data. Fig. 5 in the Appendix E shows the results of NN with additional respondents. For the utility function without interactions, it needs 500 respondents to select the optimal product, and for relatively complex Type 2 and 3 utility, the number of respondents needed is 4000 and 2000, respectively. The weak data efficiency increases the experimental cost and may be infeasible. Moreover, finding the optimal product with a trained NN utility function needs to explore all possible products as inputs, which becomes challenging when the feature dimension is high. In comparison, GBS identifies the optimal products across all utility types with less than 100 respondents. For the linear utility, when the respondent number is small, the correctly specified models outperform GBS, but the performance gap diminishes quickly when the respondent number increases to around 70. For nonlinear utilities, GBS outperforms the baseline models and achieves the global optimum.

Fig. 2 shows the results for the single product design with $K = 100$ features. Finding the optimal product with a trained NN needs to compare the predicted utility of 2^{100} items, which is infeasible to compute, so we drop it from the baselines. Similarly, we drop the ranking metric that needs to evaluate all product combinations. The performance pattern is similar to $K = 10$. GBS can identify the optimal product from around 10^{30} choices in less than half a minute. It is flexible with the underlying utility function and is efficient with data size, which makes it practically applicable.

Next, we study the personalized product design with choice experiments. The data is generated the same as before, except the individual preferences follow a mixture distribution which reflects the preference heterogeneity in the population. See Appendix D for details. Fig. 3 shows the selected products' utility. Since Logistic estimates a single optimal product, the estimate is a product with zero utility due to the symmetry of the population. NN-ind adapts to individual heterogeneity and has utility higher than Logistic. However, it is subject to model misspecification due to the last linear layer and may have low data efficiency as in the case of the single optimal product. GBS reaches the highest utility, and the performance improves with more respondents joining the experiments.

6. Discussions

This paper bridges the domains of gradient-based machine learning and discrete choice experiments. GBS is flexible with the underlying form of choice utility, is data-efficient with adaptive design, is scalable to high-dimensional features, and can be applied to uniform or personalized product designs. We include more discussions in Appendix F.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Greg M Allenby, Peter E Rossi, and Robert E McCulloch. Hierarchical bayes models: A practitioners guide. *SSRN Electronic Journal*, 2005.

Susan Athey and Stefan Wager. Policy learning with observational data. February 2017.

PV Balakrishnan and Varghese S Jacob. Genetic algorithms for product design. *Management Science*, 42(8):1105–1117, 1996.

PV Balakrishnan, Rakesh Gupta, and Varghese S Jacob. Development of hybrid genetic algorithms for product line designs. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):468–483, 2004.

Alexandre Belloni, Robert Freund, Matthew Selove, and Duncan Simester. Optimizing product line designs: Efficient methods and comparisons. *Management Science*, 54(9): 1544–1552, 2008.

Dimitris Bertsimas and Velibor V Mišić. Robust product line design. *Operations Research*, 65(1):19–37, 2017.

Alex Burnap, John R Hauser, and Artem Timoshenko. Product aesthetic design: A machine learning augmentation. *Marketing Science*, 2023.

Jeffrey D Camm, James J Cochran, David J Curry, and Sriram Kannan. Conjoint optimization: An exact branch-and-bound algorithm for the share-of-choice problem. *Management Science*, 52(3):435–447, 2006.

Arthur Delarue, Ross Anderson, and Christian Tjandraatmadja. Reinforcement learning with combinatorial actions: An application to vehicle routing. *Advances in Neural Information Processing Systems*, 33:609–620, 2020.

Zhe Dong, Andriy Mnih, and George Tucker. DisARM: an antithetic gradient estimator for binary latent variables, June 2020.

Naoki Egami and Kosuke Imai. Causal interaction in factorial experiments: Application to conjoint analysis. *Journal of the American Statistical Association*, 114(526):529–540, April 2019.

Paul B Ellickson, Wreetabrata Kar, and James C Reeder. Estimating marketing component effects: Double machine learning from targeted digital promotions. *Marketing Science*, September 2022.

Max Goplerud, Kosuke Imai, and Nicole E Pashley. Estimating heterogeneous causal effects of High-Dimensional treatments: Application to conjoint analysis. *arXiv preprint arXiv:2201.01357*, January 2022.

