Deploying a Robust Active Preference Elicitation Algorithm on MTurk: Experiment Design, Interface, and Evaluation for COVID-19 Patient Prioritization

Caroline M. Johnston
cmjohnst@usc.edu
USC Center for AI in Society
Los Angeles, California, USA

Patrick Vossler
pvossler@usc.edu
USC Center for AI in Society
Los Angeles, California, USA

Simon Blessenohl
blesseno@usc.edu
USC School of Philosophy
Los Angeles, California, USA

Phebe Vayanos
phebe.vayanos@usc.edu
USC Center for AI in Society
Los Angeles, California, USA

ABSTRACT

Preference elicitation leverages AI or optimization to learn stakeholder preferences in settings ranging from marketing to public policy. The online robust preference elicitation procedure of [34] has been shown in simulation to outperform various other elicitation procedures in terms of effectively learning individuals' true utilities. However, as with any simulation, the method makes a series of assumptions that cannot easily be verified to hold true beyond simulation. Thus, we propose to validate the robust method's performance using real users, focusing on the particular challenge of selecting policies for prioritizing COVID-19 patients for scarce hospital resources during the pandemic. To this end, we develop an online platform for preference elicitation where users report their preferences between alternatives over a moderate number of pairwise comparisons chosen by a particular elicitation procedure. We recruit 193 Amazon Mechanical Turk (MTurk) workers to report their preferences and demonstrate that the robust method outperforms asking random queries by 21%, the next best performing method in the simulated results of [34], in terms of recommending policies with a higher utility.

CCS CONCEPTS

• **Human-centered computing** → **Laboratory experiments**; **Laboratory experiments**; • **Applied computing** → *Decision analysis*; **Operations research**.

KEYWORDS

robust optimization, preference elicitation, recommender system, participatory design

1 INTRODUCTION

The difficulty in eliciting the true preferences of individuals is well documented [17]. This can occur due to various factors such as the need to elicit over a large feature space or inconsistencies when individuals report their preferences. More generally, it is difficult for individuals to express their utility exactly or in terms of specific magnitude, though it is generally agreed that, inconsistencies aside, individuals can reliably report preferences in the form of ordinal rankings.

One type of preference elicitation is conjoint analysis, where AI or optimization is used to design surveys to learn stakeholders' preferences or value judgements. Though many conjoint analysis methods attempt to address these aforementioned issues, few have been validated beyond simulation in a setting with real users. Specifically, the online robust method of [34] demonstrates strong performance in simulation in its ability to elicit individuals' preferences in settings with large feature spaces (up to 20 features) and inconsistent responses compared to other methods in the literature.

Many works involving participatory design and preference elicitation arise within social good settings, where there is an inherent tension or trade-off between fairness and efficiency, see e.g., [4]. For example, a group of policymakers may desire to design policies that match kidney donors to patients awaiting a transplant [15, 23], policies that allocate scarce resources to individuals experiencing homelessness [34], or models for recidivism prediction [35]. Specifically in [34], policymakers may need to decide between implementing a policy that assigns resources such that the number of individuals that exit homelessness is maximized overall, a policy that equalizes this outcome across different racial groups, and a policy(ies) that makes some compromise between these two outcomes. In each of these settings, an effective preference elicitation method determines which policy or model is most preferred by a policymaker as they consider these difficult trade-offs. Once this is determined, the method ultimately facilitates the decision that can be implemented in reality.

Caroline M. Johnston, Patrick Vossler, Simon Blessenohl, and Phebe Vayanos

In this work we develop an online interface for preference elicitation that evaluates the effectiveness of asking the online robust queries of [34] compared to asking randomly selected queries, the next best performing elicitation procedure in the authors' simulated experiments. Using our interface, we recruit workers from Amazon Mechanical Turk[1] (MTurk) to report their preferences between COVID-19 patient prioritization policies. Our contributions are as follows: 1) We design a preference elicitation platform implementing the approach in [34] to elicit preferences over COVID-19 patient prioritization policies using real data from the United Kingdom and recruit participants to use the platform through MTurk. 2) We conduct rigorous analysis to demonstrate that the robust method is much more likely to recommend a preferred policy to users in this real setting compared to asking random queries. Thus, we validate in an experimental setting with real users that the robust method is more effective at determining individuals' true preferences.

The rest of the paper is structured as follows. In Section 2, we review the current state of the literature. In Section 3, we summarize the robust preference elicitation and recommendation optimization model of [34]. We then discuss our COVID-19 patient prioritization setting in Section 4 and our experimental design and results in Sections 5 and 6, respectively. Finally, we discuss the limitations of the model and experiment in Section 7 and the ethical impacts of designing and implementing such a system in Section 8.

2 LITERATURE REVIEW

Preference Elicitation. Preference elicitation has been used in a plethora of AI applications from combinatorial auctions [8, 28], to homeless services [21], to food donation transportation [22], to content moderation [30]. Within healthcare, researchers have explored preference elicitation for kidney exchange [15] and services provided at rheumatology outpatient clinics [27]. Many works investigate different ways of incorporating preferences into AI systems. For example, in [15], participants are asked an exhaustive set of queries to elicit their preferences which are then encoded into a kidney exchange matching algorithm while [27] uses discrete choice experiments. Additionally, many preference elicitation works take a human-computer interaction lens, often using qualitative methods. Both [22] and [21] use interviews with real study participants to assess their opinions concerning specific algorithms in their respective domains. [30] conducts workshops in which groups collaboratively discuss which content moderation algorithm should be implemented within their community.

Conjoint Analysis. Conjoint analysis is a specific type of preference elicitation. In many conjoint analysis works, the decision-maker is able to query the user (agent) for their preferences for particular policies (items or alternatives) within a limited time frame or with a limited budget. Thus, the goal is to learn the user's true preferences (utility) by optimizing the selection of queries presented to the user. With these preferences, a suitable policy recommendation can then be made to the user.

Many works assume that the utilities of agents take a convex [1] or additive form [5, 7, 11, 32–34], facilitating the use of convex optimization tools for preference elicitation. One stream of the

literature takes a set-based approach to the uncertainty in an agent's true utility while optimizing over various types of queries. For example, [4, 32] optimize the selection of pairwise comparisons of the form "Do you prefer item A over item B?" while [33] optimizes queries that ask the agent to report the strength of their preferences between two items, i.e. "By *how much* do you prefer item A over item B?" Taking a slightly different approach, [7] optimizes the selection of gamble queries of the form "Do you prefer item A or a gamble in which you receive item A with probability p and item B with probability $1 - p$?" Though varying in the types of queries, each of these works select their respective queries such that they reduce the largest amount of uncertainty in the agent's true utility. Focusing on minimizing the regret of the recommended item, [11] uses a heuristic to select both pairwise comparison and gamble queries.

Another stream of the literature takes a Bayesian approach to the uncertainty in the agent's true utility [6, 9, 24, 29, 36]. In this setting, the uncertainty is represented by a prior distribution which is updated as the agent responds to queries. For example, [9] and [24] select gamble and pairwise comparison queries, respectively, such that they maximize the expected information gain with respect to the agent's true utility.

