Preference Learning for Real-World Multi-Objective Decision Making

Zhiyuan "Jerry" Lin zylin@cs.stanford.edu
Stanford University

Adam Obeng adamobeng@fb.com
Eytan Bakshy ebakshy@fb.com
Facebook, Inc

Abstract

In experimentation at Internet firms, it is common to make product decisions based on multiple competing or complementary objectives where the experimenter needs to select the most "favorable" option with the highest utility. However, despite the prevalence of this decision problem, such multi-objective decisions are often made in arguably sub-optimal ways. In this paper, we propose a framework that enables us to suggest the configuration in a potentially large action space that is expected to maximize the experimenter's utility function. We demonstrate the efficacy of preference learning and our proposed framework with two user studies conducted at a large Internet firm. We show how the learned preference models can accurately recover machine learning engineers' preferences in the user study and are most useful when trained on near-optimal regions where real-world tradeoffs are expected to happen the most.

Keywords: Preference Learning, Multi-Objective Decision Making, Online Experimentation, Internet Experimentation

1. Introduction

Many real-world problems involve making decisions based on multiple objectives, either competing or complementary. For example, adaptive video playback control policies might be tuned to optimize the resolution, playback start time, and buffering periods of delivered video (Yin et al., 2015). While it is relatively straightforward for Internet firms to conduct large experiments which can determine the impact of an intervention on multiple outcomes, it can often be less easy to make decisions involving multiple apparent trade-offs. In these circumstances, a common approach is to produce a single objective function that combines the objective. However, it is not obvious what the functional form of such preferences should take. Moreover, it could be the case that the true objective function we care about cannot be adequately articulated by any decision maker, but can only be noisily expressed even by the experimenter herself. Nevertheless, in this case, the experimenter is often able to make comparisons between pairs of alternative possible outcomes (Tesauro, 1989; Sirakaya et al., 2004), and this information can still be used to learn a model of the experimenter's intrinsic utility function.

In this work, we learn and characterize the utility functions of machine learning engineers who wish to optimize a recommender system at Instagram, using a preference model learned with pairwise comparisons (Fürnkranz and Hüllermeier, 2003; Chu and Ghahramani, 2005;

Brochu et al., 2010). Prior literature has focused on modeling the utility function (Chu and Ghahramani, 2005; Brochu et al., 2008, 2010; Dewancker et al., 2016) or designing the preference elicitation query (Zintgraf et al., 2018), with performance being assessed with known synthetic utility functions in simulation settings. Here, we are concerned with determining real-world experimenters' utility functions, and on implementing the restriction that the outcomes considered must actually be achievable.

At Internet firms, Bayesian optimization (BO) has been proven to be an effective method for optimizing many real-world systems. For example, Letham et al. (2019) used BO to optimize ranking system and compiler flags with noisy observations and noisy constraints; Letham and Bakshy (2019) leveraged BO to tune Facebook News Feed ranking system; and Mao et al. (2019) employed BO for video playback control and reward shaping in reinforcement learning. Preferential learning has also been explored under BO (yet mostly simulated) settings (González et al., 2017; Houlsby et al., 2011; Astudillo and Frazier, 2019), Among previous work, Astudillo and Frazier (2019) presented an algorithm for multi-objective Bayesian optimization using Thompson sampling with a linear utility model. Although our approach is similar, in that we use Thompson sampling to account for uncertainty in the learned utility function, we distinguish our work from previous literature by investigating the applicability of preference models, both linear and the more expressive Gaussian processes, to the setting of Internet experimentation.

Building on past work from preference learning and Bayesian optimization, we propose a framework that not only enables us to learn the experimenter's intrinsic utility function over the achievable outcome space, but also suggests the configuration in a large action space that is expected to maximize the experimenter's utility function. We then carry out a user study to evaluate preference models and characterize real user utility functions in this setting. From the user study results, we discover that preference models trained on the near-optimal regions can perform and generalize well. We note that while the primary goal of this work is to understand the extent to which real-world experimenter's goals can be efficiently learned and used in optimization, the framework can be trivially modified to perform Bayesian optimization with interactive preference learning.