Paul E Green and Vithala R Rao. Conjoint measurement-for quantifying judgmental data. *Journal of Marketing research*, 8(3):355–363, 1971.

Paul E Green and Venkatachary Srinivasan. Conjoint analysis in consumer research: issues and outlook. *Journal of consumer research*, 5(2):103–123, 1978.

Paul E Green, Abba M Krieger, and Manoj K Agarwal. Adaptive conjoint analysis: Some caveats and suggestions. *Journal of Marketing Research*, 28(2):215–222, 1991.

Shixiang Gu, Sergey Levine, Ilya Sutskever, and Andriy Mnih. Muprop: Unbiased backpropagation for stochastic neural networks. *arXiv preprint arXiv:1511.05176*, 2015.

Jens Hainmueller and Daniel J Hopkins. The hidden american immigration consensus: A conjoint analysis of attitudes toward immigrants. *American journal of political science*, 59(3):529–548, 2015.

John R Hauser. New developments in product-line optimization. 2011.

Maren Hein, Peter Kurz, and Winfried J Steiner. Analyzing the capabilities of the hb logit model for choice-based conjoint analysis: a simulation study. *Journal of Business Economics*, 90(1):1–36, 2020.

Joel Huber. Conjoint analysis: how we got here and where we are (an update). In *Sawtooth software conference*, volume 98382, page 360. Citeseer, 2005.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Rajeev Kohli and Ramesh Krishnamurti. Optimal product design using conjoint analysis: Computational complexity and algorithms. *European Journal of Operational Research*, 40(2):186–195, 1989.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free!, 2019.

Warren F Kuhfeld, Randall D Tobias, and Mark Garratt. Efficient experimental design with marketing research applications. *Journal of Marketing Research*, 31(4):545–557, 1994.

Russell Z Kunes, Mingzhang Yin, Max Land, Doron Haviv, Dana Pe'er, and Simon Tavaré. Gradient estimation for binary latent variables via gradient variance clipping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8405–8412, 2023.

Jordan J Louviere and George Woodworth. Design and analysis of simulated consumer choice or allocation experiments: an approach based on aggregate data. *Journal of marketing research*, 20(4):350–367, 1983.

R Duncan Luce and John W Tukey. Simultaneous conjoint measurement: A new type of fundamental measurement. *Journal of mathematical psychology*, 1(1):1–27, January 1964.

Daniel McFadden. The measurement of urban travel demand. *Journal of public economics*, 3(4):303–328, 1974.

Oded Netzer and Visvanathan Srinivasan. Adaptive self-explication of multiattribute preferences. *Journal of Marketing Research*, 48(1):140–156, 2011.

Friederike Paetz, Winfried Steiner, and Harald Hruschka. Optimal product line designing: A comparison of different discrete choice-based approaches. In *EMAC 2021 Annual Conference*, May 2021.

Lijing Qin, Shouyuan Chen, and Xiaoyan Zhu. Contextual combinatorial bandit and its application on diversified online recommendation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 461–469. SIAM, 2014.

Denis Sauré and Juan Pablo Vielma. Ellipsoidal methods for adaptive choice-based conjoint analysis. *Operations Research*, 67(2):315–338, 2019.

Fransisca Susan, Negin Golrezaei, Ehsan Emamjomeh-Zadeh, and David Kempe. Active learning for non-parametric choice models. *arXiv preprint arXiv:2208.03346*, 2022.

Olivier Toubia, Duncan I Simester, John R Hauser, and Ely Dahan. Fast polyhedral adaptive conjoint estimation. *Marketing Science*, 22(3):273–303, 2003.

Olivier Toubia, John R Hauser, and Duncan I Simester. Polyhedral methods for adaptive choice-based conjoint analysis. *Journal of Marketing Research*, 41(1):116–131, 2004.

Olivier Toubia, John Hauser, and Rosanna Garcia. Probabilistic polyhedral methods for adaptive choice-based conjoint analysis: Theory and application. *Marketing Science*, 26 (5):596–610, 2007.

Kenneth E Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.

Stelios Tsafarakis. Redesigning product lines in a period of economic crisis: a hybrid simulated annealing algorithm with crossover. *Annals of Operations Research*, 247:617–633, 2016.

George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pages 2627–2636, 2017.