We now discuss the online robust preference elicitation method of [34], which selects queries that maximize the worst-case utility of the recommended item using set-based uncertainty. This method is appealing in terms of both its design and performance. For example, many of the works mentioned above focus solely on selecting queries in order to learn the true preferences of individuals. However, in many settings, the ultimate goal is to make a suitable recommendation to the individual, not necessarily an exact determination of their true utility. The authors address this by integrating the selection of queries and the recommendation as a single robust optimization problem. The method additionally accounts for inconsistencies in individuals' responses, modeling them as idiosyncratic shocks in utility that are normally distributed. In terms of the performance of the method, robust optimization is often criticized for being over-conservative. However, the authors demonstrate through simulation that the method performs well relative to the individuals' *true* utility, in addition to its realization in the worst-case, outperforming various methods in the literature. However, as noted in Section 1, it is often difficult for individuals to specify their true utility exactly, whereas in simulation these true utilities can be generated and accessed at will. Thus, it remains to be seen whether these positive results hold in a setting with real users.

Evaluation of Preference Elicitation Algorithms with Real Users. Of the conjoint analysis works above, [11, 31–33] validate their findings in settings with real users. In [33], the authors compare their polyhedral conjoint analysis method to other online elicitation procedures for recommending laptop-computer bags ($n = 330$). They validate their method's accuracy in estimating true utilities through randomly chosen holdout queries and the user's selection of a most preferred bag from 5 randomly selected bags. To validate their probabilistic polyhedral method, [31] focuses on wine consumers ($n = 2255$). They use a similar experimental design as above except each participant answers queries selected by *both* the

---

proposed method and a competing method. [32] tests their polyhedral method with real users to design an executive education program ($n = 354$), additionally examining how quickly their method converges to its final utility estimation compared to a competing method. The validation of the heuristic method of [11] involves asking users for an exhaustive listing of their preferences over 100 apartments ($n = 40$). Users then report their preferences over the union of the highly rated apartments according to their method and the highly rated apartments according to the exhaustive preference listing.

Our own experimental design to evaluate the robust method of [34] is most similar to the validation in [31] where we require users to answer both queries chosen by the robust method and queries chosen randomly. This design, as the authors of [31] note, is favorable due to the ability to compare methods within users. Thus, we can more concretely determine which method demonstrates better performance independent of potential idiosyncrasies of a given individual.

Perhaps the most wide-scale AI preference elicitation experiment is the "Moral Machine," where real users report their preferences in ethical dilemmas faced by autonomous vehicles [2]. The platform presents participants with pairwise comparisons of scenarios with randomly selected features who must choose how the autonomous vehicle behaves. Methodologically, the authors take a different approach to those above. They do not assume any functional form of the utilities of agents and take a causal inference perspective, attempting to identify the causal effects of the randomly selected features of the scenarios. The authors find that, at least at the time of the study, individuals' preferences were in direct conflict with the governmental policy guidelines proposed by the German Ethics Commission on Automated and Connected Driving. Thus, this work provides a real-life example and strong motivation for eliciting preferences in a general public policy setting. By understanding how government guidelines may conflict with the population's value judgements for such issues, policymakers can begin to understand how to better mitigate such conflicts.

3 PREFERENCE ELICITATION MODEL

In this paper, we validate the method of [34], which for the rest of this work we will refer to as Robust, on real users from MTurk. We present a high-level description of the model here in an effort to keep this work self-contained. We refer the reader to the work itself for more detail.

Given a finite set of alternatives, the goal of Robust is to recommend an alternative to an agent whose true utility is unknown. The method can gain (noisy) information about the agent's true utility by eliciting their preferences through a moderate number of queries. These queries are pairwise comparisons of the form "Do you prefer alternative A or alternative B?" The method uses concepts from robust optimization to optimize the selection of the alternatives within the queries to recommend an alternative that maximizes the agent's worst-case utility.

3.1 Utility and Query Model

Let $\mathcal{X} \subseteq \mathbb{R}^J$ be the set of alternatives from which Robust asks queries and selects a recommendation, indexed in the set $\mathcal{I} =$ $\{1, \ldots, I\}$. In Robust, each $x \in \mathcal{X}$ is modeled as a vector of J real-valued attributes. In our COVID-19 resource allocation setting, we are interested in eliciting preferences for policies that prioritize patients for treatment. We represent these policies as a vector of J efficiency and fairness metrics. In other words, this vector encodes the values of policy outcomes that policymakers may take into account when making such decisions (see Section 6.1).

In Robust, a query is a comparison between two alternatives. The set of possible queries is $C := \{(i, i') \in \mathcal{I}^2 \mid i < i'\}$. A particular choice of K queries indexed in the set $\mathcal{K} := \{1, \ldots, K\}$ is represented by $\iota \in C^K$, which specifies which alternatives are compared in the K queries. For $\kappa \in \mathcal{K}$, $\iota^\kappa := (\iota_1^\kappa, \iota_2^\kappa) \in C$ denotes that the κth query elicits the preference between alternatives $x^{\iota_1^\kappa}$ and $x^{\iota_2^\kappa}$.

Robust assumes that each agent's utility function is linear in $x \in \mathbb{R}^J$. Therefore, for an alternative $x \in \mathcal{X}$, the method represents the agent's utility for that alternative as $u^\top x$. In this setting, the true utility of the agent is uncertain and $\mathcal{U} \subseteq \mathbb{R}^J$ is used to denote the agent's initial uncertainty set, where each element $u \in \mathcal{U}$ represents one possible realization of the agent's utility function. The method assumes that $\mathcal{U}$ is a non-empty bounded polyhedron such that $\mathcal{U} = \{u \in \mathbb{R}^J \mid Bu \geq b\}$, for some $B \in \mathbb{R}^{M \times J}$, $b \in \mathbb{R}^M$.

When asked the κth query, an agent is able to respond in one of three ways using the elements of $\mathcal{S} := \{-1, 0, 1\}$: either the agent prefers alternative 1 ($s_\kappa = 1$), is indifferent between the alternatives ($s_\kappa = 0$), or prefers alternative 2 ($s_\kappa = -1$). Robust assumes that these responses are inconsistent or noisy and defines the parameter Γ as a threshold of inconsistency in the agent's responses. These inconsistencies are assumed to lie in the set $\mathcal{E}_\Gamma := \{\epsilon \in \mathbb{R}_+^K : \sum_{\kappa \in \mathcal{K}} \epsilon_\kappa \leq \Gamma\}$ and that the elements of $\epsilon \in \mathbb{R}_+^K$ are independent and normally distributed with standard deviation σ.