2. Method

In this section, we first describe our empirical setting then outline our proposed method for performing preference learning with both achievable and near-optimal outcomes.

2.1 Multi-Objective Decision Making for Internet Experiments

The experimenter starts by picking a set of *design points* (also referred to as *interventions* or *actions*) $\mathcal{X} = \{x_1, ..., x_n \mid x \in \mathbb{R}^k\}$, which describe parameterized experimental treatments. The experimenter then launches an experiment with these design points and observes the corresponding outcomes (or objectives) $\mathcal{Y} = \{y_1, ..., y_n \mid y \in \mathbb{R}^d\}$.

Upon observing these outcomes, the experimenter needs to evaluate the desirability of the outcomes for each design point, either to guide additional sequential experimentation or to decide on which intervention should be applied in production. If there is a single scalar outcome, doing this could be as simple as choosing the design point with the optimal value of that outcome. However when there are multiple outcomes, there need not be a single

Pareto-optimal choice. In this case the multiple outcomes need to be projected to a single value by means of a utility function.

One popular way to do this is through scalarization of the outcomes (Roijers et al., 2013; Clemen and Reilly, 2013; Fürnkranz and Hüllermeier, 2010). Here, the experimenter is assumed to have a particular intrinsic utility function $\mathcal{U} : \mathbb{R}^d \mapsto \mathbb{R}$ that will map the outcome to a single scalar value representing the experimenter-specific utility on a given outcome y. Previous work on preference elicitation has suggested that this function can be learned rather than assumed, by presenting experimenters with pairwise choices between possible outcomes. Given a set of outcome instances $\mathcal{Y}$ and m pairwise comparisons provided by the experimenter $\mathcal{D} = \{v_1 \succ u_1, ..., v_m \succ u_m\}$, where $v_j \succ u_j$ indicates the experimenter prefers input y_{v_j} over y_{u_j}, we are able to model the experimenter's intrinsic utility function. By assuming Gaussian noise on the experimenter's responses, we can describe the likelihood of a pairwise comparison $v_j \succ u_j$ as:

$$P(v_j \succ u_j \mid \mathcal{U}(y_{v_j}),\mathcal{U}(y_{u_j})) = \int\int \mathbb{1}_{\mathcal{U}(y_{v_j})+\delta_{v_j} \, \geq \, \mathcal{U}(y_{u_j})+\delta_{u_j}} \varphi_{\sigma^2}(\delta_{v_j})\varphi_{\sigma^2}(\delta_{u_j}) d\delta_{v_j} d\delta_{u_j} \tag{1}$$

$$= \Phi\left(\frac{\mathcal{U}(y_{v_j}) - \mathcal{U}(y_{u_j})}{\sqrt{2}\sigma}\right) \tag{2}$$

where $\varphi_{\sigma^2}(\cdot)$ is Gaussian PDF with mean 0 and variance σ^2 and Φ is the standard Gaussian CDF. Following previous work (Chu and Ghahramani, 2005; Brochu et al., 2008, 2010), we implement a Gaussian process (GP) utility model using Laplace approximation (MacKay, 1996) with a RBF ARD (Neal, 2012; MacKay, 1996) kernel.

2.2 Preference Learning in Achievable Near-optimal Outcome Regions

Algorithm 1: Sampling achievable and near-optimal outcomes

```
input  : Historical experiment design points and outcomes $\mathcal{X}, \mathcal{Y}$;
         User pairwise comparisons on historical experiment outcomes $\mathcal{D}$;
         The number of points for each response surface sample batch $N_{batch}$;
         Total number of candidate points to be selected $N$;
output : Candidate set $S$
Train response surface model $\mathcal{M}_{RSM}$ with $(\mathcal{X}, \mathcal{Y})$;
Train Bayesian preference model $\mathcal{M}_{pref}$ with $(\mathcal{Y}, \mathcal{D})$;
Candidate set $S \leftarrow \emptyset$;
for $i = 1$ to $N$ do
    $dp \leftarrow N_{batch}$ random points in the same space as $\mathcal{X}$;
    $outcome\_sample \leftarrow \mathcal{M}_{RSM}$.posterior($dp$).sample();
    $utility\_sample \leftarrow \mathcal{M}_{pref}$.posterior($outcome\_sample$).sample();
    $j \leftarrow$ argmax($utility\_sample$);
    $S$.add($outcome\_sample_j$)
end
return $S$
```