Yuanjia Wang, Peng Wu, Ying Liu, Chunhua Weng, and Donglin Zeng. Learning optimal individualized treatment rules from electronic health record data. *IEEE International Conference on Healthcare Informatics. IEEE International Conference on Healthcare Informatics*, 2016:65–71, October 2016.

Mingzhang Yin and Mingyuan Zhou. Arm: Augment-reinforce-merge gradient for stochastic binary networks. In *International Conference on Learning Representations*, 2019.

Mingzhang Yin, Nhat Ho, Bowei Yan, Xiaoning Qian, and Mingyuan Zhou. Probabilistic best subset selection via gradient-based optimization. *arXiv preprint arXiv:2006.06448*, 2020.

Appendix A. Related Work

Conjoint analysis. Conjoint analysis (CA) uses a survey-based experiment design to measure multidimensional preferences (Luce and Tukey, 1964; Toubia et al., 2003). We collect the data with a paired comparison survey similar to CA. Different from CA, GBS does not assume a linear additive utility model and is applicable to cases with nonlinear interactive utilities. CA estimates partworths as the choice model parameters (Hainmueller and Hopkins, 2015) while GBS does not estimate a choice model and identifies the optimal product using the choice data directly.

Adaptive experimental design. Adaptive conjoint analysis (ACA) progressively refines the attribute levels presented to respondents for more accurate and efficient data collection (Toubia et al., 2004). For example, D-efficiency is designed to maximize Fisher information (Kuhfeld et al., 1994), polyhedral methods combine geometric intuition and analytic center technique to shrink feasible region (Sauré and Vielma, 2019; Toubia et al., 2003, 2007), and adaptive self-explication integrates attribute importance in design (Netzer and Srinivasan, 2011). It might be challenging to generalize heuristics to high-dimensional features. GBS design is derived from variance reduction of score function gradient and is aligned with the uncertainty reduction intuitions. Apart from ACA, an active learning approach is proposed to learn nonparametric choice models using a directed acyclic graph. Nodes in such a graph are the alternative profiles, which limit the number of nodes to a small scale (Susan et al., 2022).

Product design by optimization. Existing product design methods often adopt optimization heuristics like Genetic Algorithms (Balakrishnan and Jacob, 1996; Balakrishnan et al., 2004), simulated annealing (Tsafarakis, 2016), evolutionary algorithm, and beam search (Hauser, 2011; Paetz et al., 2021). Though achieving empirical improvements, the properties and generalization abilities of the heuristics is largely unclear. Discrete optimization methods such as Lagrangian relaxation with branch-and-bound have also been applied (Belloni et al., 2008; Camm et al., 2006). More broadly, deep learning methods such as variational autoencoders are used for the design of product aesthetics (Burnap et al., 2023).

Policy learning. The customized product design is related to policy learning. The offline policy learning often estimates the outcome function (Wang et al., 2016) or directly optimizes the value function by propensity weighting (Athey and Wager, 2017). However, it is hardly possible for the observational data to contain the outcomes of all the actions from a combinatorial space. The learned policy might be suboptimal due to the overlapping issue. Online policy learning uses bandits and reinforcement learning to maximize the cumulative reward. The combinatorial actions are studies with mixed integer optimation in reinforcement learning (Delarue et al., 2020). Qin et al. (2014) explores combinatorial action spaces for bandits, which require users to select actions from the complete action space and need external covariates for each feature dimension. GSB overcomes these challenges using the partial profile design from the conjoint analysis.

Appendix B. Proof

This section provides the proof of Lemma 1 and Lemma 2.

Proof [Lemma 1] The optimization objective is

$$V(\phi) = \mathbb{E}_{Z\sim p(Z;\sigma(\phi))}\mathbb{E}[Y_i(Z, Z_0) \,|\, Z],$$

where $p(Z; \sigma(\phi)) = \prod_{k=1}^{K} \mathrm{Bern}(z_k; \sigma(\phi_k))$, $Z = (z_1, \cdots, z_K)$. The k-th element for the gradient of $V(\phi)$ is

$$\nabla_{\phi_k} V(\phi) = \mathbb{E}_{\boldsymbol{z}_{\setminus k}\sim\prod_{\nu\neq k}\mathrm{Bern}(z_\nu;\sigma(\phi_\nu))}\{\nabla_{\phi_k}\mathbb{E}_{z_k\sim\mathrm{Bern}(\sigma(\phi_k))}\mathbb{E}[Y_i(Z, Z_0) \,|\, Z]\} \tag{7}$$