For each $\kappa \in \mathcal{K}$, once the agent responds to query ι^κ with response s_κ, Robust updates the uncertainty set for the agent's utility according to

$$\mathcal{U}_{\Gamma(\kappa)}(\bar{\iota}^\kappa, \bar{s}_\kappa) := \left\{ \begin{array}{c} u \in \mathcal{U} : \exists \ \epsilon \in \mathcal{E}_{\Gamma(\kappa)} \text{ s.t. } \forall k \in \{1, \ldots, \kappa\} \\ u^\top \left(x^{\iota_1^k} - x^{\iota_2^k} \right) \geq -\epsilon_k : s_k = 1 \\ \left| u^\top \left(x^{\iota_1^k} - x^{\iota_2^k} \right) \right| \leq \epsilon_k : s_k = 0 \\ u^\top \left(x^{\iota_1^k} - x^{\iota_2^k} \right) \leq \epsilon_k : s_k = -1 \end{array} \right\},$$

where $\bar{\iota}^\kappa := \{\iota^k\}_{k=1}^\kappa$ and $\bar{s}_\kappa := \{s_k\}_{k=1}^\kappa$, denote the κ queries and responses, respectively, that have been elicited thus far and $\Gamma(k)$ denotes the level of inconsistency at query $k \in \{1, \ldots, \kappa\}$. By the assumptions on ϵ, we have that $\Gamma(k) = 2\sigma\sqrt{k} \, \mathrm{erf}^{-1}(2p - 1)$ for $k \in \{1, \ldots, \kappa\}$, ensuring that $\epsilon \in \mathcal{E}_{\Gamma(k)}$ with probability p.

3.2 Preference Elicitation Optimization

For a set of alternatives $\mathcal{X}$ and agent's initial uncertainty set $\mathcal{U}$, Robust aims to solve the following *robust recommendation problem* to select an alternative that will maximize the agent's worst-case utility

$$z := \max_{x \in \mathcal{X}} \min_{u \in \mathcal{U}} u^\top x. \tag{$R^{\mathcal{U}}$}$$

As mentioned above, the method is able to query the agent and update its knowledge of the agent's true utility in the uncertainty set relative to these queries and respective responses. This is referred to as *online* elicitation, in which queries are selected one

at a time. As each query is selected and the agent provides their response, the method integrates this information into the uncertainty set and uses it to adaptively select the next query. This online elicitation procedure will ask more informative queries and lead to a recommendation with higher worst-case utility than those chosen offline, or chosen all at once before receiving any of the agent's responses, as shown in [34].

Robust solves a series of optimization problems for each κ for $\kappa = 1, \ldots, K$, updating the uncertainty set as it elicits the agent's preference at each query. Specifically, for each κ, the method solves the following problem

$$\max_{\iota^\kappa \in C} \min_{s_\kappa \in \mathcal{S}} \max_{x \in \mathcal{X}} \min_{u \in \mathcal{U}_{\Gamma(\kappa)}(\bar{\iota}^\kappa, \bar{s}^\kappa)} u^\top x, \qquad (1)$$

which can be reformulated as a finite mixed-binary linear program (see [34] for details). Note that in the problem above, $\{\iota^k\}_{k=1}^{\kappa-1}$ and $\{s_k\}_{k=1}^{\kappa-1}$ are data according to the queries and responses that have been previously elicited. As formulated above, the method is robust with respect to both the worst-case response to the selected query $s_\kappa \in \mathcal{S}$ and the realization of the agent's utility coefficients $u \in \mathcal{U}_{\Gamma(\kappa)}(\bar{\iota}^\kappa, \bar{s}^\kappa)$ as characterized by the selected queries and respective responses. Finally, we point out that Problem (1) is a deterministic process. Thus, we can compute each optimal query $\iota^\kappa \in C$ for every possible sequence of previous responses $\{s_k\}_{k=1}^{\kappa-1}$ that a user can give. We will leverage this in our experimental design (see Section 5.2).

4 COVID-19 PATIENT PRIORITIZATION

We will evaluate Robust through the problem of designing policies that allocate scarce critical care unit (CCU) beds during the COVID-19 pandemic, using real data from the United Kingdom. Throughout the pandemic, many hospitals have become inundated with an influx of patients, leading to resource shortages of life-saving equipment such as ventilators and CCU beds [25]. Doctors must then decide which patients receive this scarce equipment and who must go without. Not only does this impose an emotional burden on doctors and medical staff [13, 18], but without a disciplined way of allocating these resources, this can likely lead to an inefficient, unfair, or inconsistent allocation. Therefore, hospitals implement policies that assign these scarce resources to patients in times of overwhelming need. There currently exist many candidate policies for prioritizing patients for resources, from first-come-first-served, to point scoring rules based on the severity of disease, to a prediction of probability of hospital death (see e.g., [12, 26] for surveys and discussion). Each of these policies has unique performance outcomes and ethical implications that must be considered when implementing them in a real hospital. Thus, the use of a preference elicitation tool on stakeholders such as doctors can bring great value in determining the allocation policy that these stakeholders most prefer.

5 EXPERIMENTAL DESIGN

We now describe the experiment we have designed to evaluate the effectiveness of Robust in a setting with real users from MTurk. We recruit participants to report their preferences in a questionnaire of pairwise comparisons chosen by Robust and randomly selected pairwise comparisons. As the final query in the questionnaire, users

directly report their preference between the robustly recommended alternatives according to each procedure, determining which is more effective at recommending an item that the user prefers.

5.1 Participants and Incentives

We recruited 250 participants from MTurk to take our questionnaire. We collected participants' demographic information such as age group, race/ethnicity, gender, and whether the individual works in healthcare or not. All participants were required to be at least 18 years old, have access to the Internet, and be proficient in English. Participants were allowed to take the survey once with a time limit of two hours. They were paid $2.5 once they finished answering all pairwise comparisons in the questionnaire. Our study is IRB approved through an exempt review.

5.2 Procedure

Pairwise Comparisons. To successfully complete the questionnaire, each participant answers $2K + 1$ pairwise comparison queries. Each user is randomly assigned to one of two groups. The first group answers K queries that are chosen by Robust as described in Section 3.2, a set of "memory-cleansing" questions (see below), and then K queries that are chosen randomly. The second group takes the survey in the opposite order, first answering the randomly selected queries, "memory-cleansing" questions, and then the queries chosen by Robust. For each query, we randomize the order of its presentation, i.e., which alternative will appear on the left side of the screen versus the right side of the screen. This randomization of placement on the screen and the random assignment of users to each group offsets the presence of order biases that has been well-studied in choice-based conjoint analysis, see e.g. [10].

Regardless of which group the user is placed in, once they have completed the $2K$ queries described above, we then compute the recommended alternatives for the user according to Problem $R^{\mathcal{U}}$ relative to the selected queries and responses for each method. More concretely, let $\iota^{\text{robust}} \in C^K$ and $s^{\text{robust}} \in \mathcal{S}^K$ be the queries chosen by Robust and corresponding responses, respectively. We solve Problem $R^{\mathcal{U}_\Gamma(\iota^{\text{robust}}, s^{\text{robust}})}$ to obtain $x^*_{\text{robust}} \in \mathcal{X}$, the optimal alternative to recommend to the user with respect to the preference information elicited by Robust, and $z^*_{\text{robust}} \in \mathbb{R}$, the worst-case utility guaranteed by this recommendation. We similarly solve Problem $R^{\mathcal{U}_\Gamma(\iota^{\text{rand}}, s^{\text{rand}})}$ to obtain $x^*_{\text{rand}} \in \mathcal{X}$ and $z^*_{\text{rand}} \in \mathbb{R}$ for the queries selected randomly and corresponding responses, respectively.