Many existing approaches to preference elicitation operate in (or assume) situations where every point in outcome space is achievable. However, this may not be true in practice —

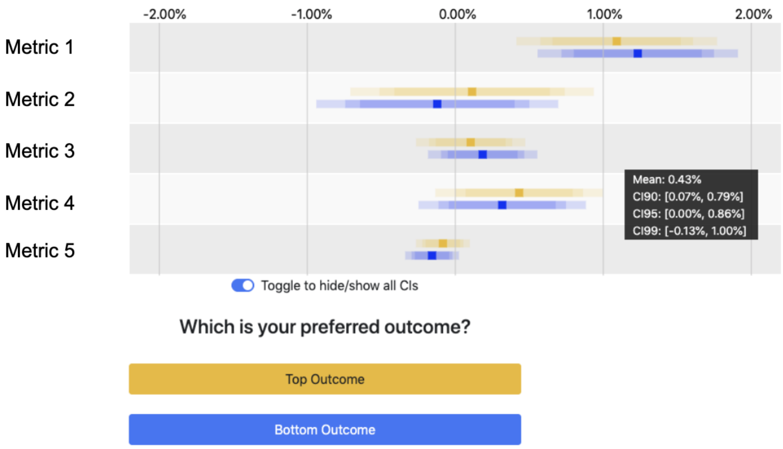

Figure 1: Preference elicitation user interface

as confirmed by feedback from the user study, which suggested that the random comparisons would rarely happen in reality because both outcomes were bad. Hence, in the rest of this subsection, we describe a framework for preference elicitation for real-world Internet experimentation which allows us to generate pairwise comparison queries about outcomes that are both achievable and relatively optimal. The algorithm is outlined in Algorithm 1.

In this framework, both our interventions and preference are parameterized, and therefore the framework uses two models: a response surface model ($\mathcal{M}_{RSM}$) which maps design points to outcomes and a Bayesian preference model ($\mathcal{M}_{pref}$) which maps outcomes to a scalar utility value. The response surface model is trained on historical experiments' design points and corresponding outcomes; the preference model can be trained with user responses on a set of (either randomly or strategically selected) outcomes. We can then execute Thompson sampling on the predicted utility posterior obtained from $\mathcal{M}_{pref}$ within the achievable outcome space provided by the response surface model $\mathcal{M}_{RSM}$. We note that we could easily perform interactive preferential Bayesian optimization with this framework by refitting preference model and regenerating the outcome samples after each round of user preference elicitation query. We implemented both the response surface model and the pairwise preferential model using `BoTorch` (Balandat et al., 2019). The linear model is implemented using `Stan` (Carpenter et al., 2017).

3. Experiments

In this section, we describe two user studies we run to evaluate the efficacy of preference learning models as well as our proposed framework in Internet experimentation setting.

3.1 User Studies

Study 1: Random Historical Outcomes The first research question we attempt to arbitrate is whether preference models can fit real-world utility functions from Internet experimentation at all given that we never explicitly observe the true underlying utility values. We select 4 historical Internet experiments conducted by a recommender system team at Instagram, each consisting of between 32 and 64 exploratory random design points. Each design point has 5 corresponding outcomes: metrics which are relevant to the success of the intervention. During the user study, we asked one machine learning engineer from Instagram who often makes experiment launch decisions to answer a series of pairwise comparison queries. In these queries, we show side-by-side two sets of randomly selected historical experiment outcomes, with the mean and confidence intervals for each metric (Figure 1).