Denote $f(z_k) = \mathbb{E}[Y_i(Z, Z_0) \,|\, Z]$, we have

$$\begin{aligned}
&\nabla_{\phi_k}\mathbb{E}_{z_k\sim\mathrm{Bern}(\sigma(\phi_k))}[f(z_k)]\\
=&\sigma(\phi_k)\sigma(-\phi_k)[f(1) - f(0)]\\
=&\mathbb{E}_{u\sim\mathrm{Unif}(0,1)}[f(\mathbf{1}[u < \sigma(\phi)])(1 - 2u)]\\
=&\mathbb{E}_{u\sim\mathrm{Unif}(0,1)}[f(\mathbf{1}[u < \sigma(\phi)])(1/2 - u)] + \mathbb{E}_{\tilde{u}\sim\mathrm{Unif}(0,1)}[f(\mathbf{1}[\tilde{u} < \sigma(\phi)])(1/2 - \tilde{u})]\\
=&\mathbb{E}_{u\sim\mathrm{Unif}(0,1)}\left[\big(f(\mathbf{1}[u > \sigma(-\phi)]) - f(\mathbf{1}[u < \sigma(\phi)])\big)(u - 1/2)\right].
\end{aligned} \tag{8}$$

The first two equations can be straightforwardly evaluated since it is an expectation with a scalar variable. The third equation is applying antithetic sampling with $\tilde{u} = 1 - u$. The last equation is summing up the two expectations. Plugging Eq. (8) into Eq. (7) gives

$$\begin{aligned}
&\nabla_{\phi_k} V(\phi)\\
=&\mathbb{E}_{\boldsymbol{z}_{\setminus k}\sim\prod_{\nu\neq k}\mathrm{Bern}(z_\nu;\sigma(\phi_\nu))}\big\{\mathbb{E}_{u_k\sim\mathrm{Unif}(0,1)}\Big[\big(\mathbb{E}[Y_i(Z, Z_0) \,|\, Z = (\boldsymbol{z}_{\setminus k}, z_k = \mathbf{1}[u_k > \sigma(-\phi_k)])] -\\
&\mathbb{E}[Y_i(Z, Z_0) \,|\, Z = (\boldsymbol{z}_{\setminus k}, z_k = \mathbf{1}[u_k < \sigma(\phi_k)])]\big)(u_k - \frac{1}{2})\Big]\\
=&\mathbb{E}_{\boldsymbol{u}\sim\prod_{k=1}^{K}\mathrm{Unif}(0,1)}\Big[\big(\mathbb{E}[Y_i(Z, Z_0) \,|\, Z = \mathbf{1}[\boldsymbol{u} > \sigma(-\boldsymbol{\phi})])] - \mathbb{E}[Y_i(Z, Z_0) \,|\, Z = \mathbf{1}[\boldsymbol{u} < \sigma(\boldsymbol{\phi})])]\big)(u_k - \frac{1}{2})\Big]
\end{aligned}$$

Therefore,

$$\nabla_{\phi} V(\phi) = \mathbb{E}_{u\sim\prod_{k=1}^{K}\mathrm{Unif}(0,1)}\left[\mathbb{E}[(Y_i(Z_1(u), Z_0) - Y_i(Z_2(u), Z_0))(u - \frac{1}{2}) \,|\, u]\right], \tag{9}$$

where $Z_1(u) = \mathbf{1}[u > \sigma(-\phi)], Z_2(u) = \mathbf{1}[u < \sigma(\phi)]$. ■

Proof [Lemma 2] $Y(Z_1, Z_0) - Y(Z_2, Z_0) \,|\, Y(Z_1, Z_0) \neq Y(Z_2, Z_0) \overset{d}{=} (2Y(Z_1, Z_2) - 1)$ Denote $S = Y(Z_1, Z_0) - Y(Z_2, Z_0)$, $T = 2Y(Z_1, Z_2) - 1$. We have

$$\begin{aligned}
p(S = 1) =&p(Y(Z_1, Z_0) = 1, Y(Z_2, Z_0) = 0)\\
=&\frac{e^{U(Z_1)}}{e^{U(Z_1)} + e^{U(Z_0)}} \cdot \frac{e^{U(Z_0)}}{e^{U(Z_2)} + e^{U(Z_0)}}
\end{aligned}$$