The $(2K + 1)$th, or final, pairwise comparison we present to the user is a comparison between x^*_{robust} and x^*_{rand} where we similarly randomize the presentation of the alternatives. Since each alternative is the optimal recommendation under each elicitation procedure, this final comparison enables us to directly determine which of the two methods recommends the item that is preferred.

Memory-cleansing and Attention-checking. Memory-cleansing questions play an important role in our experiment design by allowing us to directly compare the performance of both methods on each user. For this process, users answer the following three questions, known as the "Cognitive Reflection Test" [14]:

(1) A bat and a ball cost \$1.10 in total. The bat costs \$1.00 more than the ball. How much does the ball cost?
(2) If it takes 5 machines 5 minutes to make 5 widgets, how long would it take 100 machines to make 100 widgets?
(3) In a lake, there is a patch of lily pads. Every day, the patch doubles in size. If it takes 48 days for the patch to cover the entire lake, how long would it take for the patch to cover half of the lake?

These questions serve two purposes. First, by having the users focus on this unrelated task before switching from one part of the questionnaire to the other, these questions help to reduce the presence of a carry-over effect from method to method. Secondly, we require the users to answer these questions as free-form text as a form of attention-checking. Note that we do not check for correctness of the users' answers but rather that the answers are "sensible", e.g., not a random string of characters or other "bot-like" behavior. We also record the time it took for the users to answer each pairwise comparison query for *post hoc* attention-checking analysis (see Section 6.3).

Robust *Query and Recommendation Calculation.* We recall that, the formulation of Problem (1) is a deterministic process and that we can compute each optimal query for every possible sequence of responses that a user can give before running the experiment. By storing these optimal queries in a lookup table, this further reduces the potential bias between the two query selection methods. Now each query that is presented to the user can be shown "instantaneously" whether it is selected by Robust or randomly. In other words, there is no need to wait for Robust to solve its optimization problem to select the next query to ask.

To create the lookup table we solve the mixed-binary linear programming formulation of Problem (1) for $K = 10$ and with initial partworth uncertainty set $\mathcal{U} = \left\{ \boldsymbol{u} \in \mathbb{R}_+^J : \boldsymbol{u}^\top \boldsymbol{e} = 1 \right\}$, as in the experimental results of [34]. We solve this for every possible sequence of responses $\{s_k\}_{k=1}^{\kappa-1}$ that a user can report for $\kappa \in \mathcal{K}$. We generate each $\Gamma(k)$ according to the formula given in Section 3.1 with $p = 90\%$ and $\sigma = 0.1$, representing a 90% confidence level that the ϵ_k lie within $\mathcal{E}_{\Gamma(k)}$. We use the same p value as [34] but choose a slightly larger σ value than what the authors present. We use this larger σ value since we cannot validate whether a choice of σ is misspecified without access to the user's true utilities, the difficulty of which in an experimental setting with real users is discussed in Sections 1 and 2. We note that by using a higher value of σ, we decrease the chance of misspecifying the model of Robust at the risk of obtaining over-conservative results. Nevertheless, we demonstrate that, even if this choice of σ is over-conservative, Robust performs well in this setting (see Section 6.4). To conclude, with the parameters above, we solve Problem (1) for each $3^1 + 3^2 + \cdots + 3^9$ possible sequence of responses. For each user's respective queries and responses, we solve the robust recommendation problems $R^{\mathcal{U}_\Gamma}(\boldsymbol{\iota}^{\text{robust}}, \boldsymbol{s}^{\text{robust}})$ and $R^{\mathcal{U}_\Gamma}(\boldsymbol{\iota}^{\text{rand}}, \boldsymbol{s}^{\text{rand}})$ with an initial partworth uncertainty set and $\Gamma = \Gamma(10)$. All problems are solved on 6 cores of Xeon 2.6GHz cores and 1GB of memory using Gurobi 10.0.0.

6 RESULTS

Using our platform, study participants take a questionnaire where they report their preferences between prioritization policies for COVID-19 patients. These policies are generated using real data from the United Kingdom as in [20]. For each user, we evaluate if the robust recommendation under the preference information obtained by Robust is preferred compared to the robust recommendation under the preference information obtained by randomly selected queries.

6.1 COVID-19 Patient Prioritization Policy Generation

We now describe how we use real data to simulate policies for prioritizing COVID-19 patients for treatment which we use as the alternatives in our experiment. We note that one could evaluate a preference elicitation method on randomly generated vectors of policy features. However, evaluating on feature vectors that are the outcomes of implementable policies computed using real data will arguably yield stronger evidence about the method's performance in real-world settings. Additionally, this differentiates our work from discrete choice experiments used in healthcare [27]. In a discrete choice experiment, features and their levels are combined and individuals are asked their preferences between such combinations in pairwise comparisons. However, it is entirely possible that no policy exists that achieves such combinations of features, especially if these features are policy outcomes. Thus, by simulating policies and using these simulated outcomes as features, we can obtain an actionable policy that aligns with individuals' preferences.

We create a simulation to estimate the outcomes of 25 counterfactual COVID-19 patient prioritization policies by age group at the country-level using data from the United Kingdom from April 1st to July 15th, 2020. As estimates of the rate of daily arrivals of patients in need of critical care, we use the United Kingdom's daily projections of the expected number of COVID-19 CCU patients over this time period from the Institute for Health Metrics and Evaluation (IHME) model.[2] As estimates of both the distribution of these patients' ages and their probability of recovery, we use the historical proportions and survival rates of COVID-19 patients in the CCU by age as provided by the UK Intensive Care National Audit and Research Centre[3] over this time period. Note that we use only age as a patient characteristic due to the unavailability of patient outcome data based on, say, both age and race.

A policy assigns a score to each patient based on their age and the number of days they have waited for a critical care unit (CCU) bed, allocating beds to the highest-scoring patients. We consider policies in the form of regression trees. For a given policy, we calculate a patient's score by starting at the root node and traversing the tree, proceeding to the left or right child of a node depending on whether the patient satisfies the given condition until a leaf node is reached. A leaf node contains a real number between 0 and 1, representing the patient's score. We generate 25 regression tree policies of depth three by randomly picking a feature and a comparison value for each branching node and a random number between 0 and 1 for the leaf nodes. These leaf node values create a priority ranking

---

[2]http://www.healthdata.org/covid/data-downloads
[3]https://www.icnarc.org/Our-Audit/Audits/Cmp/Reports

Caroline M. Johnston, Patrick Vossler, Simon Blessenohl, and Phebe Vayanos

for individuals for treatment based on patient characteristics, for which this general concept is used in many CCUs (see Section 4).

For a given prioritization policy, we simulate events in the following order for each day in the time frame of our simulation: 1) First, COVID-19 patients who require CCU beds arrive at a hospital(s) according to our estimated daily arrival rate; 2) Some patients in critical care pass away or recover according to our estimated survival rate that is dependent on their age, making these CCU beds newly available; 3) Some waiting patients, who did not receive a bed, pass away according to a constant probability independent of their age; 4) Finally, the policy assigns waiting patients to available (finite) CCU beds, making these CCU beds unavailable to other patients. If at this point there are no patients waiting for a bed, no patients in the CCU, and no expected future arrivals of patients, we terminate the simulation.