Unlike in a simulation setting, we do not have access to the underlying true utility function and therefore cannot use model performance measures such as correlation which relies on knowing the true model (Dewancker et al., 2016). We instead measure the model's performance by examining how often it can recover the experimenter's stated preference using leave-one-out cross-validation on the collected pairwise comparisons. We observe that both the GP and linear models agree with 90% of experimenter's responses for randomly-sampled design points, indicating an overall good fit for the utility function. This similar performance suggests that the a linear (or weighted sum) utility function might be reasonable model for real-world utility functions.

Study 2: Near-Optimal Simulated Outcomes One piece of verbal feedback we heard during the the first user study was that many of the randomly selected design points almost produced undesirable outcomes, for which a decision maker would never have to choose between in practice. This not only makes the task difficult and possibly irrelevant, but also led us to explore how the learned preference function may perform when all design points are sampled from a near-optimal region of the design space, where we expect the decision tradeoffs to occur more often in practice.

The setup of the second user study is the same as Study 1, but instead of requesting the machine learning engineer to compare random outcome pairs, we ask the engineer to compare randomly selected pairs from a near-optimal outcome set generated with Algorithm 1. The preference model is pre-trained using data collected in study 1. $\mathcal{X}$ and $\mathcal{Y}$ for training those models are from the same set of historical experiments as in the first user study. $\mathcal{D}$ for training the preference model $\mathcal{M}_{pref}$ is also collected from the first user study. We generate 50 near-optimal outcomes for each of the 4 historical experiments, and collect 119 comparisons between pairs of these outcomes. Similar to Study 1, we evaluated the performance using leave-one-out cross-validation on this near-optimal outcome dataset. Although the overall accuracy is high across all dataset-model combination, there starts to show a gap between the linear model's accuracy (82%) and GP's accuracy (87%), implying that the utility function is more properly modelled by a non-linear GP model in this region.

3.2 Results and Discussion

The user study results show that although the linear model can fit the overall utility function shape fairly well, the more expressive GP model is able to model the tradeoffs better, especially for the near-optimal outcomes. To further investigate the non-linearity of the utility function in the near-optimal region and to mitigate the effect of between-subject difference in utility function shape, we compare the differences in normalized predicted utility given by GP and the linear model when training the model using only either the random

Training-Evaluating dataset	R-R	R-O	O-O	O-R
Kendall-Tau Rank Correlation	0.966	0.920	0.799	0.704
Mean Squared Difference	0.0001	0.0001	0.0210	0.0380

Table 1: Predicted utility similarity between normalized GP utility and linear model utility. Similarity is measured using both Kendall-Tau rank correlation (the higher the more similar) and the mean squared difference (the smaller the more similar). The first row in the table indicates on which dataset is the preference model being trained and evaluated on. For example, R-O suggests models are trained on random outcome dataset (R) and evaluated on the near-optimal outcome dataset (O).

outcome data or the near-optimal region data [1] (Table 1). When the models are trained on the random outcome dataset, we observe a high rank correlation and low mean squared difference between the predictions of the GP and linear model, regardless of whether utility being measured using the same random outcome dataset (*R-R*) or using the near-optimal outcome dataset (*R-O*). This suggests that the learned utility surface for the GP is linear overall, despite the fact that it uses a flexible RBF ARD kernel. On the other hand, when the GP model is trained using the near-optimal outcome dataset (*O-R* and *O-O*), the GP's behavior starts to deviate from the linear model, possibly because it learns to capture the non-linear shape along the Pareto front near the optimal region.