and

$$
\begin{aligned}
&p(S=1 \mid Y(Z_1,Z_0) \neq Y(Z_2,Z_0)) \\
=&\frac{p(Y(Z_1,Z_0)=1, Y(Z_2,Z_0)=0)}{p(Y(Z_1,Z_0)=1, Y(Z_2,Z_0)=0) + p(Y(Z_1,Z_0)=0, Y(Z_2,Z_0)=1)} \\
=&\frac{e^{U(Z_1)+U(Z_0)}}{e^{U(Z_1)+U(Z_0)} + e^{U(Z_2)+U(Z_0)}} \\
=&\frac{e^{U(Z_1)}}{e^{U(Z_1)} + e^{U(Z_2)}} \\
=&p(Y(Z_1,Z_2)=1) \\
=&p(T=1).
\end{aligned}
$$

Similarly, we have $p(S=-1 \mid Y(Z_1,Z_0) \neq Y(Z_2,Z_0)) = p(T=-1)$. Therefore, $S \mid Y(Z_1,Z_0) \neq Y(Z_2,Z_0) \overset{d}{=} T$. ■

Proof [Lemma 3] Denote the event $\mathcal{A}_i = \{Y_i(Z_1,Z_0) \neq Y_i(Z_2,Z_0)\}$, we have

$$
\begin{aligned}
&\mathbb{E}_{u\sim\prod_{k=1}^K \mathrm{Unif}(0,1)}\left[\mathbb{E}\Big[(Y_i(Z_1(u),Z_0) - Y_i(Z_2(u),Z_0))(u-\frac{1}{2}) \mid u\Big]\right] && (10)\\
=&\mathbb{E}_{u\sim\prod_{k=1}^K \mathrm{Unif}(0,1)}\left[\mathbb{E}\Big[(Y_i(Z_1(u),Z_0) - Y_i(Z_2(u),Z_0))(u-\frac{1}{2}) \mid u, \mathcal{A}_i\Big]p(\mathcal{A}_i)\right] && (11)\\
=&\mathbb{E}_{u\sim\prod_{k=1}^K \mathrm{Unif}(0,1)}\left[\mathbb{E}\Big[(2Y_i(Z_1(u),Z_2(u)) - 1)(u-\frac{1}{2})p(\mathcal{A}_i) \mid u\Big]\right]. && (12)
\end{aligned}
$$

The first equality is by the law of total expectation; the second is by Lemma 2 and the law of unconscious statistician (LOTUS). The Monte Carlo estimation of the gradient in **??** is $\tilde{g} = (2Y_i(Z_1(u),Z_2(u)) - 1)(u-\frac{1}{2})p(\mathcal{A}_i)$, $u \sim \prod_{k=1}^K \mathrm{Unif}(0,1)$. ■

Appendix C. Experimental Details

Data generation. We consider three types of representative utilities $V_i(Z)$. The first type is linear utility $V_i(Z) = W_i^\top Z$. The second type includes pairwise interactions $V_i(Z) = W_i^\top Z + \sum_{k,k'} \tilde{W}_i^{kk'} Z_k Z_{k'}$ where $\tilde{W}_i^{kk'} \sim \mathrm{Unif}(-2a, 0)$. For $K=10$, all the pairwise interactions are included, and for $K=100$, a subset of 100 pairwise interaction terms is used to compute $V_i(Z)$. The third type set $V_i(Z) = f_0(Z)$ where $f_0(\cdot)$ is a pre-trained neural network. This type of utility includes higher-order interactions of the product features. The data for each method are the query product features and the choices collected from paired choice questions, i.e., $\{(Y_i(Z_1^{ij}, Z_0^{ij}), Z_1^{ij}, Z_0^{ij})\}_{i=1:N}^{j=1:n_q}$. For non-adaptive methods, the item pair in a question is generated randomly. Each respondent makes $n_q = 10$ times the choices.