The features we estimate, and by which users will evaluate policies, are: 1) the total number of life-years saved, 2) the overall survival probability of patients, 3) the survival probabilities of patients across six age groups, and 4) the probabilities of patients receiving a CCU bed across these same six age groups. We also record the coefficients of variation (CV) with respect to metrics 3) and 4) in order to capture the notion that individuals may prefer more "fair" policies, i.e., policies with low CV values. We choose these metrics to represent various efficiency and fairness metrics that policymakers may consider in such a setting, where 1) and 2) are efficiency metrics and 3) and 4) are fairness metrics with respect to age. Specifically, 3) represents fairness in terms of outcomes and 4) represents fairness in terms of access. Thus, the trade-offs between these metrics must be considered when users report their preferences. Domain experts may include other metrics of interest as appropriate.

Because the CV value is less interpretable compared to the other features, we do not show this metric directly to the user within the pairwise comparisons. However, since Robust and the robust recommendations will have access to these values, the methods may still determine how this notion of fairness contributes to the user's utility. In total, each of the 25 randomly generated policies is characterized by 16 features, a similar scale in number of alternatives and features as tested in [34]. These 25 policies become our set of alternatives $\mathcal{X}$, for which there are 300 unique pairwise comparisons that each user could be asked.

6.2 Preference Elicitation Platform

We designed a platform[4] to evaluate Robust on MTurk over a period of two consecutive days in an effort to diversify the sample of the study population. After participants provide consent, they view the landing page of the questionnaire. This page familiarizes them with our setting, asking them to imagine that they are a healthcare professional working at a hospital during May of 2020 – before the wide-scale availability of vaccines and resources for treating COVID-19. We instruct them that their goal is to help determine a set of guidelines at a hospital to decide which patients will receive a bed, ventilator, or other lifesaving treatment in a CCU, when there are more patients than available resources. They are also given a high-level overview of the outcomes upon which they will evaluate

---

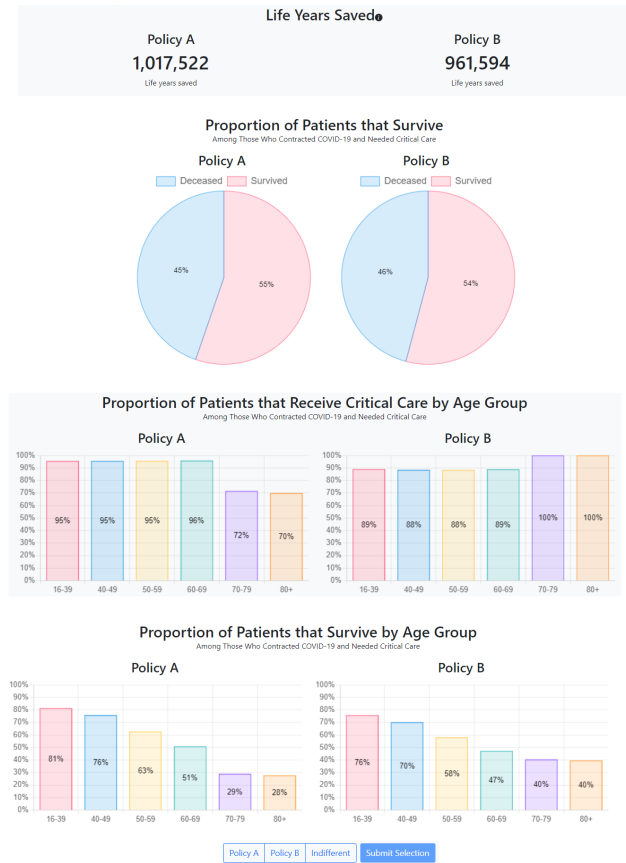[4]https://www.cais-preference-elicitation.com/

Figure 1: An example of a pairwise comparison between two COVID-19 patient prioritization policies displayed in the interface.

the policies. An example of a pairwise comparison in the platform and the outcomes of policy features is seen in Figure 1.

6.3 User Preference Data Pre-Processing

We now describe our data cleaning procedure. For any survey responses where the user took the survey multiple times, though they were instructed not to, we kept their first attempt but removed later attempts to avoid potential bias in these repeated exposures. We also manually checked and removed any users who provided bot-like answers to the memory-cleansing questions (see Section 5.2). We removed any users that took less than 15 seconds to answer the first query or that took less than an average of 3 seconds to answer the following queries as *post hoc* attention checks. We characterize these using two different cut-off values because we observed that users take a longer amount of time on the first query to become familiar with the structure of our questionnaire and interface design. As the survey continues, they may answer queries more quickly as they become more familiar with the process. We also removed an outlier user that took more than one hour to complete the survey as no other user took more than 40 minutes to complete the survey.

Table 1: Demographics of Study Participants (n = 193)

Demographic	Frequency (%)	Demographic	Frequency (%)	Demographic	Frequency (%)
White	82	18-39 years of age	78	High school or equivalent	18
Asian	11	40-49 years of age	16	Some college or more	82
Black/African American	3	50-69 years of age	6	Works in healthcare	49
Hispanic/Latino	3	Male	61	Does not work in healthcare	51
American Indian/Alaska Native	1	Female	39		

We note that our analysis below remains significant when filtering out participants who took less than 30 seconds to answer the first query and less than 5 seconds on average to answer the remaining queries. Introducing larger cut-off values removed too many samples to be able to make any statistical claims such as those that follow.

Finally, we note that it is possible that the final query in the questionnaire between $x^*_{\text{robust}} \in X$ and $x^*_{\text{rand}} \in X$ is such that the two policies are the same, i.e., $x^*_{\text{robust}} = x^*_{\text{rand}}$. From Robust's construction of queries in C and the manner in which we select random queries, this is the only time in which a query with the same two policies can be presented to a user. When this occurs, this means that the preference information gained from asking robust queries and asking random queries is equivalent insofar as the same policy will maximize the user's worst-case utility. We remove any responses in which this occurs and the user does not report that they are indifferent between the two policies as a form of attention-checking, since these two policies are exactly the same.

After we clean the data as described above, our final population size is 193 participants. We can see the demographic information of this group in Table 1. Our population skews toward young, white males with at least some college education. These types of biases are well-documented for MTurk workers, see e.g., [3, 19]. The participants took an average of 6.5 minutes with a standard deviation of 5.5 minutes to complete the questionnaire.

6.4 Analysis of User Preference Data

We analyze the unique preferences of the users, which is noteworthy for two reasons. First, many elicitation methods evaluated in simulation assume that utilities are randomly distributed over the population [5, 29, 34]. Because our study population lacks demographic diversity (see Table 1), it is important to determine if there is also a lack of diversity in preferences. Second, if the questionnaire reveals that a majority of users' preferences are the same, developing an online preference elicitation tool is moot. We could find a recommended policy that is "one-size-fits-all," no matter the user.