We also note that the high leave-one-out cross-validation accuracy only reflects the model can fit the given dataset well, but does not tell the full story of how well the learned utility surface can generalize to other datasets. To assess models' potential generalizability, we examine the cross-dataset test accuracy. Specifically, we evaluate the accuracy when we train the model on the random outcome dataset and evaluate on near-optimal outcome dataset, and the reverse scenario when we train on near-optimal outcome dataset and evaluate on the random outcome dataset. When models are trained using comparisons of the random outcomes, both models perform slightly better than chance. In contrast, when trained on near-optimal dataset, both models perform significantly better than chance on the random outcome dataset, and the GP outperforms linear model in this case. This result suggests that it is important to focus preference learning on the most optimal regions of the design space.

In this work, we propose a framework to efficiently learn a preference model in Internet experimentation settings. We then examine the efficacy of the proposed framework as well as the characteristics of real-world Internet experimenters' utility functions through user studies. We show that pure exploration using random comparison is not necessarily the optimal strategy to learn a good utility model. This work is among the first study to examine multi-objective optimization with preference learning for real Internet experimenters and tasks. We hope this study could provide insights for future researchers on better incorporating human decision preferences in Bayesian optimization as well as Internet experimentation.

[1] We limit the training data size to be the smallest dataset size in all cases during our cross-dataset performance evaluation

References

Raul Astudillo and Peter I Frazier. Bayesian optimization with uncertain preferences over attributes. November 2019.

Maximilian Balandat, Brian Karrer, Daniel R Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: Programmable bayesian optimization in PyTorch. October 2019.

Eric Brochu, Nando D Freitas, and Abhijeet Ghosh. Active preference learning with discrete choice data. In J C Platt, D Koller, Y Singer, and S T Roweis, editors, *Advances in Neural Information Processing Systems 2008*, pages 409–416. Curran Associates, Inc., 2008.

Eric Brochu, Vlad M Cora, and Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. December 2010.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

Wei Chu and Zoubin Ghahramani. Preference learning with gaussian processes. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 137–144, New York, NY, USA, 2005. ACM.

Robert T Clemen and Terence Reilly. *Making hard decisions with DecisionTools*. Cengage Learning, 2013.

Ian Dewancker, Michael McCourt, and Samuel Ainsworth. Interactive preference learning of utility functions for Multi-Objective optimization. December 2016.

Johannes Fürnkranz and Eyke Hüllermeier. Pairwise preference learning and ranking. In *European conference on machine learning*, pages 145–156. Springer, 2003.

Johannes Fürnkranz and Eyke Hüllermeier. *Preference learning*. Springer, 2010.

Javier González, Zhenwen Dai, Andreas Damianou, and Neil D Lawrence. Preferential Bayesian optimization. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1282–1291, International Convention Centre, Sydney, Australia, 2017. PMLR.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. December 2011.

Benjamin Letham and Eytan Bakshy. Bayesian optimization for policy search via online-offline experimentation. *Journal of Machine Learning Research*, 20(145):1–30, 2019.

Benjamin Letham, Brian Karrer, Guilherme Ottoni, Eytan Bakshy, et al. Constrained bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519, 2019.

David JC MacKay. Bayesian methods for backpropagation networks. In *Models of neural networks III*, pages 211–254. Springer, 1996.

Hongzi Mao, Shannon Chen, Drew Dimmery, Shaun Singh, Drew Blaisdell, Yuandong Tian, Mohammad Alizadeh, and Eytan Bakshy. Real-world video adaptation with reinforcement learning. 2019.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48: 67–113, 2013.

Ercan Sirakaya, James Petrick, and Hwan-Suk Choi. The role of mood on tourism product evaluations. *Annals of Tourism Research*, 31(3):517–539, 2004.

Gerald Tesauro. Connectionist learning of expert preferences by comparison training. In *Advances in neural information processing systems*, pages 99–106, 1989.

Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. A control-theoretic approach for dynamic adaptive video streaming over http. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, page 325–338, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450335423. doi: 10.1145/2785956.2787486. URL `https://doi.org/10.1145/2785956.2787486`.

Luisa M Zintgraf, Diederik M Roijers, Sjoerd Linders, Catholijn M Jonker, and Ann Nowé. Ordered preference elicitation strategies for supporting Multi-Objective decision making. February 2018.