Baselines. A logistic model (Logistic) assumes $\hat{V}_i(Z) = W^\top Z$. For a pair of products (Z_1, Z_2), the likelihood of choosing Z_1 is $1/(1+\exp(W^\top(Z_2 - Z_1)))$. The parameter W is estimated by maximum likelihood and the optimal product is $\mathbf{1}[\hat{W}_{MLE} > 0]$. A mixed logit model is a hierarchical Bayes (HB) model widely used in conjoint analysis (Allenby et al., 2005). It assumes $m \sim \mathcal{N}(0, I)$, $w_i \,|\, m \sim \mathcal{N}(\mu, I)$, and $p(Y(Z_1, Z_2) = 1 \,|\, w_i, Z_1, Z_2) = 1/(1 + \exp(w_i^\top (Z_2 - Z_1)))$. We estimate m as the maximum a posteriori estimation using the PyMC package, and the estimated optimal product is $\mathbf{1}[\hat{m}_{MAP} > 0]$. Another baseline takes the representative utility $\hat{V}_i(Z) = f_\gamma(Z)$. $f_\gamma(\cdot)$ is a feedforward neural network (NN) with two hidden layers and parameters γ. γ is estimated as an MLE. The estimated optimal product is $\arg\max_Z f_\gamma(Z)$, which requires enumerating all possible products.

Evaluation metrics. The first metric for evaluating a chosen product is the average utility on a hold-out test set. It compares the relative performance of different methods. When computationally feasible, we also rank all the possible products according to their average utility on the population from high to low. The ranking allows the evaluation of absolute performance compared to the global optimum.

Appendix D. Data Generating Process for Personalized Product Design Experiments

The data is generated the same as before, except the individual preferences follow a mixture distribution $W_i \sim 0.5\mathcal{N}(\mu_1, \Sigma) + 0.5\mathcal{N}(\mu_2, \Sigma)$ where μ_1 is positive on the first half of elements and negative on the others, and μ_2 is opposite. The mixture distribution reflects the preference heterogeneity in the population. We assume the observed covariates $X_i = \exp(W_i)$ of individual i and take the utility as the nonlinear Type 2. We modify the neural network utility model as $\hat{V}(Z, X_i) = Z^\top f_\gamma(X_i)$ (denoted as NN-ind). The utility form gives an estimated optimal product for individual i as $\mathbf{1}[f_\gamma(X_i) > 0]$ without the need to evaluate all possible products' utility for each individual.

Appendix E. Additional Results

This section contains the product design algorithms and additional results for the empirical study in § 5. Fig. 5 shows the utility and ranking for the NN baseline with a large number of respondents.

Algorithm 1 Gradient-based Survey for Product Design

```
input : Number of features K, number of questions per respondent n_q, stepsize η
Initialize the logits φ randomly
while not converged do
    Sample a random individual i from the population.
    for j = 1, 2 ··· n_q do
        Sample u ~ ∏_{k=1}^{K} Unif(0, 1)
        Generate a pair of products Z_1(u) = 1[u > σ(−φ)], Z_2(u) = 1[u < σ(φ)]
        Record the respondent's choice Y_i(Z_1(u), Z_2(u))
        Compute the gradient estimate g_GBS = (2Y_i(Z_1(u), Z_2(u)) − 1)(u − 1/2)
        Update φ ← φ + η g_GBS
    end
end
```

Algorithm 2 Gradient-based Survey for Policy Learning

input : Individual covariates $\{X_i\}$, policy function $g(\cdot;\theta)$, number of features K, number of questions per respondent n_q, stepsize η

Initialize the policy parameters θ randomly.

while *not converged* **do**

  Sample a random individual i from the population.

  for $j = 1, 2 \cdots n_q$ **do**

    Sample $u \sim \prod_{k=1}^{K} \text{Unif}(0,1)$

    Generate a pair of products $Z_1(u) = \mathbf{1}[u > \sigma(-\phi)], Z_2(u) = \mathbf{1}[u < \sigma(\phi)]$

    Record the respondent's choice $Y_i(Z_1(u), Z_2(u))$

    Compute the gradient estimate $g_{\text{GBS}}(\theta)$ by Eq. (6)