To determine the diversity of preferences, we recall that the robust portion of the questionnaire is the same for those that report the same preferences. Thus, a subset of users can be asked the same sequence of queries $\iota^{\text{robust}} \in C^K$ by Robust if they report the same sequence of responses $s^{\text{robust}} \in S^K$. Of the 193 users, 128 had unique preferences and, therefore, answered a unique set of queries selected by Robust. In Figure 2, we show a breakdown of the remaining 65 users who shared the same preferences with at least one other user for the queries selected by Robust. Though
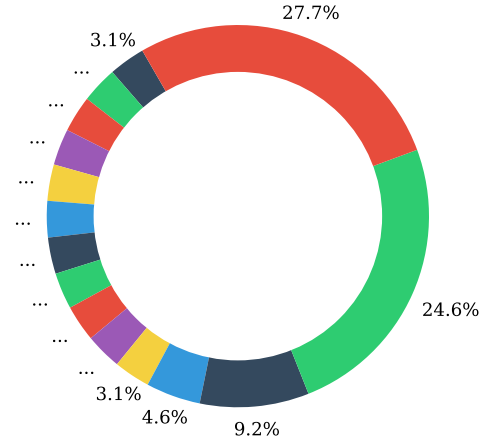

Figure 2: The percentage of users who had the same preferences as at least one other user in the study (n = 65) for the queries chosen by Robust. Each sector refers to a unique set of preferences, $s^{\text{robust}} \in S^K$.

some subsets of the study population have the same preferences, a majority differ in this regard.

Even though users may have differing preferences with respect to the individual queries selected by Robust, this may be less relevant if the policy that we recommend to these users is the same. In Figure 3, we report the number of users who prefer each policy in the last query of the questionnaire. Note that we exclude those that are indifferent in this final query. Though there are a small number of policies that many users prefer in the final query, overall, there is a high amount of variability in these responses. Thus, we can reasonably conclude that because of the diversity of preferences of the users, designing and using such an elicitation and recommendation procedure is of practical importance.

We recall that Robust selects queries that optimize the worst-case utility of the recommended item. We additionally emphasize that both x^*_{robust} and x^*_{rand} are selected to maximize the worst-case utility of the user with respect to the uncertainty set as characterized by the queries and responses obtained by each method. Thus, even though random queries are selected, the recommendation of x^*_{rand} is made in a robust manner. In Figure 4, we investigate the distribution of the difference in the worst-case utilities between the recommended policies x^*_{robust} and x^*_{rand} for each user as characterized by their respective uncertainty sets, i.e., the value of $z^*_{\text{robust}} - z^*_{\text{rand}}$.
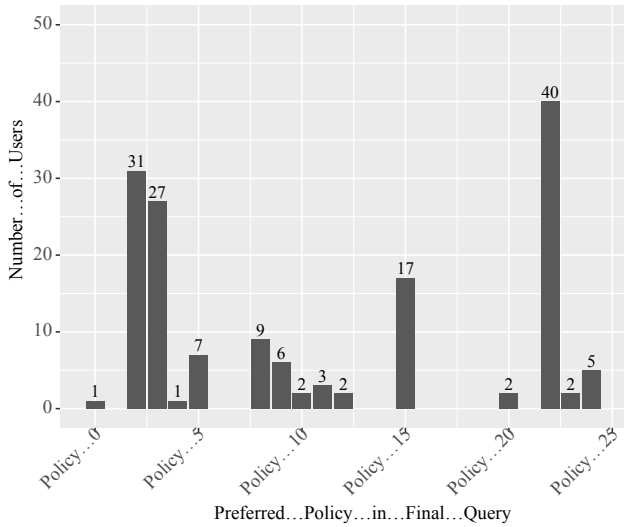
Figure 3: The number of users who select the given policy as their preferred policy in the last query of the questionnaire, which directly compares the policy recommended by Robust and the policy recommend by asking random queries ($n = 155$). Note that we exclude those that are indifferent in the last query.

Robust recommends a policy with an average worst-case utility of 0.60, while asking random queries recommends a policy with an average worst-case utility of 0.52. This difference in average guarantee of utility is statistically significant, $t(384) = 4.58, p < 0.001$ and demonstrates the benefit of Robust when optimizing for this worst-case performance.

Ultimately, we desire favorable performance in terms of the user's true utility, or how they actually report their preference. Figure 5 displays the number of users who report that they prefer x^*_{robust}, the policy that Robust recommends; prefer x^*_{rand}, the policy that asking random queries recommends; or are indifferent between the two policies. As mentioned in Section 6.3, it is possible that x^*_{robust} is the same as x^*_{rand}. Thus, we disaggregate users who report that they are indifferent by whether these policies are the same or not. By disaggregating, we can differentiate between when the two methods make an equivalent recommendation in terms of the policies themselves versus an equivalent recommendation in terms of the user's utility for two unique policies.

We first note that indifferences are reported between the two policies for ~20% of the users, i.e., the two methods are equivalent in their recommendations for ~20% of users by either recommending the same policy or two policies with similar utility. When considering users that are not indifferent in the final query ($n = 155$), Robust is able to recommend a policy that is more preferred for 33, or ~21%, more users compared to asking queries randomly.

We validate that we cannot detect various order effects in our results. We cannot detect a statistically significant difference in the results shown in Figure 5 between the users who first answered the queries selected by Robust ($n = 94$) versus those that first answered the queries selected randomly ($n = 99$), $t(191) = 0.80, p = 0.43$. We
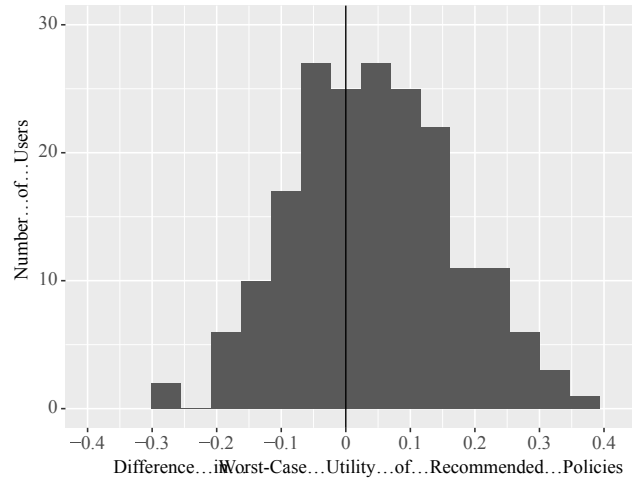

Figure 4: The distribution of the difference in the worst-case partworth utilities of the policy recommended by Robust and the policy recommended by asking random queries ($n = 193$). Any user with a positive difference value will have a higher utility in the worst-case for x^*_{robust} than for x^*_{rand}. The difference in utility values is normalized between -1 and 1, where a value of $1(-1)$ corresponds to $x^*_{\text{robust}}(x^*_{\text{rand}})$ as the recommended policy if the utility is known, i.e., the best possible recommendation, and $x^*_{\text{rand}}(x^*_{\text{robust}})$ as the recommended policy when asking no queries, i.e., the worst possible recommendation.

similarly do not have enough statistical evidence to believe there is a bias in our results concerning whether, in the final query, x^*_{robust} appears on the right side ($n = 76$) or the left side ($n = 117$) of the screen, $t(191) = 1.47, p = 0.14$.