    Update $\theta \leftarrow \theta + \eta g_{\text{GBS}}(\theta)$

  end

end

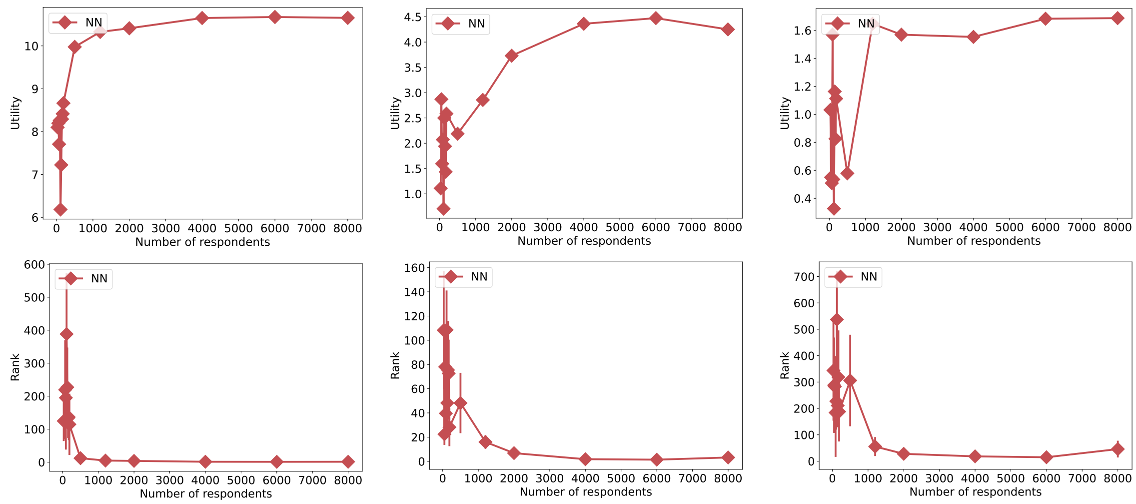
Features	A	B
Background color	Green	Blue
Shape	Round	Square
Image on the logo	Ingredient	Salad
…	…	…
Text	Lower case	Upper case
Shape outline	Yes	Yes

Figure 4: Demonstration of a paired choice question for a logo design.

Appendix F. More Discussions and Future Work

This paper bridges the domains of gradient-based machine learning and discrete choice experiments. GBS is flexible with the underlying form of choice utility, is data-efficient with adaptive design, is scalable to high-dimensional features, and can be applied to uniform or personalized product designs.

Bridging gradient-based machine learning and experiments borrow strength from both worlds. The sequential nature of SGD naturally provides an adaptive approach to design experiments. Unlike traditional heuristic or rule-based adaptive design (Green et al., 1991; Netzer and Srinivasan, 2011), the gradient method is derived mathematically and is readily to incorporate new statistical tools for variance control. The greedy property and the variance reduction technique of the gradient maximize the information extracted from each paired comparison question. The proposed GBS collects the data like a typical paired conjoint design where the respondents are randomly selected from a population and asked



(a) Type 1 (b) Type 2 (c) Type 3

Figure 5: The utility (Top row) and the ranking (Bottom row) for the NN baseline with a large number of respondents.

to choose between a pair of items with different attributes. Hence, GBS can be seamlessly integrated into commercial adaptive conjoint software like Sawtooth Software (Huber, 2005). On the other hand, the experiment offers the data generation by which machine learning can explore combinatorial action spaces. In contrast, using observational data may face overlapping and extrapolation problems since the data may only be collected with a small subset of attribute combinations in practice.

However, there is no free lunch. GBS does not estimate a choice model. It does not provide a full rank of all the possible products. To explain people's preferences, GBS may need explainable AI techniques such as saliency map (Adebayo et al., 2018) rather than estimating the preferences as a part of model parameters. Nevertheless, GBS provides a flexible optimization framework. Except for maximizing the market share, the objective may incorporate the costs and prices of a product to maximize the profit. The constrained optimization might be considered using a proximal gradient if the product design is under a budget constraint.

From a manager's view, a company often needs to design a product line consisting of several products (Balakrishnan et al., 2004; Belloni et al., 2008). One way to apply GBS for this task is by a separate approach, where the population is clustered into segments, and a single best product is determined for each segment (Paetz et al., 2021). It is also feasible to model a product line as several binary vectors and apply GBS to design a product line jointly.

GBS builds on the inference of discrete latent variables in machine learning (Dong et al., 2020; Gu et al., 2015; Jang et al., 2017; Kool et al., 2019; Kunes et al., 2023; Tucker et al., 2017). If a gradient estimator contrasts two function values, as the ARM gradient (Yin and Zhou, 2019) adopted in this paper, it can potentially be used for the adaptive question design. Combining recent discrete optimization techniques with experiments is an interesting future direction.