Finally, we validate that our results in Figure 5 are not due to randomness. Specifically, we compare our results to that of users uniformly selecting their preference between the policies displayed in the final query. We focus on our results that have strict preferences, i.e., that are not indifferent ($n = 155$). This is justifiable since we observe that users are unlikely to report that they are indifferent between any two policies. Out of the 21 total queries asked to users across all questionnaires, users reported an indifference for only 9% of the queries. In fact, only 46% of users reported any indifference at all during the questionnaire, while the rest always reported strict preferences. Therefore, we do not believe there exists uniformity between the three choices and only investigate uniformity when selecting a strict preference between the two policies. To this end, we compare our results to that of a uniform distribution, or users selecting between the two policies in the final query at random, and find that our result is statistically significant and not due to randomness, $\chi^2(1, 155) = 6.61, p < 0.01$. Thus, we have significant evidence that Robust will lead to robust recommendations that users are more likely to prefer, or that have higher utility, compared to the robust recommendations that result from asking random queries.
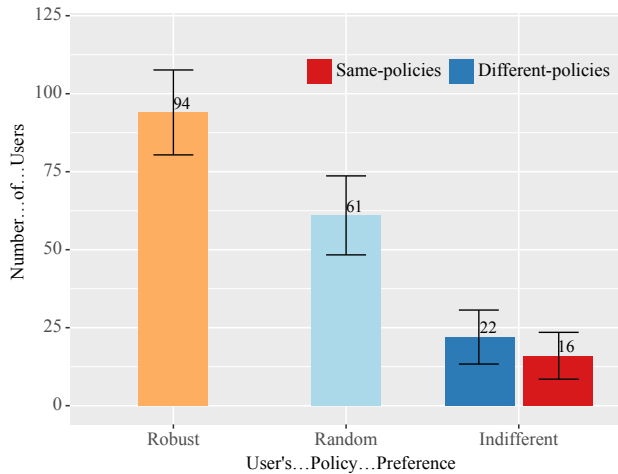
Figure 5: Results of the last query of the questionnaire which directly compares the policy recommended by Robust and the policy recommend by asking random queries ($n = 193$) with 95% confidence intervals. Robust (Random) is the number of users that selected that they prefer x^*_{robust} (x^*_{rand}). Indifferent refers to the number of users that reported that they are indifferent. Different-policies (Same-policies) refers to the instances where a user is presented with policies $x^*_{\text{robust}} \neq x^*_{\text{rand}}$ ($x^*_{\text{robust}} = x^*_{\text{rand}}$) and reports an indifference between the two.

7 LIMITATIONS

The ideal audience for our interface and application as described in Section 4 is stakeholders in the healthcare field, not necessarily members of the general public. Though about half of the participants reported that they work in healthcare (see Table 1), the ones that do not may have lacked the necessary domain knowledge to report their preferences in a faithful or meaningful way. However, we tried to familiarize all users with our particular setting through the information provided in the platform's landing page (see Section 6.2).

Our study evaluates whether Robust can recommend an alternative with higher utility compared to the recommended alternative when asking random queries. We make no claim as to whether, for example, either alternative is the user's top-ranked alternative. This would require an exhaustive listing of the user's preferences, a time-intensive process, or for the user to be able to characterize their utility exactly, which is highly impractical as discussed in Sections 1 and 2.

8 ETHICAL IMPACTS

Through our study, we have shown that using the online robust method of [34] results in a recommendation to an individual that they are more likely to prefer compared to asking queries randomly. However, as with the adoption of any AI system, the end-users of such a method must be accepting of any potential shortcomings in the method's performance. From our results in Figure 5, there certainly exist users for which Robust does not recommend

a preferred policy. However, if this method is still an improvement over stakeholders' status quo procedure for policy design and implementation, then it may be reasonable to adopt it. Furthermore, comparison of Robust to other preference elicitation methods in the literature in an experimental setting with real users is a sensible area for future exploration.

Any recommendation system faces the potential issue of automation bias, where individuals are more likely to trust in decisions proposed by AI models as opposed to humans, see e.g., [16]. To mitigate this risk in a real decision-making setting, we suggest that the policy this system recommends serves as a starting point for further refinement, as appropriate, and not necessarily strict adoption.

Finally, the elicitation and recommendation system is limited to the features that are explicitly provided in the alternatives. In our COVID-19 resource allocation policy setting in Section 6.1, there may exist other latent policy features concerning efficiency, fairness, or equity that are not stated as features of the alternatives but have real consequences if the recommended policy is implemented in reality. One way to address this is to have an initial elicitation process in which stakeholders report their preferences for what features should be represented in the alternatives [15]. Once these features are determined, the appropriate alternatives can be generated to find an agent's recommendation.

9 CONCLUSION

In this work, we develop an interface for validating the online robust preference elicitation method of [34] in settings with real users. Focusing on an application of designing COVID-19 policies that assign scarce resources to patients, we investigate this method compared to asking random queries using workers from MTurk. We find that these robust queries recommend a preferred policy for 21% more users compared to the policy that is recommended when making a robust recommendation with randomly selected queries. Thus, we validate that the robust method is an effective tool for eliciting individuals' preferences in an experimental setting with real users beyond that of simulation.

REFERENCES

[1] Benjamin Armbruster and Erick Delage. 2015. Decision making under uncertainty when preference information is incomplete. *Management Science* 61, 1 (1 2015), 111–128. https://doi.org/10.1287/mnsc.2014.2059

[2] Edmond Awad, Sohan Dsouza, Richard Kim, Jonathan Schulz, Joseph Henrich, Azim Shariff, Jean François Bonnefon, and Iyad Rahwan. 2018. The Moral Machine experiment. *Nature* 563, 7729 (11 2018), 59–64. https://doi.org/10.1038/s41586-018-0637-6

[3] Adam J. Berinsky, Gregory A. Huber, and Gabriel S. Lenz. 2012. Evaluating online labor markets for experimental research: Amazon.com's mechanical turk. *Political Analysis* 20, 3 (7 2012), 351–368. https://doi.org/10.1093/pan/mpr057

[4] Dimitris Bertsimas, Vivek F. Farias, and Nikolaos Trichakis. 2012. On the efficiency-fairness trade-off. *Management Science* 58, 12 (12 2012), 2234–2250. https://doi.org/10.1287/mnsc.1120.1549

[5] Dimitris Bertsimas and Allison O'Hair. 2013. Learning preferences under noise and loss aversion: An optimization approach. *Operations Research* 61, 5 (9 2013), 1190–1199. https://doi.org/10.1287/opre.2013.1209

[6] Craig Boutilier. 2002. A POMDP Formulation of Preference Elicitation Problems. In *Proceedings of The Eighteenth National Conference On Artificial Intelligence*. 239–246.

[7] Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans. 2006. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artificial Intelligence* 170, 8-9 (2006), 686–713. https://doi.org/10.1016/j.artint.2006.02.003

[8] Gianluca Brero, Benjamin Lubin, and Sven Seuken. 2019. Machine Learning-powered Iterative Combinatorial Auctions. (11 2019). http://arxiv.org/abs/1911.08042

[9] Urszula Chajewska, Daphne Koller, and Ronald Parr. 2000. Making Rational Decisions using Adaptive Utility Elicitation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*. 363–369.

[10] Keith Chrzan. 1994. Three Kinds of Order Effects in Choice-Based Conjoint Analysis. *Marketing Letters* 5, 2 (1994), 165–172.

[11] Darius Braziunas and Craig Boutilier. 2010. Assessing regret-based preference elicitation with the UTPREF recommendation system. In *EC '10: Proceedings of the 11th ACM conference on Electronic commerce*. Association for Computing Machinery, 219–228.

[12] Ezekiel J. Emanuel, Ross Upshur, Beatriz Thome, Michael Parker, Aaron Glickman, Cathy Zhang, Connor Boyle, Maxwell Smith, and James P. Phillips. 2020. Fair Allocation of Scarce Medical Resources in the Time of Covid-19. *New England Journal of Medicine* 382 (2020), 2049–2055.

[13] Mattia Ferraresi. 2020. A coronavirus cautionary tale from Italy: Don't do what we did. https://www.bostonglobe.com/2020/03/13/opinion/coronavirus-cautionary-tale-italy-dont-do-what-we-did/

[14] Shane Frederick. 2005. Cognitive Reflection and Decision Making. *Journal of Economic Perspectives* 19, 4 (2005), 25–42.

[15] Rachel Freedman, Jana Schaich Borg, Walter Sinnott-Armstrong, John P. Dickerson, and Vincent Conitzer. 2020. Adapting a Kidney Exchange Algorithm to Align With Human Values. *Artificial Intelligence* 283 (2020).

[16] Kate Goddard, Abdul Roudsari, and Jeremy C. Wyatt. 2012. Automation bias: A systematic review of frequency, effect mediators, and mitigators. *Journal of the American Medical Informatics Association* 19, 1 (1 2012), 121–127. https://doi.org/10.1136/amiajnl-2011-000089

[17] Paul E. Green and V. Srinivasan. 1978. Conjoint Analysis in Consumer Research: Issues and Outlook. *Journal of Consumer Research* 5, 2 (1978), 103–123. https://www.jstor.org/stable/2489001

[18] Neil Greenberg, Mary Docherty, Sam Gnanapragasam, and Simon Wessely. 2020. Managing mental health challenges faced by healthcare workers during covid-19 pandemic. *BMJ* 368 (3 2020). https://doi.org/10.1136/bmj.m1211

[19] Joel Ross, Lilly Irani, M. Six Silberman, Andrew Zaldivar, and Bill Tomlinson. 2010. Who are the crowdworkers? Shifting demographics in Mechanical Turk. In *CHI EA '10: CHI '10 Extended Abstracts on Human Factors in Computing Systems*. 2863–2872.

[20] Caroline M. Johnston, Simon Blessenohl, and Phebe Vayanos. 2020. Preference Elicitation and Aggregation to Aid with Patient Triage during the COVID-19 Pandemic. *Workshop on Participatory Approaches to Machine Learning* (2020).

[21] Tzu-Sheng Kuo, Hong Shen, Jisoo Geum, Nev Jones, Jason I. Hong, Haiyi Zhu, and Kenneth Holstein. 2023. Understanding Frontline Workers' and Unhoused Individuals' Perspectives on AI Used in Homeless Services. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. https://doi.org/10.1145/3544548.3580882

[22] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, and Ariel D. Procaccia. 2019. WeBuildAI: Participatory framework for algorithmic governance. *Proceedings of the ACM on Human-Computer Interaction* 3 (2019), 1–35. https://doi.org/10.1145/3359283

[23] Duncan C. McElfresh, Lok Chan, Kenzie Doyle, Walter Sinnott-Armstrong, Vincent Conitzer, Jana Schaich Borg, and John P. Dickerson. 2021. Indecision Modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. http://arxiv.org/abs/2012.08485

[24] Thomas Pfeiffer, Xi Alice Gao, Andrew Mao, Yiling Chen, and David G. Rand. 2012. Adaptive polling for information aggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 1. 122–128.

[25] Megan L. Ranney, Valerie Griffeth, and Ashish K. Jha. 2020. Critical Supply Shortages — The Need for Ventilators and Personal Protective Equipment during the Covid-19 Pandemic. *New England Journal of Medicine* 382, 18 (4 2020), e41. https://doi.org/10.1056/nejmp2006141

[26] Amy Grace Rapsang and Devajit C. Shyam. 2014. Scoring systems in the intensive care unit: A compendium. *Indian Journal of Critical Care Medicine* 18, 4 (2014), 220–228. https://doi.org/10.4103/0972-5229.130573

[27] Mandy Ryan, Angela Bate, Clifford J. Eastmond, and Anne Ludbrook. 2001. Use of discrete choice experiments to elicit preferences. *BMJ Quality in Health Care* (2001), i55–i60.

[28] Tuomas Sandholm and Craig Boutilier. 2006. Preference Elicitation in Combinatorial Auctions. In *Combinatorial Auctions*. The MIT Press, Chapter 10, 233–263.

[29] Denis Sauré and Juan Pablo Vielma. 2019. Ellipsoidal methods for adaptive choice-based conjoint analysis. *Operations Research* 67 (2019), 295–597.

[30] Hong Shen, Leijie Wang, Wesley H. Deng, Ciell Brusse, Ronald Velgersdijk, and Haiyi Zhu. 2022. The Model Card Authoring Toolkit: Toward Community-centered, Deliberation-driven AI Design. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. 440–451. https://doi.org/10.1145/3531146.3533110

[31] Olivier Toubia, John Hauser, and Rosanna Garcia. 2007. Probabilistic polyhedral methods for adaptive choice-based conjoint analysis: Theory and application. *Marketing Science* 26, 5 (9 2007), 596–610. https://doi.org/10.1287/mksc.1060.0257

[32] Olivier Toubia, John R. Hauser, and Duncan I. Simester. 2004. Polyhedral Methods for Adaptive Choice-Based Conjoint Analysis. *Journal of Marketing Research* 41, 1 (2004), 116–131. http://ssrn.com/abstract=382380

[33] Olivier Toubia, Duncan I. Simester, John R. Hauser, and Ely Dahan. 2003. Fast Polyhedral Adaptive Conjoint Estimation. *Marketing Science* 22, 3 (2003), 273–303. https://doi.org/10.1287/mksc.22.3.273.17743

[34] Phebe Vayanos, Yingxiao Ye, Duncan C. McElfresh, John P. Dickerson, and Eric Rice. 2020. Robust Active Preference Elicitation. (3 2020). http://arxiv.org/abs/2003.01899

[35] Bowen Yu, Ye Yuan, Loren Terveen, Zhiwei Steven Wu, Jodi Forlizzi, and Haiyi Zhu. 2020. Keeping designers in the loop: Communicating inherent algorithmic trade-offs across multiple objectives. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. Association for Computing Machinery, Inc, 1245–1257. https://doi.org/10.1145/3357236.3395528

[36] Zhibing Zhao, Haoming Li, Junming Wang, Jeffrey Kephart, Nicholas Mattei, Hui Su, and Lirong Xia. 2018. A Cost-Effective Framework for Preference Elicitation and Aggregation. (5 2018). http://arxiv.org/abs/1805.05287